



KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Artūras Hakas

**Valstybinio žemės kadastro
duomenų bazės reinžinerija**

Magistro darbas

Darbo vadovas
doc. S.Gudas

Kaunas
2004

1. Įvadas	3
2. Aplinkos teorinė analizė	5
2.1. Valstybinis žemės kadastras	5
2.2. Organizacijos veiklos aprašymas	6
2.3. CASE – kompiuterizuota programų inžinerija	7
2.4. Objektiškai orientuotas IS kūrimas	8
2.5. Veiklos proceso modeliai	10
2.6. Informacijos sistemos komponentinis modelis	11
2.7. Kompiuterizuojamos srities aprašymas	19
3. Programinės priemonės	25
3.1. <i>ProvVision Workbench</i> paketas	25
3.2. <i>Microsoft access</i>	25
3.3. Programinės įrangos perdarymo inžinerija (re-engineering)	27
4. Projektinė dalis	29
4.1. Projektuojamos IS aprašymas	29
4.1.1. Biznio sąveikų modelis	29
4.1.2 Darbų sekos modelis, pirmasis lygmuo	30
4.1.3. Darbų sekos modelis, antrasis lygmuo	31
4.1.4. Panaudojimo atvejų modelis	32
4.1.5. Objektų modelis	35
4.1.6. Subtipų modelis	37
4.1.7. Prototipo generavimo rezultatai	38
4.2. Reikalavimų dokumentas	42
4.3. Projekto grafikų ir išlaidų planas	47
4.4. Rizikos mažinimas, kontrolė ir valdymas	48
4.5. Testavimo strategija	52
4.6. ER diagrama	55
4.7. Reliacinis modelis	56
4.8. Vartotojo sąsajos projektas	61
4.9. Vartotojo dokumentacija	64
5. Išvados	69
6. Literatūra	70
7. Summary	71
8. Terminų ir santrumpų žodynas	72
9. Priedai	73

1. Įvadas

Kiekviena valstybė, kuri vykdo savo nekilnojamojo turto kadastro ir registro darbus, atsižvelgdama į tai kokius duomenis ji nori turėti, kuria savo duomenų bazes skirtas, nekilnojamojo turto kadastro ir registro duomenims saugoti ir apdoroti. Lietuvoje, atkūrus nepriklausomybę, taip pat atsirado poreikis kompiuterizuotai duomenų bazei. Programinė įranga skirta, žemės kadastro ir registro duomenims rinkti, apdoroti, teikti įvairias ataskaitas buvo sukurta 1993 m. Tam naudojama programa *Clipper 6.0* sukurta duomenų bazė. Tuo metu kompiuteriuose buvo plačiai naudojama DOS operacinė sistema ir duomenų bazė atitiko jai keliamus reikalavimus. Tačiau šiuo metu atsirado suderinamumo problemų su programomis sukurtomis *Microsoft* priemonėmis.

Darbo tikslas - valstybės įmonėje, atliekančioje žemės sklypų registraciją, atlikti žemės kadastro duomenų bazės reinžineriją. Duomenų bazę pritaikyti darbui *Windows* operacinės sistemos aplinkoje, suderinti su turimomis duomenų bazėmis (užsakymų priėmimo, statinių kadastro).

Vartotojai kelia reikalavimus, kad sistema turėtų duomenų saugyklą, iš kurios duomenys lengvai būtų perduodami iš vieno proceso etapo į kitą vietinio tinklo pagalba.

Darbo priemonės - pasirinkau metodologijos realizavimo proceso modelį - IS gyvavimo ciklą. Gyvavimo ciklo pasirinkimas tiesiogiai susijęs su turimos programinės įrangos galimybėmis. Pasirinkau tradicinį GC (*top-down*) arba kitaip vadinamą - krioklio GC. Šio gyvavimo ciklo skiriami trys etapai (analizė, projektavimas, realizavimas).

Užduotis duomenų bazei :

Sudaryti žemės kadastro duomenų bazę, kurią sudaro tokie laukai: Žemės sklypo adresas, identifikavimo numeris, plotas, kaina, žemės ūkio našumo balas, asmens vardas, pavardė, asmens kodas, dokumento pavadinimas, data, numeris, grafinė informacija.

Informacijos sistemos kūrimas vykdomas asižvelgiant į pagrindinius vartotojo reikalavimus. Atliekamas bandomasis diegimas, išanalizuojamos klaidos ir aliekami pataismai.

Darbe išnagrinėta : Objektiškai orientuotas informacinės sistemos (IS) kūrimas, CASE - kompiuterizuota programų inžinerija, paketo *ProVision Workbench* teikiamos galimybės bei duomenų bazės kūrimas panaudojant *MS Access*.

Programinei įrangai buvo paruoštas projektas:

- Atlikta išsami esamos programinės įrangos analizė, išskirti jos privalumai ir trūkumai.
- Išanalizuoti vartotojo reikalavimai bei išskirti funkciniai reikalavimai.
- Sukurta visa eilė prototipų papildomiems programos funkciniam ir nefunkciniam reikalavimams išsiaiškinti.
- Sudarytas programinės įrangos architektūrinis modelis.
- Buvo sudarytos programos klasių struktūros ir nustatyti jų tarpusavio ryšiai.
- Išskirtos galimos duomenų struktūros.
- Sudarytas projekto atlikimo grafikas.
- Nustatytos projekto rizikos ir jų išvengimo priemonės.
- Sukurta vartotojo sąsaja. Jai sukurti naudoti prototipai ir esamos programinės įrangos analizės medžiaga.
- Sudarytas sistemos testavimo planas ir nustatyti būdai jam realizuoti.
- Parašyta vartotojo dokumentacija, susidedanti iš sekančių dalių: programinės įrangos paskirties aprašymo, detalaus sistemos aprašymo, žemės sklypo registravimo sekos aprašymo ir sistemos įdiegimo aprašymo.

Darbe atlikta esamos programinės įrangos analizė, išskirti jos privalumai bei trūkumai. Sudarytas žemės kadasro programinės įrangos sistemos projektas. Idėjos iliustracijai paketu *ProVision Workbench* buvo sukurti programinės įrangos prototipai. Visa tai detaliau išdėstoma darbe.

2. Aplinkos teorinė analizė

Pagrindinis darbo uždavinys : sukurti žemės kadastro ir registro IS.

2.1. Valstybinis žemės kadastras

1. Lietuvos Respublikos valstybinis žemės (su nekilnojamojo turto elementais) kadastras (toliau vadinama – Valstybinis žemės kadastras) yra Lietuvos valstybinio kadastro dalis, apimanti informaciją apie žemės kiekį, sudėtį, vertę, ūkinio naudojimo būklę ir teisinę priklausomybę. Sudarant Valstybinį žemės kadastrą, remiamasi žemėtvarkos, geodezijos, kartografijos, melioracijos, dirvožemio, miškotvarkos, nekilnojamojo turto, ekonominiais ir kitais duomenimis, taip pat teisiniais dokumentais bei aprašymais, kurie tvarkomi pagal nustatytąją sistemą;

2. Valstybinio žemės kadastro objektas yra Lietuvos Respublikos žemės fondas. Objekto pirminės apskaitos vienetas – privatūs ir valstybiniai žemės sklypai;

3. Valstybinio žemės kadastro darbų sistemą sudaro:

3.1. žemės sklypų, kaip nekilnojamojo turto objektų, įforminimas nurodant jų priklausomybę ir pagrindinę tikslinę žemės naudojimo paskirtį;

3.2. Valstybinio žemės kadastro duomenų bankų, susidedančių iš tekstinės ir grafinės informacijos apie žemės savybes, jos įvertinimą, su žeme susijusio nekilnojamojo turto vertę, žemės naudojimo apribojimus, žemės sklypų plotus, jų buvimo vietą, ribas ir unikalų numerį, formavimas, tvarkymas bei papildymas;

3.3. kartografinės ir tyrimų medžiagos, kurios reikia šiai informacijai kaupti ir sisteminti, parengimas bei atnaujinimas;

3.4. žemės kadastro žemėlapių sudarymas, tvarkymas ir papildymas;

3.5. valstybinė žemės apskaita;

4. kartografinę medžiagą, kurios reikia Valstybiniam žemės kadastrui, rengia Valstybinė geodezijos ir kartografijos tarnyba prie Lietuvos Respublikos Vyriausybės (toliau vadinama – Valstybinė geodezijos ir kartografijos tarnyba). Ši medžiaga rengiama laikantis bendrųjų kartografijos principų, pagal Lietuvos Respublikos Vyriausybės nustatyta tvarka patvirtintus standartus;

5. Valstybinį žemės kadastrą tvarko Lietuvos Respublikos Vyriausybės įgaliota Žemės ir kito nekilnojamojo turto kadastro ir registro valstybės įmonė. Duomenis, kurių reikia Valstybiniam žemės kadastrui, šios įmonės nustatyta tvarka teikia apskričių viršininkų administracijų rajonų, miestų žemėtvarkos skyriai, valstybės įmonė Valstybinis žemėtvarkos

institutas, rajonų, miestų savivaldybės, įmonės, įstaigos, organizacijos ar kitos valstybinės tarnybos bei privatūs asmenys;

6. Valstybinio žemės kadastro duomenys naudojami nekilnojamajam turtui registruoti, teritorijoms planuoti, žemėtvarkos schemoms ir projektams rengti, žemės reformos darbams atlikti, žemės savininkams apmokestinti, aplinkos apsaugai užtikrinti, Valstybinės žemės fondui valdyti ir žemės naudojimui kontroliuoti;

7. Valstybinio žemės kadastro duomenų bankai kuriami ir tvarkomi Žemės ir kito nekilnojamojo turto kadastro ir registro valstybės įmonėje pagal Lietuvos Respublikos valstybės registrų įstatymo ir Lietuvos Respublikos nekilnojamojo turto registro įstatymo nuostatas.

2.2.Organizacijos veiklos aprašymas

Veiklos apibūdinimas. Įmonė renka ir kaupia duomenis apie fizinių ir juridinių asmenų valdomus žemės sklypus, statinius ir kitą nekilnojamąjį turtą. Įmonė teikia duomenis išrašų, pažymų ir įv. suvestinius duomenis valstybinėms organizacijoms bei fiziniams ir juridiniams asmenims.

Veiklos produktas: teisiškai įregistruotas nekilnojamas turtas ir išduotas registracijos pažymėjimas, pažyma, išrašas.

Įmonės pagrindinės valdymo funkcijos:

buhalterinė apskaita ir finansų valdymas. Jų dėka yra įvertinama įstaigos finansinė būklė, ieškoma būdų kaip ją gerinti. Jų dėka nustatomas įstaigos metinis biudžetas, kuriame yra paskirstoma finansinės lėšos visiems padaliniais.

kontrolės funkcija, skirta kontroliuoti padalinių atliekamus darbus

vadovavimo funkcija – personalas yra pastoviai valdomas, kad atliktų organizacijos užsibrėžtus tikslus ir metinę programą, nurodoma gerai atlikti darbus.

Informacijos kaupimo ir rinkimo funkcija - duomenų surinkimas, kaupimas, papildymas, atrinkimas pagal tam tikrus kriterijus, netinkamų duomenų šalinimas (trynimas), ataskaitų peržiūra ir spausdinimas, įrašų peržiūra ir spausdinimas, duomenų papildymas, peržiūra ir tvirtinimas.

Numatyta kompiuterizuoti veiklos sritis – informacijos apie žemės sklypus rinkimas, kaupimas ir apdorojimas. Turimi duomenys nuolat bus tikrinamai, prireikus šalinami. Duomenų rinkimą ir visas kitas funkcijas susijusias su duomenų apdorojimu atlieka įmonės padaliniai. Aiškiau tai matyti įmonės duomenų srautų diagramuose .

Tam, kad galėtume kurti programinės įrangos projektą, pirmiausiai turime išanalizuoti pasaulyje egzistuojančius IS kūrimo būdus, iširti jų privalumus, trūkumus bei galimybes.

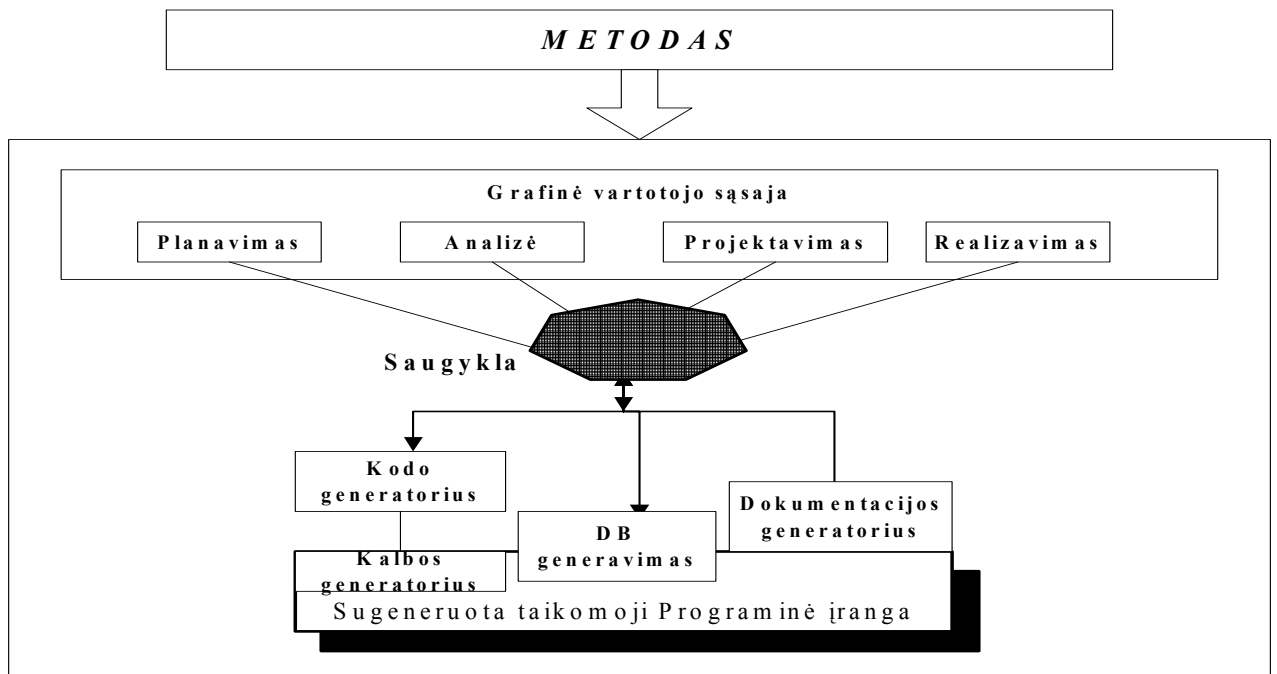
Žemiau pateikiama taikomosios programinės įrangos kūrimo technologijų raida

1. 1970-ų pradžia - struktūrinis programavimas
2. 1970 ū vidurys - struktūrinis projektavimas
3. 1970-ų pabaiga - struktūrinė analizė ir duomenų modeliavimas
4. 1980-ų pradžia - prototipų kūrimas
5. 1980ų vidurys - informacijos inžinerija (IS kūrimo proceso formalizavimas)
6. 1980ų vidurys - JAD metodas - IS kūrimas kartu su vartotoju,
7. 1980ų vidurys - CASE priemonės (struktūrinių-funkcinių metodų pagrindu),
8. 1980ų pabaiga - RAD technologijos - greitas IS kūrimas
9. 1990ų pradžia - Objektinis modeliavimas ir projektavimas
10. 1990ų vidurys - CASE priemonės OO metodų pagrindu;
11. 1990pabaiga - agentinis modeliavimas (Agent-oriented modeling)
12. 2000m. - metodų ir priemonių UML pagrindu plėtra (Fusion, RUP, Catalysis, Rubico, OCTOPUS)

Detaliau panagrinėsime keletą IS kūrimo būdų.

2.3.CASE - kompiuterizuota programų inžinerija

Paskutinė naujovė struktūriniame programavime yra CASE - kompiuterizuota programų inžinerija. CASE priemonių pagalba kompiuteris valdo funkcinės dekompozicijos procesą, analizuodamas grafines schemas (diagramas), kuriose nurodomos paprogramės ir jų tarpusavio sąveikos. šiuolaikinė CASE sistema iš tiesų gali sukurti išbaigtą, tinkančią vartojimui programų sistemą pagal diagramas ir kitą projektavimui reikalingą informaciją. Tipinė IS kompiuterizuoto projektavimo aplinka - CASE sistemos architektūra pavaizduota 1 paveiksle.



1 pav. CASE sistemos architektūra

CASE priemonės nekuria programinės įrangos, jos tiesiog paverčia grafinį sistemos projektą tekstiniu projektu.

Patirtis rodo, kad išbaigto grafinio projekto sukūrimas reikalauja tiek pat daug laiko ir pastangų, kaip ir programos kūrimas įprastiniu būdu.

2.4. Objektiškai orientuotas IS kūrimas

Objektiškai orientuotas IS kūrimas – tai ne tik OO programavimas. Objektiškai orientuotas IS kūrimas tai : OO biznio analizė ir modeliavimas + OO informacinių sistemų projektavimas + OO programavimas + OO duomenų modeliai ir OO DBVS.

Objektinis metodas skiriasi nuo funkcinės metodologijos, kurioje pagrindinis dėmesys skiriamas sistemos funkcijoms specifikuoti. Objektinis požiūris pirmiausia skiria dėmesį probleminės srities objektams identifikuoti, po to sutelkia aplink juos funkcijas. Tai leidžia lengviau pritaikyti programinę įrangą, keičiantis vartotojo reikalavimams.

Objektiškai orientuotas IS projektavimas susideda iš eilės žingsnių. Prieš vykdant kiekvieną iš projektavimo žingsnių turi būti surinkta atitinkama informacija apie organizacijos veiklą, suklasifikuota, nustatytos vykdomų operacijų klasės ir jų vykdymo sąlygos.

Pagrindinės OO modeliavimo (analizės ir projektavimo) kalbos:

UML	<i>Unified modeling language, 1995</i>
OML	<i>Open modeling language</i>

Tarptautinio lygmens standartu laikomos objektinio modeliavimo kalbos **UML** (*Unified modeling language*) sukūrimas siejamas su Grady Booch, James Rumbaugh ir I.Jacobson vardais.

Objektinė modeliavimo technologija (OMT) sistemos atvaizdavimui naudoja 3 modelius:

- **objektų modelį**, aprašantį objektus, klases ir jų ryšius;
- **dinaminį modelį**, aprašantį sąveiką tarp objektų klasių;
- **funkcinį modelį**, aprašantį sistemoje vykstančias duomenų transformacijas.

Visi šie modeliai praeina visus sistemos sukūrimo etapus. Pilnam projektuojamos IS aprašymui reikalingi visi 3 modeliai.

Objektų modelis

Objektų modelis aprašo statines sistemos savybes

Objektų modelis susideda iš klasių diagramų (*class diagrams*) ir objektų-(egzempliorių) diagramų (*instance diagrams*).

Klasių diagrama yra grafas, kurio viršūnės yra objektų klasės, o lankai - santykiai tarp objektų klasių.

Objektų diagrama atitinka klasių diagramą, tačiau jos elementai žymi konkrečius probleminės srities objektus (egzempliorius).

Dinaminis modelis

Dinaminis modelis aprašo savybes, kurios keičiasi laike. Dinaminis modelis specifikuoja ir realizuoja sistemos valdymo aspektus.

Dinaminį modelį sudaro būsenų diagramos (*state diagram*) ir įvykių sekos diagramos (*event trace diagrams*).

Būsenos diagrama yra grafas, kurio viršūnės yra būsenos, o lankai - **įvykių** iššaukti perėjimai tarp būsenų.

Įvykių diagrama nurodo sistemos veiklos metu atsirandančių įvykių, kurie sieja konkrečių objektų aibę, seką.

Funkcinis modelis

Funkcinis modelis aprašo duomenų transformacijų eigą sistemoje. Funkcinį modelį sudaro duomenų srautų diagramos (*DFD - data flow diagrams*).

Duomenų srautų diagramos atvaizduoja skaičiavimus. **Duomenų srautų diagrama** yra grafas, kurio viršūnės yra procesai, o lankai - duomenų srautai.

Šie trys modeliai yra tarpusavyje susieti. Svarbiausias yra objektinis modelis, kadangi pirmiausia reikia aprašyti, **kas** keičias ar transformuojasi, o po to - **kada ir kaip**.

2.5. Veiklos proceso modeliai

Ir struktūrinėje, ir objektinėje informacijos sistemų (IS) kūrimo technologijoje sudaromas kompiuterizuojamos organizacijos veiklos modelis. Veiklos modelio paskirtis yra aprašyti duomenų transformacijas sistemoje.

1970-jų pabaigoje Chris Gane ir Trish Sarson sukūrė grafiniais modeliais – duomenų srautų diagramomis pagrįstą sistemų analizės būdą. Jų metodologija naudojama dar ir šiomis dienomis kaip vienas populiariausių metodų struktūrinėje analizėje ir sistemų projektuose. Klasikiniu tapęs terminas “Data Flow Diagrams” (DFD) į lietuvių kalbą verčiamas “duomenų srautų diagramos”. Jos skirtos veiklos sričiai apibrėžti, t.y. sistemos funkcijoms (procesams) ir jų sąveikoms (technologiniams ir duomenų srautams) vaizduoti. Naudojant grafinius metodus suteikiama galimybė vartotojams, analitikams ir projektuotojams gauti aiškų ir bendrą sistemos paveikslą. Taip pat, analizuojant ir patenkinant vartotojų poreikius, DFD leidžia matyti, kaip atskiros sistemos dalys dera viena prie kitos.

Duomenų srautų diagramas galima sudaryti keliomis skirtingomis notacijomis: Gane-Sarson, Ward & Mellor, Yourdan/DeMarco.

Metodologija apima sistemos kūrimą “iš viršaus žemyn”, žingsnis po žingsnio iki galutinės tobulos sistemos. Pirmiausiai sukuriama aukščiausiojo lygmens DFD, kurioje visa analizuojama organizacija (t.y. visa veiklos sritis) žymima kaip vienintelis procesas, modeliuojamos išorinės organizacijos sąveikos su aplinka.

Toliau kuriama vienintelė nulinio lygmens DFD, kuri nurodo pagrindinius analizuojamos veiklos (organizacijos) procesus ir juos siejančius srautus. Po to dekomponuojamas kiekvienas nulinio lygmens DFD procesas, jiems sudaromos atskiros DFD, detalai aprašančios vidinius subprocesus ir jų sąveikas. Duomenų srautų ir procesų detalės turi būti aiškiai apibrėžtos: kiekvienam srautui ir procesui suteikiamas vardas (identifikatorius), jie trumpai aprašomi. Taip suformuojamas hierarchinis veiklos modelis - DFD hierarchija. Duomenų srautai ir procesai būtinai turi savo pavadinimus (identifikatorius).

Duomenų srautai yra vienas iš modelio elementų, kuris parodo, kokie duomenys naudojami procesuose, iš kur jie imami ir kur saugomi. Duomenų srautų šaltiniai ir vartotojai yra veiklos dalyviai (išoriniai objektai) arba duomenų saugyklos.

Kitas, pagrindinis, diagramos elementas yra procesas. Pagal apibrėžimą procesas modeliuoja duomenų apdorojimo operacijas. Tačiau DFD taikomos ir organizacijų veiklai modeliuoti, aprašyti gamybinius procesus ir materialius srautus (medžiagų, žaliavų, energijos ir kt.). Todėl literatūroje pagal DFD notaciją sudaryti modeliai, kuriuose yra informaciniai ir materialūs procesai bei srautai, dar vadinami “procesų diagramomis” arba “kontekstinėmis diagramomis”.

Esamą sistemą aprašysiu naudodamasis *Ward & Mellor* notacija.

2.6. Informacijos sistemos komponentinis modelis

Šiuo metu yra aktualus integruotų organizacijos informacijos sistemų kūrimas, tokių kaip ERP sistemos (*Enterprise Resource Planning Systems*). Sudėtingų informacijos sistemų (IS) kūrimas turėtų vykti efektyviai - sparčiai ir kokybiškai, surenkant jas iš komponentų. Atlikus integruotų informacijos sistemų projektavimo metodų analizę, pastebėta tendencija integruoti architektūrinį ir komponentinį IS kūrimo metodus. Architektūrinis organizacijos veiklos modelis reikalingas sistemos dalių integralumui užtikrinti, o komponentinis - dalių pakartotino panaudojimo galimybei užtikrinti.

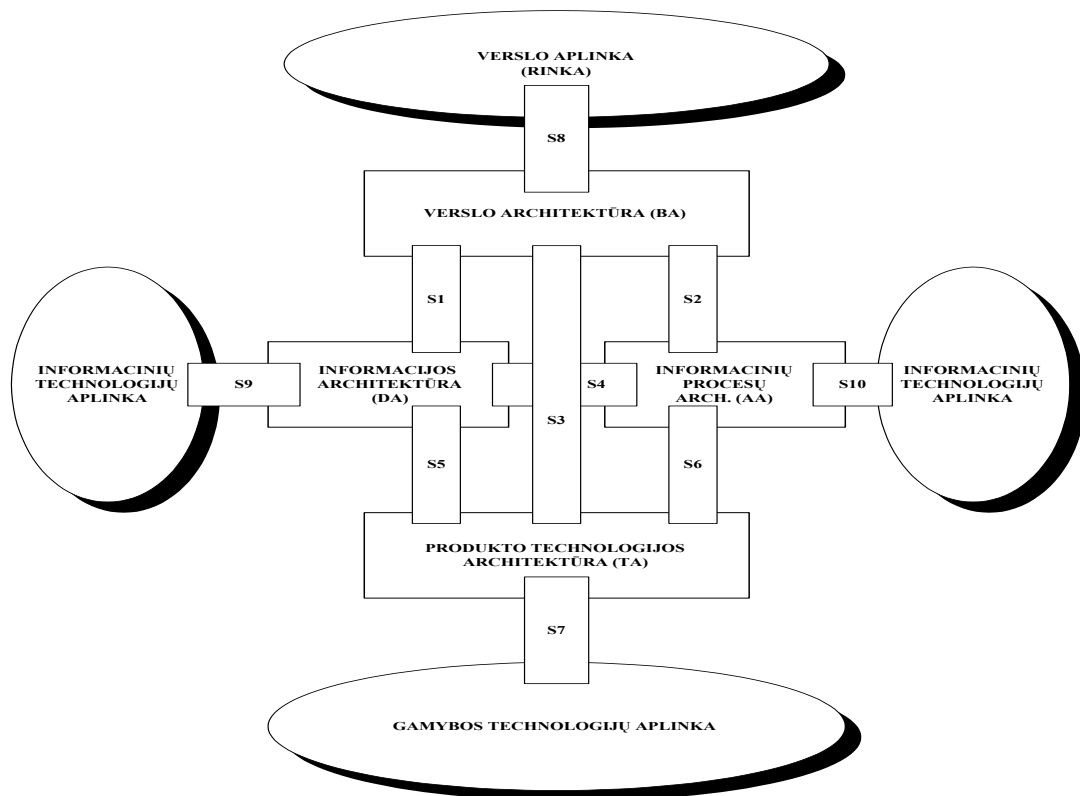
Architektūra grindžiamas IS projektavimas

Viena iš pažangiausių veiklos procesų ir taikomųjų programų integravimo metodologijų vadinama “architektūriniu modeliavimu” ar “architektūra grindžiamas IS projektavimas” (architecture-driven). Veiklos informacinė architektūra apima bendros sistemos struktūros, sistemos komponentų, loginių jų ryšių ir išoriškai matomų savybių modeliavimą (projektavimą). Organizacijų informacinės architektūros modeliavimas tiesiogiai skirtas informacijos sistemų, atitinkančių realius veiklos poreikius, projektavimo ir realizavimo metodams vystyti.

Organizacijos veiklos informacinės architektūros modelis

Organizacijos veiklos informacinės architektūros (VIA) modeliavimas skirtas informacijos sistemų, atitinkančių realius veiklos poreikius, projektavimo ir realizavimo metodams plėtoti. Organizacijos veikla gali būti nagrinėjama iš skirtingų pozicijų, išskiriant skirtingo pobūdžio veiklos dalykines sritis, vadinamas veiklos domenais. Veiklos domenai nurodo organizacijos dalis, kuriose vyksta skirtingos prigimties procesai.

Organizacijos veiklos informacinės architektūros (VIA) modelis, sudarytas iš keturių domenu, pateiktas 2 paveiksle.



2 pav. Organizacijos veiklos informacinės architektūros (VIA) modelis

Pagrindiniai organizacijos veiklos domenai, kurių visuma ir sąveikos užtikrina organizacijos funkcionavimą, aprašyti 1 lentelėje.

1 lentelė. Pagrindiniai organizacijos veiklos domenai

Veiklos domenas	Žymėjimas	Domeno paskirtis
Verslo procesų domenas	BD	Tai ekonominę ir gamybinę veiklą vykdančios organizacijos dalies (valdymo funkcijos, ekonominė veikla) informaciniai poreikiai ir reikalavimai IS
Informacijos domenas	DD	Tai duomenys, žinios ir tikslai, jų saugojimo ir perdavimo organizacijos padaliniams procesai;
Informacinių procesų domenas	IPD	Organizacijoje atliekami skaičiavimai, sprendimo priėmimo procesai, galima vadinti taikomųjų uždavinių domenu
Technologinių procesų domenas	TPD	Tai organizacijos dalies, atliekančios produkto gamybą (ar formavimą) - produkto gamybos procesų informaciniai poreikiai ir reikalavimai IS
Darbo vietų domenas	DVD	Darbo vietų visuma, informaciniai reikalavimai darbo vietose atliekamoms funkcijoms. Darbo vietų

		domenas yra pasiskirstęs, t.y. darbo vietos išsidėstę kituose domenuose ir sąsajose
--	--	---

Pirmieji keturi domenai yra pagrindiniai. Literatūroje yra minimas dar vienas (išvestinis) organizacijos veiklos domenas – darbo vietų (KDV) domenas . Darbo vietų domenas - tai organizacijos suvokimas kaip darbo (veiklos) vietų visumos, aprašant reikalavimus darbo vietose vykdomoms funkcijoms

Kiekvienas domenas žymi specifinę organizacijos veiklos sritį (biznio procesus, informaciją, informacijos apdorojimo procesus, produkto gamybos procesus), kuri modeliuojama kaip savarankiškas objektas (komponentė).

VIA domenų sąsajų tipai:

Veiklos domenai sąveikauja tarpusavyje. Domenų sąsajų paskirtis yra integruoti domenų sąveikas, siekiant organizacijos tikslų. Domenų sąsajų architektūros modelis yra gaunamas iš informacijos, surinktos apie domenus, t.y. yra išvedamas iš domenų informacinės architektūros modelio. Domenų sąsajos yra šių domenų informacinės architektūros (IA) komponentų sąsajos (2 lentelė).

Domenų sąsajų tipai aprašyti 2 lentelėje. Sąsajų S1 – S6 paskirtis yra integruoti domenų tarpusavio sąveikas, sąsajų S7 –S10 paskirtis - užtikrinti domenų sąveikas su išorine aplinka.

2 lentelė. Sąsajų tarp veiklos domenų tipai

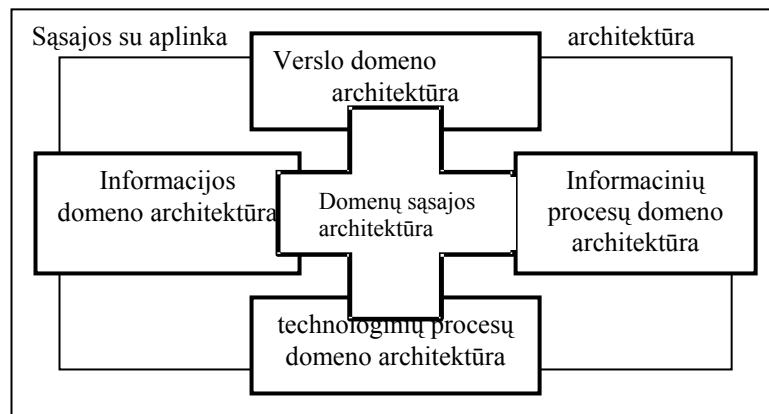
Sąsajos tipas	Nurodyto tipo sąsajos siejami komponentai
S1	Duomenų domeno (duomenų komponentų) ir verslo domeno (verslo komponentų) sąsaja
S2	Informacinių procesų domeno (funkcinių komponentų) ir verslo domeno (verslo IA komponentų) sąsaja
S3	Technologinių procesų domeno (TP IA komponentų) ir verslo domeno (komponentų) sąsaja
S4	Informacinių procesų domeno (funkcinių komponentų) ir informacijos domeno (komponentų) sąsaja
S5	Technologinių procesų domeno (TP IA komponentų) ir informacijos domeno (duomenų komponentų) sąsaja
S6	Informacinių procesų domeno ir technologinių procesų domeno (TP IA komponentų) sąsaja
S7	Technologinių procesų domeno sąsaja su produkto gamybos

	technologijų aplinka (aplinkos IA komponentais).
S8	Verslo domeno sąsajos su verslo aplinka
S9, S10	Informacinių procesų domeno ir informacijos domeno sąsajos su informacinių technologijų aplinka (aplinkos IA komponentais)

Kiekviena domenu (t.y. jų atitinkamų komponentų) sąsaja modeliuojama kaip atskiras objektas - atskira IS architektūros komponentė, siejanti du konkrečius VIA domenus.

Skirtingų domenu tarpusavio informacinės sąveikos užtikrinimas, taip pat pagrindinių domenu sąveikos su išorine aplinka realizavimas ir reiškia organizacijos veiklos integravimą.

3 paveiksle yra apibendrintas veiklos informacinės architektūros modelis, kuriame skirtingų tipų domenu sąsajos sujungtos į vieną bloką (interfeisų lauką), per kurį integruojama VIA pagrindinių domenu sąveika.



3 Pav. Veiklos informacinės architektūros modelis (apibendrintas)

Komponentinio IS projektavimo metodo principai

Komponentinis projektavimas teoriškai turi daug privalumų, iš kurių svarbiausias - pakartotino komponentų panaudojimo galimybė. Dėl šios savybės padidėja produktyvumas, palaikymo ir modifikavimo galimybės, o lygiagrečiai sumažėja projekto kūrimo ciklas ir kaštai.

IS projekto lygmens komponentai projektuojami pagal modelių pagrįstą (*model-driven*) projektavimo paradigmą, kurioje komponentai paveldi aprašus iš veiklos proceso modelio. IS komponentai turi būti visiškai save aprašantys. Tai reiškia, kad IS komponentas turi aiškiai apibrėžtą interfeisą ir atitinka nurodytą elgseną, bendrą visiems sistemos architektūros vidaus komponentams.

Aptariamas metodas aprašo architektūrinio IS projektavimo etapą, kuriame identifikuojami IS projekto komponentai ir jų sąsajos (interfeisai). Toliau, detalaus

projektavimo etape, komponentai turi būti specifikuojami, parengiant projektą IS programinės įrangos generavimui.

IS projekto komponentai yra skirstomi į vartotojo sąsajos komponentus (menu, ekrano formos, ataskaitos), duomenų komponentus (duomenų bazėse ar duomenų saugykloje talpinami informacijos vienetai), funkcinis komponentus (skaičiavimai ir taikomųjų uždavinių logika). IS projekto komponentus identifikuoja projektuotojas, CASE sistemos aplinkoje analizuodamas darbų sekų modelį, kuris aprašo konkrečią veiklos funkciją ar procesą. Taip projektuotas sudaro komponentinį sistemos modelį, kuris aprašo identifikuotus IS komponentus ir jų sąveikas. Toliau sudaromi žemesnių lygmenų komponentiniai sistemos modeliai, taip tikslinama IS komponentų sudėtis ir specifikacijos. Detalaus IS komponentų specifikavimo etape gali būti naudojami atitinkami objektiniai modeliai (UML, OML).

IS projekto komponentus realizuoja programinės įrangos lygmens komponentai. Programinės įrangos lygmens komponentas yra programinės įrangos objektas, sąveikaujantis su kitais komponentais, atliekantis tam tikrą funkciją ar aibę funkcijų. Komponentų valdymo ir funkcionavimo optimizavimo priemonės naudoja vieningą komponentų aprašų saugyklą.

Komponentinio sistemos modelio elementai

Organizacijos informacijos sistemos komponentams ir sąsajoms tarp jų identifikuoti siūloma nauja grafinė notacija – komponentinis sistemos modelis. Šis modelis apjungia veiklos informacinės architektūros (VIA) modelio ir darbų sekos modelio savybes.

Veiklos informacinės architektūros modelis apibrėžia IS komponentų tipus, atitinkančius organizacijos veiklos domenų, kurie aprašyti 1 lentelėje. Remiantis tuo, komponentinis sistemos modelis (analogija su darbų sekų modeliu) skirstomas į penkis takelius, kurie skirti atitinkamo vieno veiklos domeno komponentams:

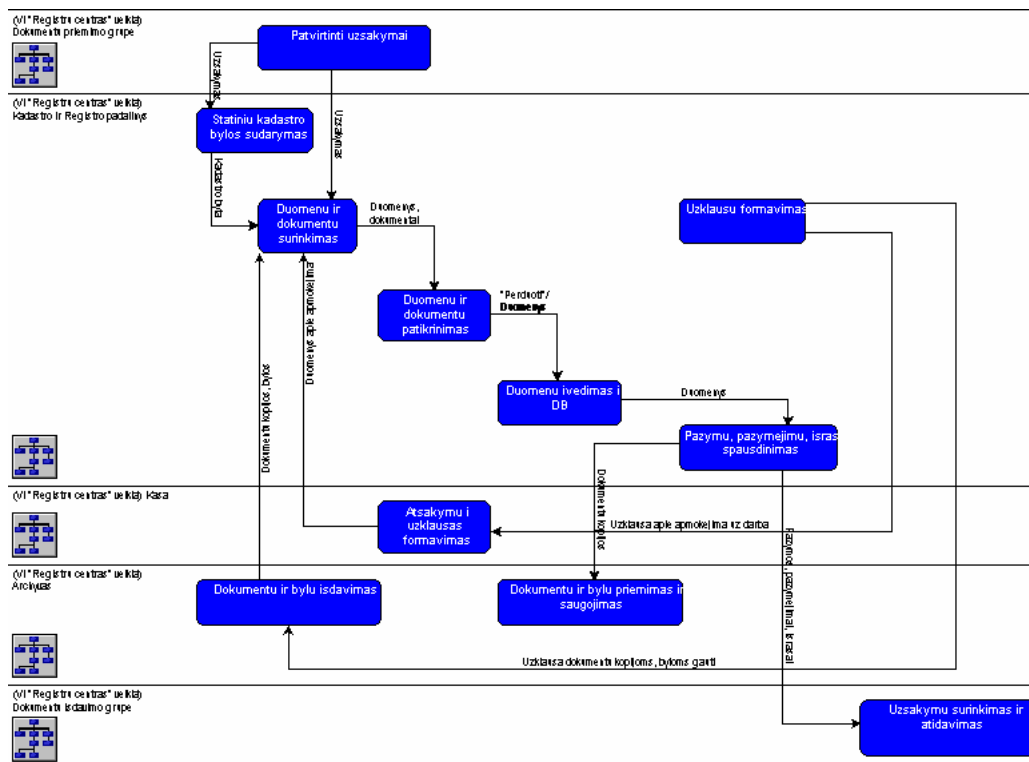
- takelis “valdymo funkcijos” atitinka verslo domeną ir skirtas šiame domene naudojamiems IS komponentams (tai IS vartotojo sąsajos komponentai) specifikuoti;
- takelis “taikomieji uždaviniai” atitinka informacinių procesų domeną ir skirtas IS taikomųjų uždavinių logiką (skaičiavimus ir kitoki duomenų apdorojimą) realizuojantiems komponentams (tai IS funkciniai komponentai) specifikuoti;
- takelis “duomenų struktūros” atitinka informacijos domeną ir skirtas IS saugyklose (duomenų bazėse, duomenų sandėliuose) saugomos informacijos elementams, t.y. duomenų komponentams specifikuoti;
- takelis “technologiniai procesai” atitinka technologinių procesų domeną ir skirtas šiame domene naudojamiems IS komponentams (tai IS vartotojo sąsajos komponentai) specifikuoti;

- takelis “išorinės aplinkos veiksniai” atitinka VIA modelio aplinkos domenų (verslo rinkos, technologijų ir informacinių technologijų rinkos) ir skirtas šiuose domenuose esantiems aktualiems komponentams (sąveikaujantiems su jau aptartais IS komponentais) specifikuoti.

Darbų sekos modelis

Darbų sekos modelis yra kuriamas per biznio sąveikų modelį. Naudojant darbų sekos modeliavimą galima sukurti labiau detalesnį veiklų modelį.

Šis modelis atvaizduoja biznio procesus išreiškiant juos veiklos komponentais ir darbų seka tarp tų veiklų. Darbų sekos modelis koncentruotas į darbų seką nuo biznio pradžios iki galo. Tai atspindi pagrindinis aukščiausio lygio modelis. Jei koks procesas ar darbas yra sudėtinis kitų darbų atžvilgiu, tai šiam procesui yra sukuriamas detalesnis žemesnio lygio darbų sekos modelis. Darbų sekos modelio pavyzdys pateikiamas 4 paveiksle.



4 pav. Darbų sekos modelis

Komponentinio organizacijos veiklos modelio sudarymas

Komponentinis sistemos modelis suformuojamas transformuojant darbų sekų modelį pagal šias taisykles:

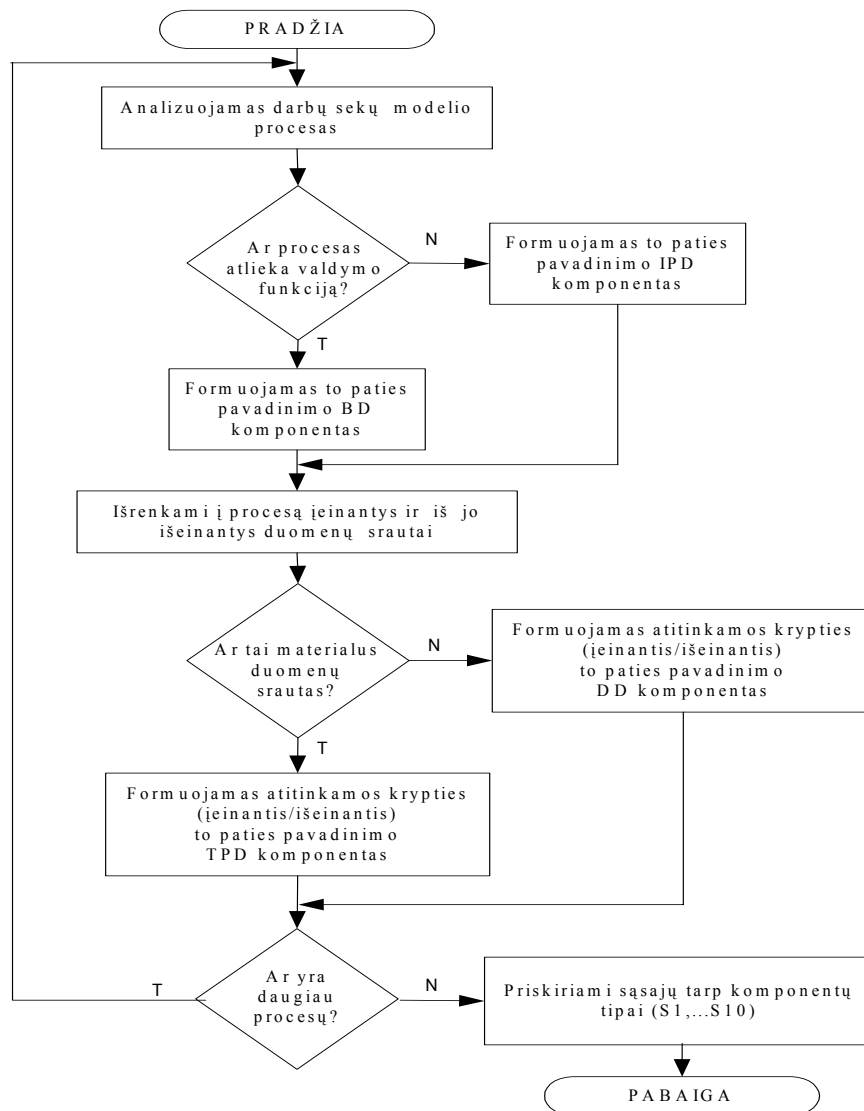
Darbų sekos modelyje skaičiavimą atliekantys procesai transformuojami į informacinių procesų domeno (IPD) komponentus.

Darbų sekos modelyje valdymą atliekantys procesai transformuojami į verslo domeno (BD) komponentus.

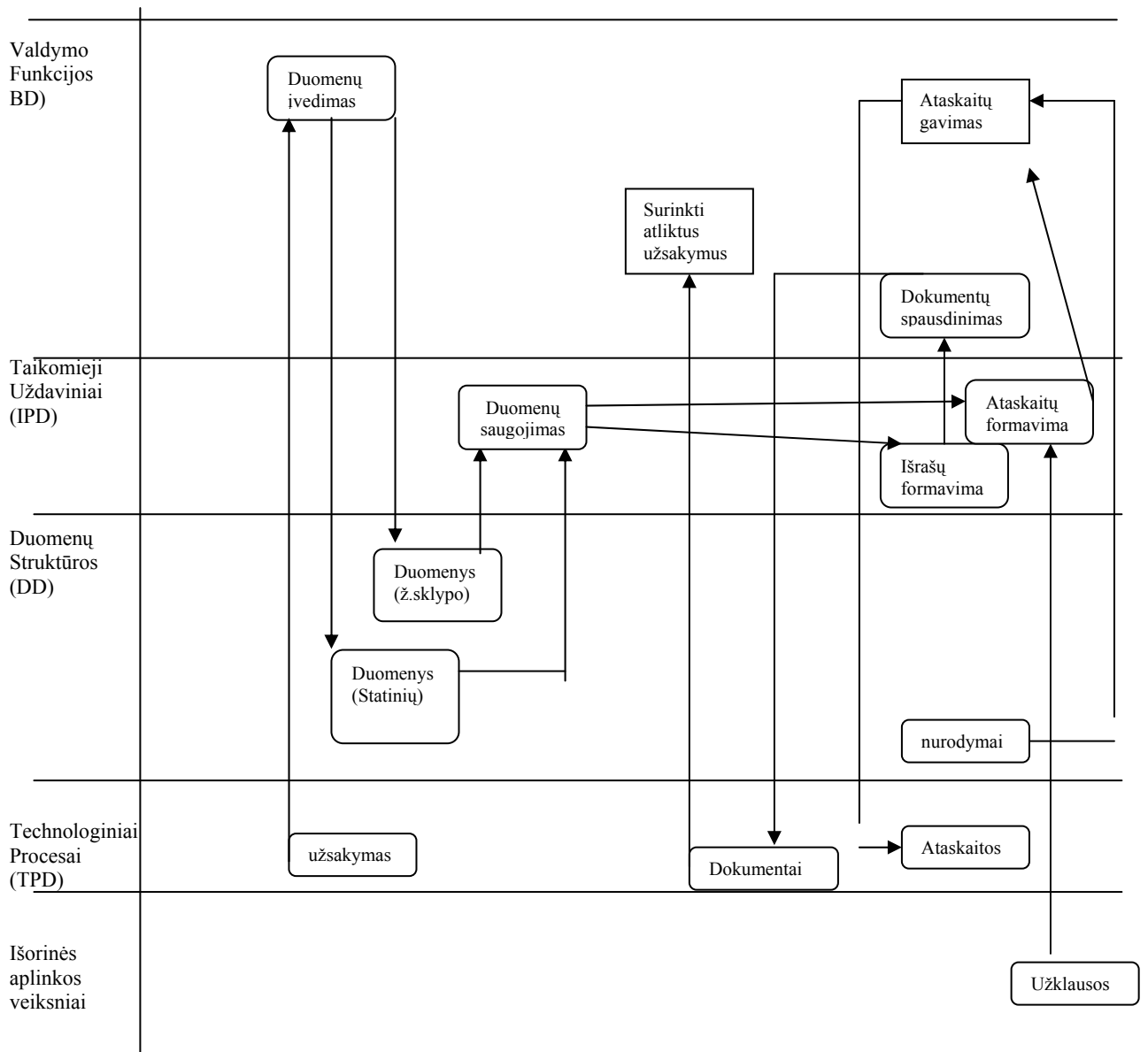
Informacijos srautai, jungiantys procesus darbų sekos modelyje, transformuojami į duomenų domeno (DD) komponentus.

Materialūs srautai darbų sekos modelyje transformuojami į technologinių procesų domeno (TPD) komponentus.

Darbų sekos modelio transformavimo į komponentinį sistemos modelį aiškina algoritmas:



Komponentinio sistemos modelio pavyzdys pateikiamas 5 paveiksle.



5 pav. Komponentinis sistemos modelis

Analizuojant komponentinį sistemos modelį, galima numatyti kokie bus reikalingi vartotojo sąsajos langai.

Išvados

Projektuojant integruotas kompiuterizuotas informacijos sistemas tikslinga apjungti IS kūrimą informacinės architektūros modelio pagrindu bei komponentinį IS projektavimą, siekiantį surinkti IS iš kompiuterizuotų veiklos komponentų.

Pasiūlytas metodas susieja IS architektūros modelį, darbų sekų modelį ir atvaizduoja juose esančią informaciją į naujo tipo modelį - komponentinį sistemos modelį, kuriame išskiriami tokio tipo komponentai: valdymo funkcijos (BD), skaičiavimai arba funkciniai komponentai (IPD), duomenų struktūros (DD), technologiniai procesai (TPD), išorinė aplinka (ENV).

Pagrindiniai komponentinio sistemos modelio sudarymo tikslai yra išsaugoti veiklos modelyje egzistuojančias sąsajas tarp IS informacinės architektūros komponentų bei tiksliau specifikuoti komponentus ir jų sąsajas. Toks modelis padėtų užtikrinti organizacijos veiklos ir visų projektuojamų sistemų integralumą

Organizacijos IS komponentų identifikavimas leistų IS projekte pakartotinai naudoti specifikuotus komponentus, kurie išsaugo veiklos srities procesų ir objektų semantiką, o realizavus projektą - surinkti IS taikomąsias programas iš komponentų, turinčių prasminius vardus aprašytoje straipsnyje sistemos komponentų diagramoje.

Naudojantis aprašytu metodu, galima modifikuoti CASE priemonių (ir struktūrinių funkcinių, ir objektiškai orientuotų), kuriose naudojami veiklos procesų modeliai (darbų sekų modeliai arba jų analogai), projektavimo aplinkas.

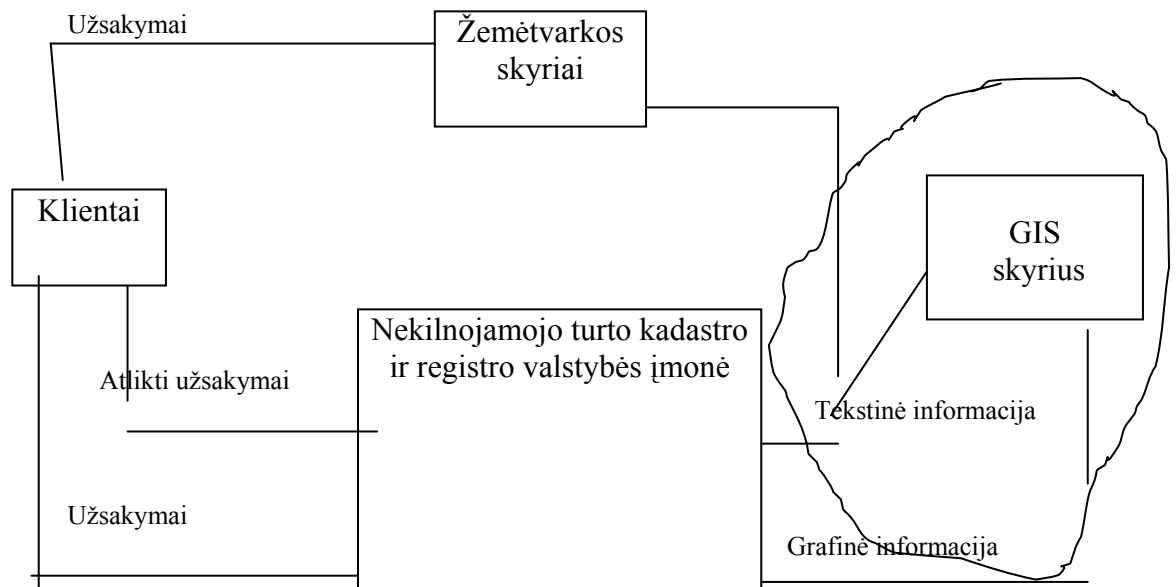
2.7. Kompiuterizuojamos srities aprašymas

Šiuo metu žemės kadastrui ir registrui vesti naudojama *Clipper* kalba parašyta programa, kuri yra pritaikyta dirbti *Norton Comander* aplinkoje. Kitos programos naudoja *Microsoft Access* duomenų bazines, todėl esama programa negali pasinaudoti tais duomenimis. Dirbant *Norton Comander* aplinkoje, iškyla darbo tinkle problemos. Programa gerai dirba, esant ne daugiau kaip trimis darbo vietoms, kitu atveju atsiranda nenumatytų programos reakcijos laiko sutrikimų ir kitų problemų. Esamai ir projektuojamai sistemoms palyginti, pasirinkau duomenų srautų diagramomis pagrįstą sistemų analizės būdą ir Taikomųjų uždavinių modelius.

Sistemą aprašysiu naudodamasis *Ward & Mellor* notacija.

Aukščiausio lygmens DFD

Aukščiausio lygmens DFD vaizduoja Nekilnojamojo turto kadastro ir registro VĮ aplinką. Ši duomenų srautų diagrama parodo, kokie elementai yra sistemos išorėje ir ryšį su jais. Kaip matome diagramoje, išoriniai elementai yra: Klientai, žemėtvarkos skyriai ir GIS skyrius (numatoma, kad turės ryšį su kitais elementais naujai projektuojamoje sistemoje)

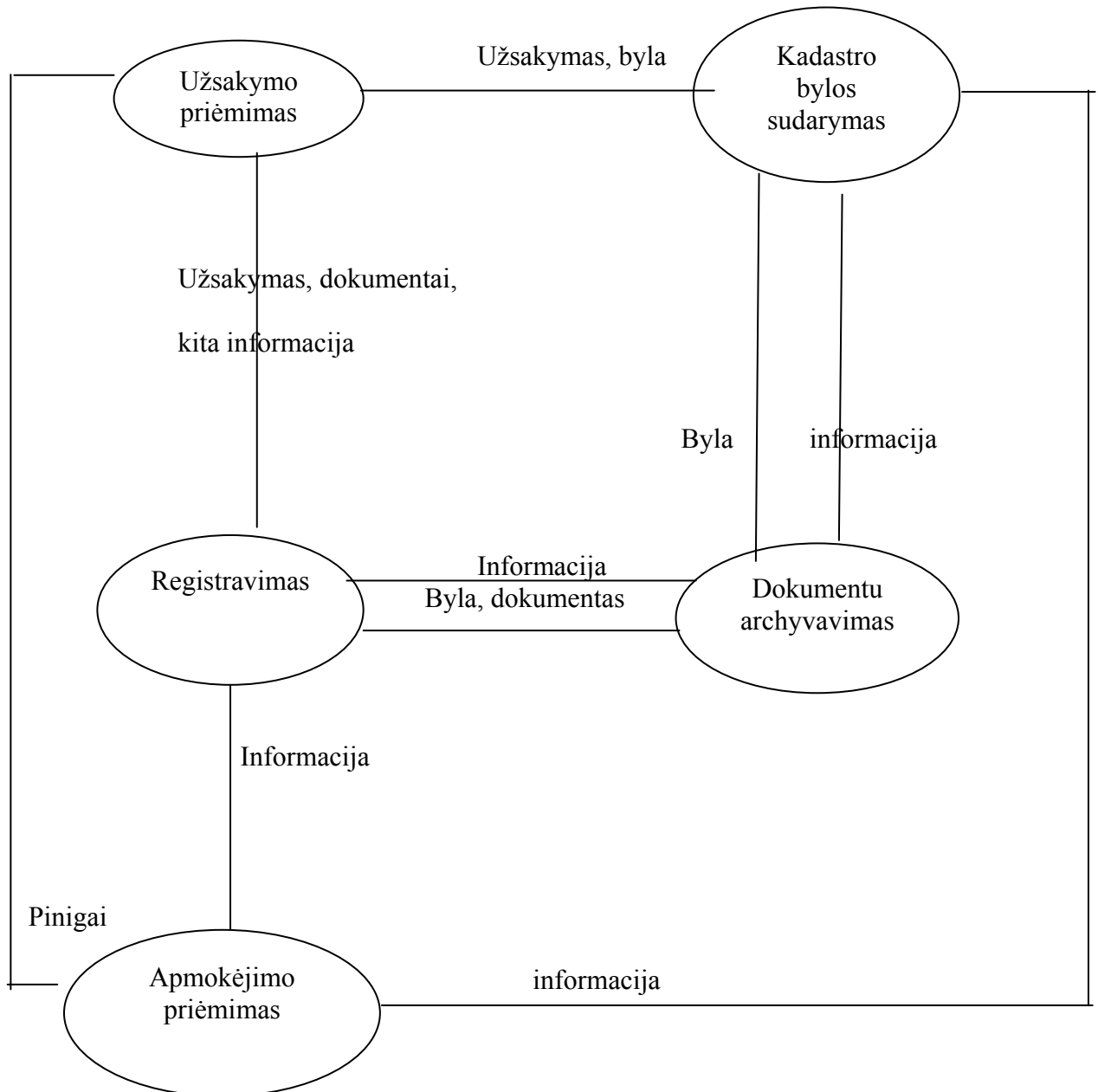


6.pav. Aukščiausio lygmens DFD

Pastaba: Vingiuota linija apvesta dalį numatoma įvesti projektuojant naują sistemą.

Nulinio lygio DFD

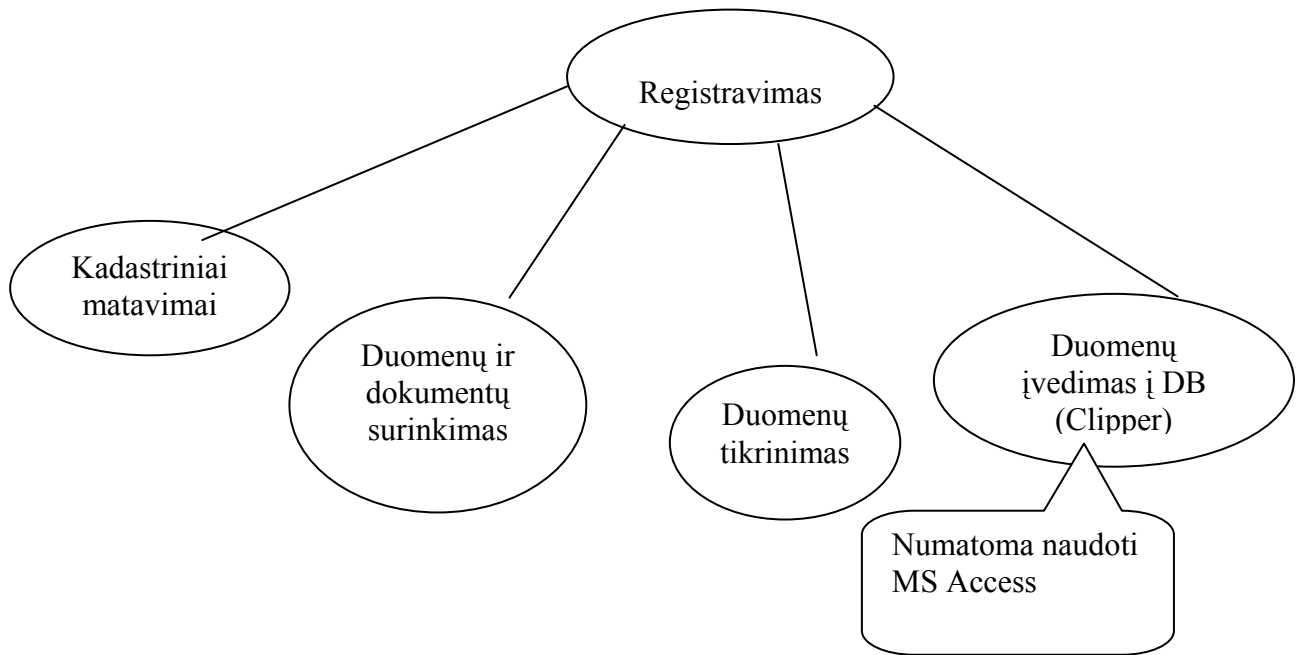
Nulinio lygio duomenų srautų diagrama vaizduoja pagrindinius sistemos procesus, duomenų srautus – detalizuojama aukščiausio lygmens DFD. Nulinio lygio DFD



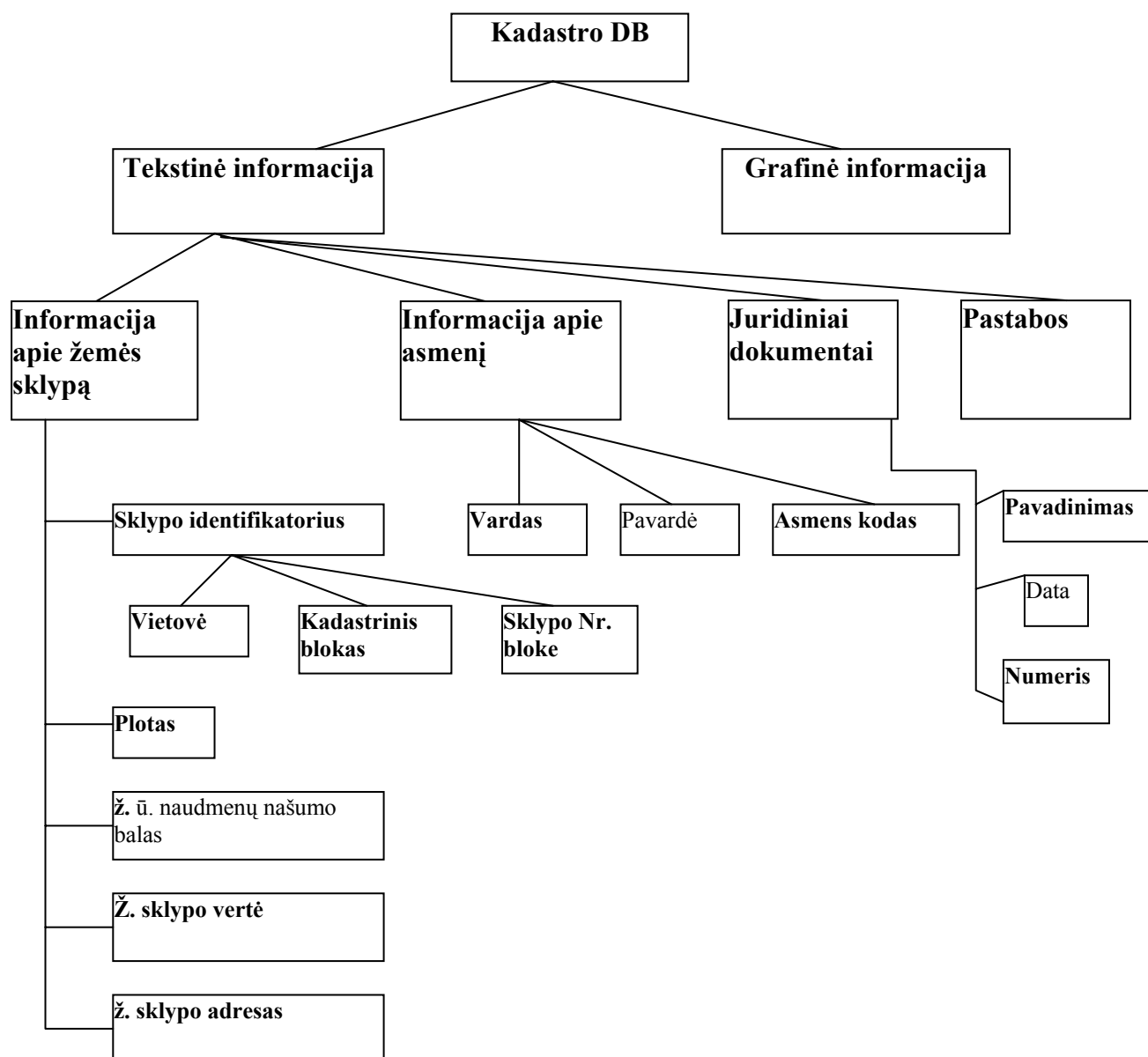
7.pav. Nulinio lygio DFD

Pirmo lygio DFD

Kiekvieną procesą, pavaizduotą nulinio lygio diagramoje, paprastai sudaro keli subprocesai. Smulkiau detalizuojame registravimo procesą. Įvedame papildomą elementą – duomenų bazę.

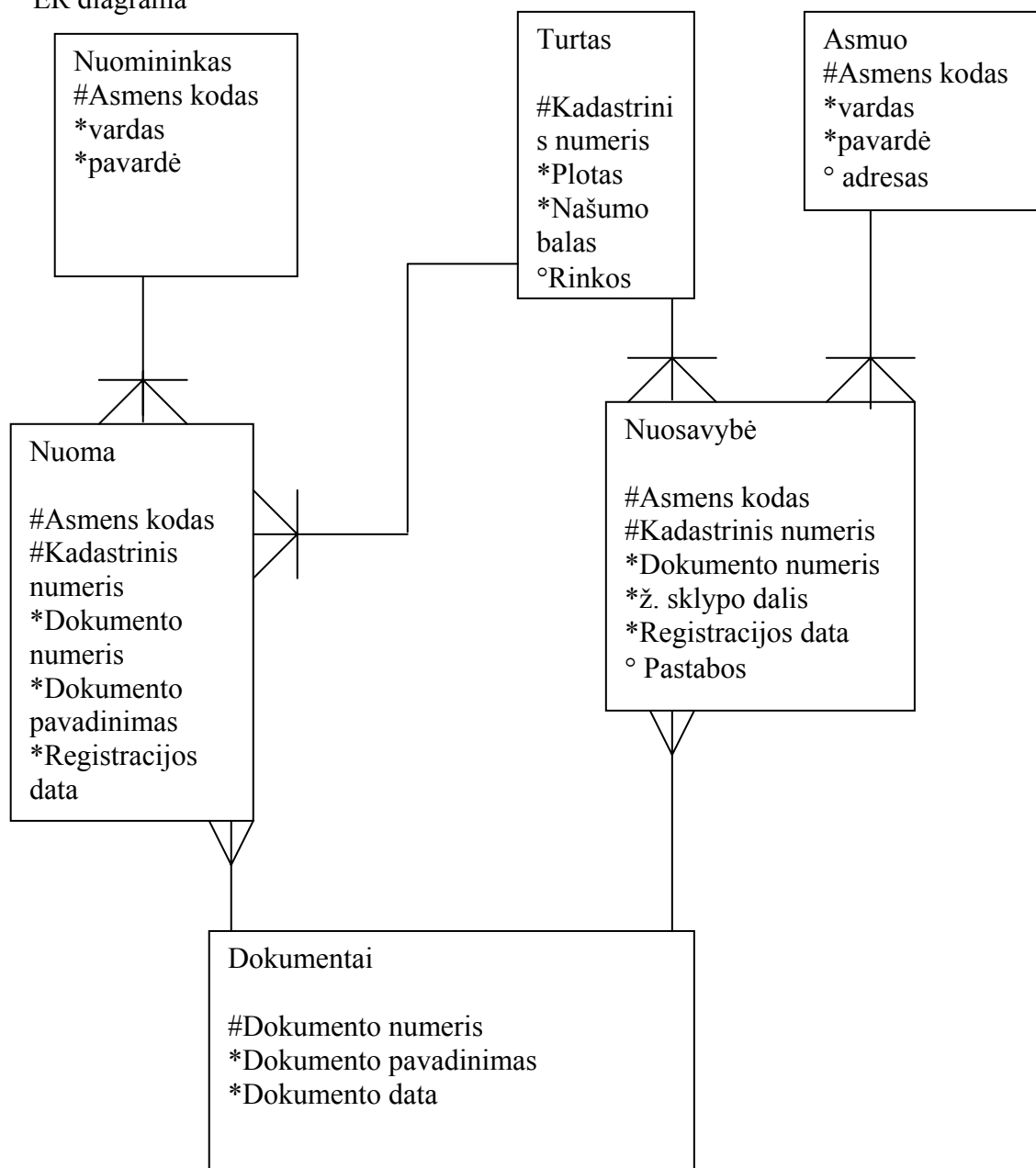


8.pav. Pirmo lygio DFD



9.pav. Duomenų struktūros modelis

ER diagrama



10.pav. ER diagrama

ER diagrama

Realaus pasaulio modeliavimo būdas, kuris vadinamas esybių-ryšių modeliavimu, skirtas projektuoti duomenų bases. Esybių-ryšių modeliavimo metu identifikuojami svarbūs organizacijos objektai (vadinami esybėmis), šių esybių savybės (atributai) ir tarpusavio ryšiai. Esybių-ryšių modeliavimo tikslas - sudaryti organizacijos, kuriai kuriama nauja arba tobulinama jau egzistuojanti informacijos sistema, **konceptualų duomenų modelį**, atvaizduojama grafiškai esybių-ryšių diagrama (ERD). Esybių diagramoje pastebimas ryšių kardinalumas (1:∞).

3. Programinės priemonės

3.1. *ProVision Workbench* paketas

Išanalizavęs eilę CASE priemonių, kuriamos IS aplinkos analizavimui, modelių kūrimui pasirinkau *ProVision Workbench* paketą.

Palyginus su daugelio tipo modeliavimo priemonių *ProVision Workbench* paketas plačiausiai apima pagrindinius gyvavimo ciklo etapus :

- Biznio procesų modeliavimą ir reinžineriją (BPR);
- Biznio objektų modeliavimą;
- Informacinės sistemos modeliaivimą;
- Informacinės sistemos realizavimą.

ProVision Workbench – tai organizacijos modeliavimo priemonė, kuri apjungia BPR ir OO į vieną integruotą modeliavimo priemonę. Paketas palaiko geresnes biznio modeliavimo galimybes, ypač unikalus biznio sąveikų modelis (BIM) ir populiarūs darbų sekos ir biznio objektų modeliai. 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.

3.2. *Microsoft Access*

Microsoft Access (toliau *MS Access*) yra viena populiariausių vidutinės apimties DBVS. Galima kurti tiek autonomines, tiek tinklines taikomasias programas, taip pat programas, skirtas dirbti sistemose „Klientas – serveris“. *MS Access* stabiliai dirba, turi daug funkcijų.

Jos angliško pavadinimo *Access* lietuviškas atitikmuo yra **kreiptis** (į duomenis). Ji tinka kurti DB, kurias sudaro dešimtys ar šimtai tūkstančių įrašų ir kuriomis naudojasi nedidelis vartotojų skaičius. *MS Access* dažniausiai naudojama saugoti ir apdoroti informaciją vidutinėse ir mažose įmonėse bei organizacijose. *MS Access* yra reliacinė DBVS: duomenys pateikiami kaip normalizuotų dvimačių lentelių rinkinys. DBVS *MS Access* pagrindinės funkcijos yra šios:

- ◆ DB struktūros kūrimas;
- ◆ DB užpildymas ir redagavimas;

- ◆ duomenų peržiūra, vaizdavimas vartotojui tinkama forma;
- ◆ navigacija DB;
- ◆ duomenų rikiavimas (rūšiavimas), filtravimas, paieška.

DBVS *MS Access* vykdo ir eilę kitų, pagalbinių, bet labai svarbių funkcijų: užtikrina duomenų korektiškumo ir vientisumo kontrolę, įgalina atlikti įvairius matematinius, statistinius skaičiavimus, leidžia įslaptinti tam tikrus duomenis ir uždrausti nesankcionuotą kreipti į juos. Be to, vartotojas, naudodamas DBVS priemones (makrokomandas, SQL ar *Visual Basic* kalbą) gali sukurti savąją meniu sistemą ir taikomas programas, skirtas specifiniams poreikiams tenkinti. Minėtoms funkcijoms atlikti *MS Access* turi specialius instrumentus - grafinius dizainerius, vedlius (meistrus), išraiškų konstruktorius, programų generatorius ir kt.

MS Access priemonėmis sukurta **reliacinė DB** išsaugoma faile, kuriam suteikiamas plėtinys **mdb**. Šiame faile saugomos ne tik dvimatės lentelės su duomenimis, bet ir kiti objektai:

Lentelės (Tables). Jose, kaip minėjome, saugomi duomenys. Kiekvienoje lentelėje nurodomi laukai ar jų rinkiniai, įgyjantys tik nepasikartojančias reikšmes. Vienas jų paskelbiamas pirminiu raktu, vienareikšmiškai apibrėžiančiu kiekvieną lentelės įrašą. Lentelė yra bazinis **MS** objektas. Kiti objektai yra išvestiniai ir gali būti sudaromi, kai sukurta bent viena lentelė.

Užklausos (Queries). Įgalina atrinkti duomenis pagal tam tikrus požymius (kriterijus) ir, jei reikia, juos modifikuoti ar pašalinti. Taigi užklausų dėka iš DB galima bet kuriuo momentu išrinkti reikalingą informaciją.

Formos (Forms). Palengvina įvesti, peržiūrėti ir redaguoti duomenis. *MS Access* priemonėmis vartotojas gali sukurti vaizdžias ir informatyvias formas.

Ataskaitos (Reports). Įgalina sukurti ir spausdinti įvairius dokumentus, pagal DB sukaupus duomenis. Juos galima rūšiuoti, grupuoti, atlikti skaičiavimus.

Puslapiai (Pages). Tai dialoginiai **WEB** puslapiai, kurie palaiko dinaminę ryšį su DB ir leidžia juose atvaizduoti bei redaguoti *MS Access* duomenis.

Makrokomandos (Macros). Naudojamos aprašyti veiksmų sekoms, kurios aktyvizuojamos, vartotojui išrinkus atitinkamą meniu punktą ar paspaudus mygtuką.

Moduliai (Modules). programuotojo sukurtos procedūros ir funkcijos, kurios rašomos *Visual Basic* kalba.

3.3. Programinės įrangos perdarymo inžinerija (re-engineering)

Sistemos perdarymo inžinerija - perstruktūrizavimas arba perrašymas tik dalies arba visos liktinės sistemos nekeičiant jos funkcionalumo.

Yra pritaikoma ten kur kelios bet ne visos posistemės didelės sistemos reikalauja dažno palaikymo

Perdarymo inžinerija reikalauja pastangų kad padaryti sistemą lengviau palaikoma. Sistemai gali reikėti perstruktūrizavimo arba pakeitimų dokumentacijoje (perdokumentavimo)

Sistema yra perdaroma šiais atvejais:

- Kai pakeitimai dažniausiai vykdomi tam tikroje sistemos dalyje, būtina perdaryti tą dalį
- Kuomet techninės arba programinės įrangos palaikymas pasensta (tampa nebevertotinu)
- Kuomet turime įrankius reikalingus perprojektavimui

Perdarymo privalumai :

1. Sumažėjusi rizika:

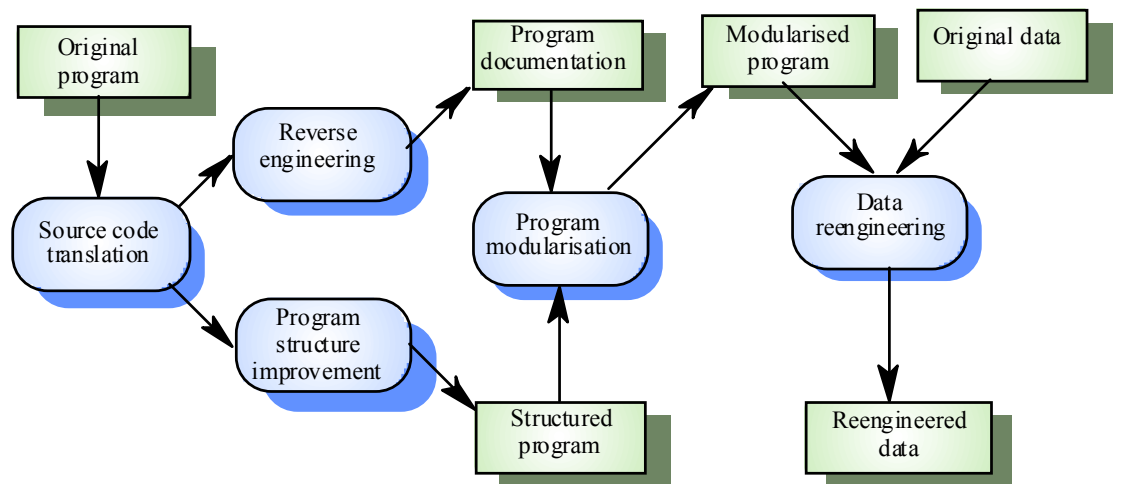
- Yra didelė rizika naujos programinės įrangos gamyboje. Gali iškilti gamybos, personalo ir specifikacijų problemų

2. Sumažėję kaštai:

- Perdarymo kaina dažniausiai yra žymiai mažesnė nei naujos programinės įrangos kūrimas

Biznio procesų perdarymas yra pagrįstas biznio procesų pertvarkymu darant juos atsakingesnius ir efektyvesnius; dažnai priklauso nuo naujų kompiuterinių sistemų, ir jų galimybių palaikyti pakeistus procesus; gali iššaukti programinės įrangos perdarymą kadangi liktinės sistemos skirtos palaikyti egzistuojančius procesus

Perdarymo procesas pavaizduotas 6 paveiksle:



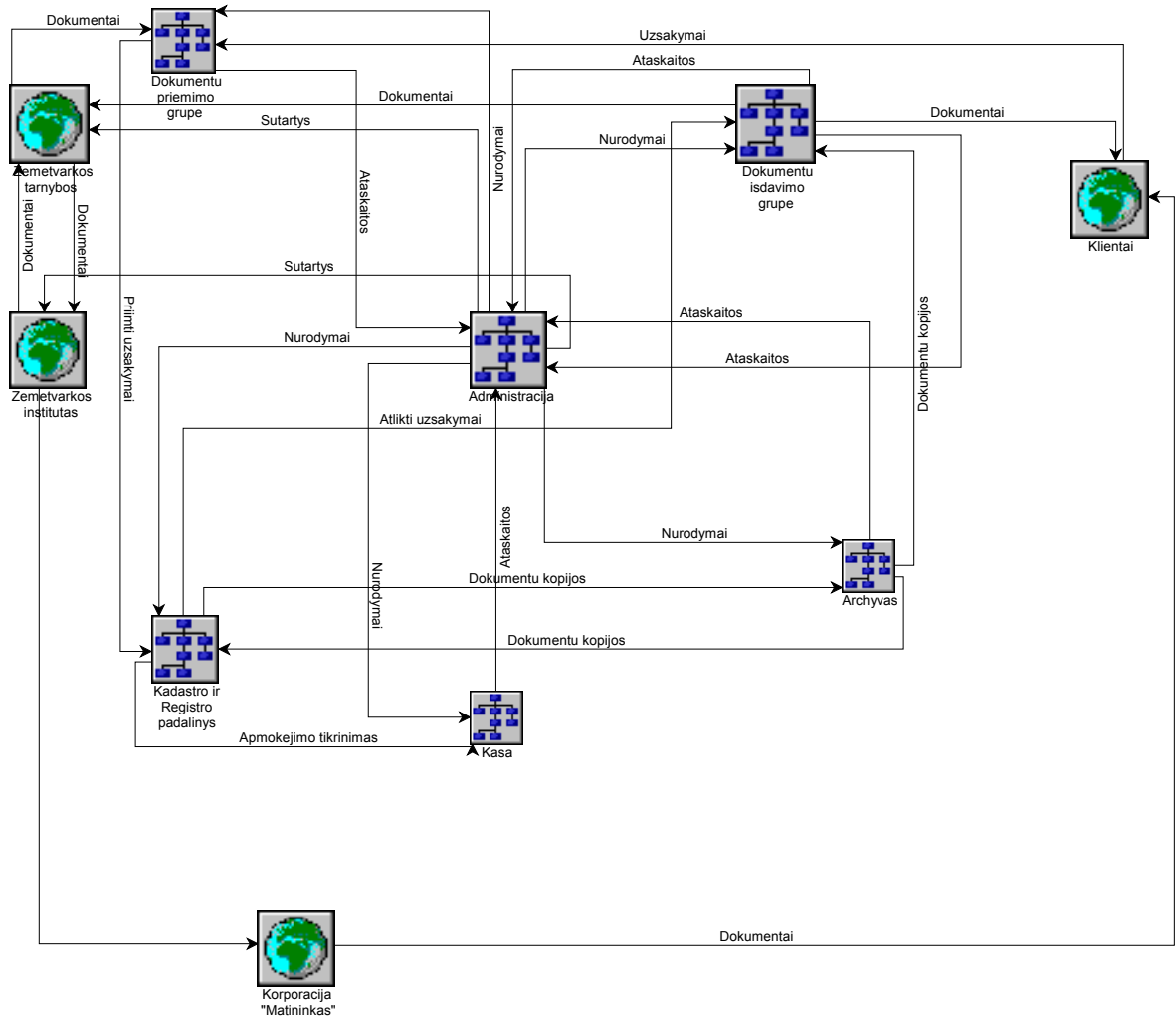
11.pav. Perdarymo procesas

4. Projektinė dalis

4.1. Projektuojamos IS aprašymas

4.1.1. Biznio sąveikų modelis

Biznio sąveikų modelis atvaizduoja vartotojo biznio veiklą strateginėje perspektyvoje, parodo sąveiką tarp vidinių organizacijos objektų ir išorinių organizacijų. Šiame modelyje yra modeliuojamos ir įvertinamos organizacijos objektų sąveikos ir ryšiai su vartotojais, tiekijais ir konkurentais.

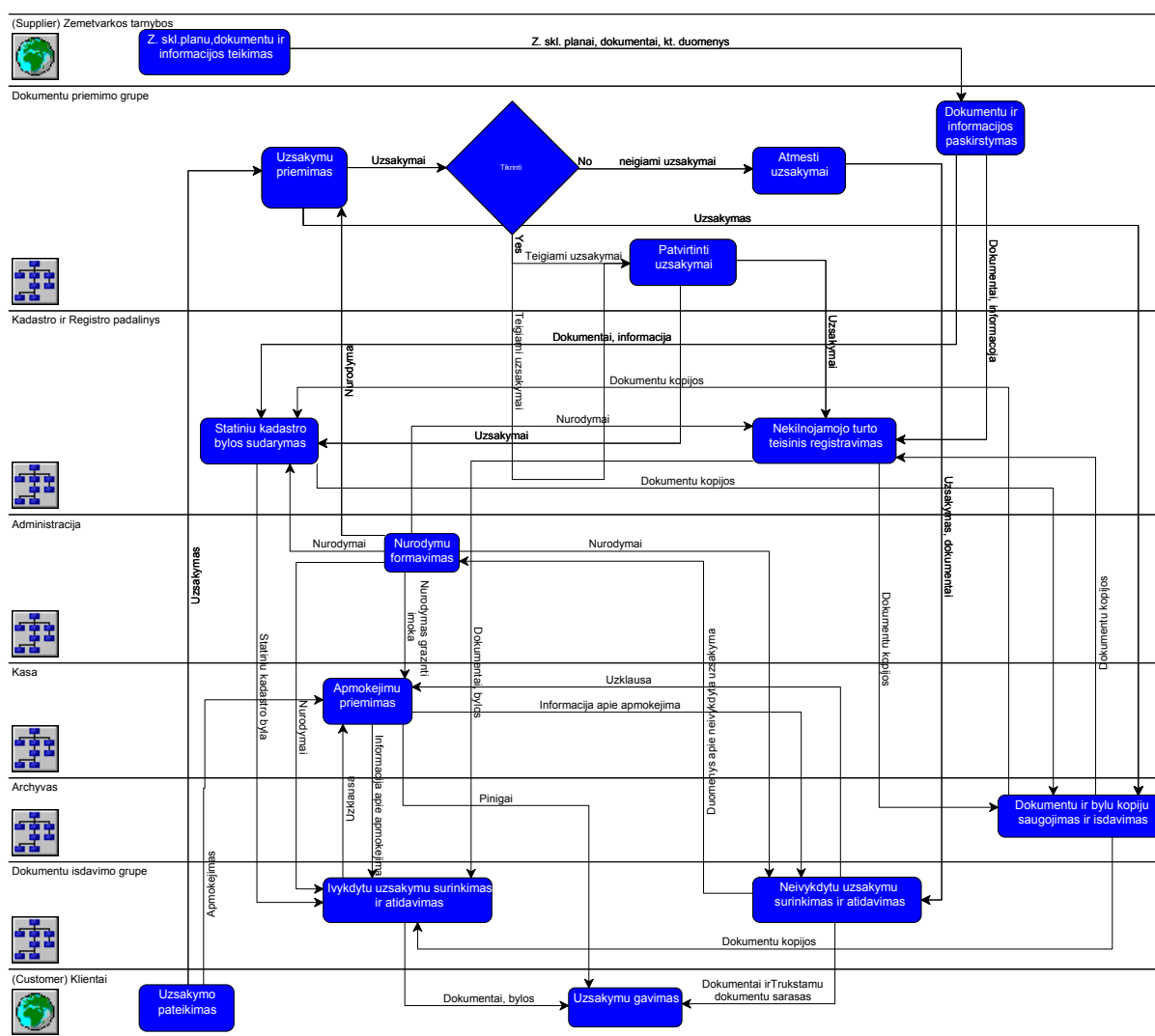


12.pav. Biznio sąveikų modelis

4.1.2. Darbų sekos modelis, pirmasis lygmuo

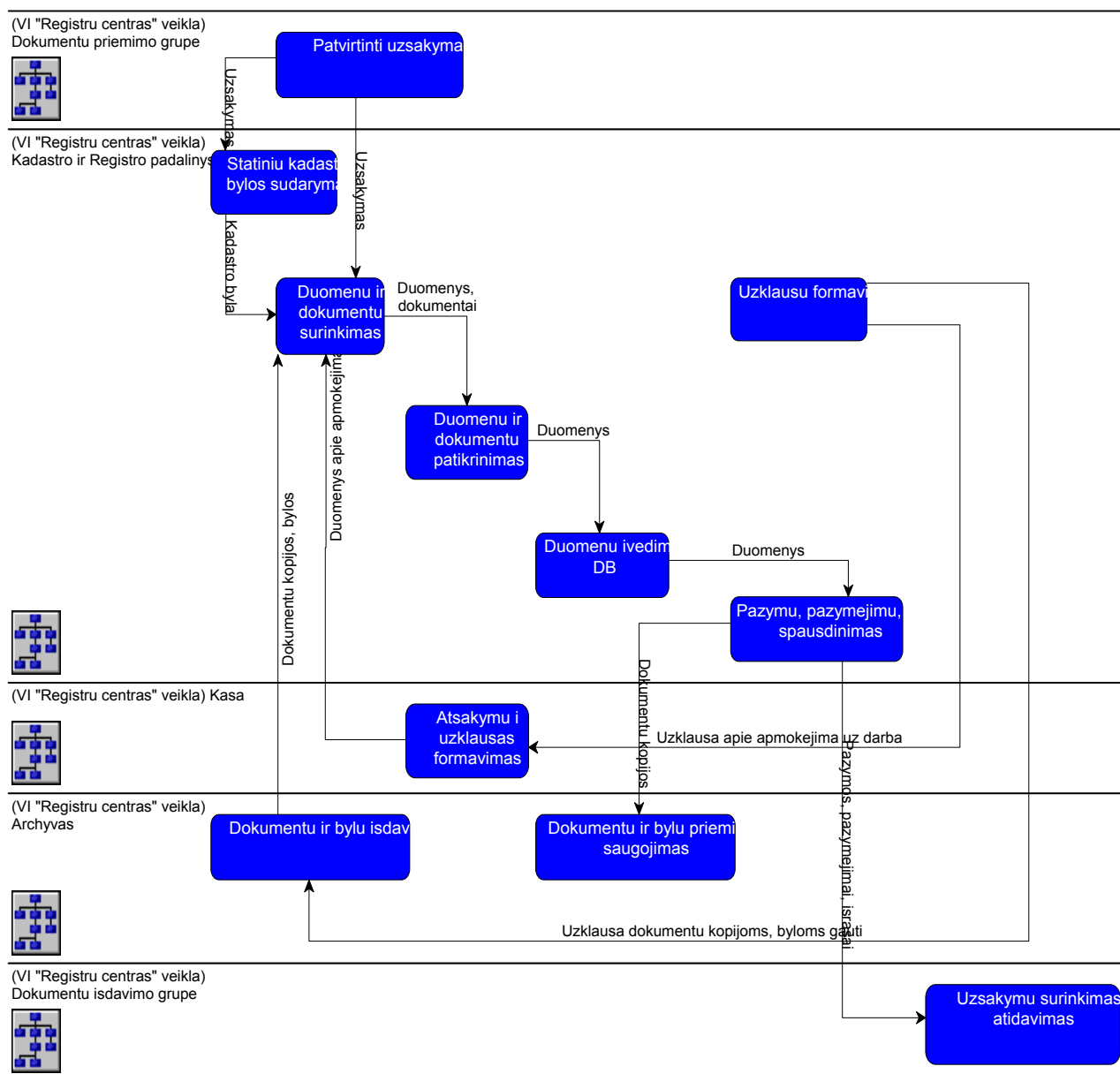
Darbų sekos modelis yra kuriamas per biznio sąveikų modelį. Naudojant darbų sekos modeliavimą galima sukurti labiau detalesnę veiklų modelį.

Šis modelis atvaizduoja biznio procesus išreiškiant juos veiklos komponentais ir darbų seka tarp tų veiklų. Darbų sekos modelis koncentruotas į darbų seką nuo biznio pradžios iki galo. Tai atspindi pagrindinis aukščiausio lygio modelis. Jei koks procesas ar darbas yra sudėtinis kitų darbų atžvilgiu, tai šiam procesui yra sukuriamas detalesnis žemesnio lygio darbų sekos modelis.



13.pav. Darbų sekos modelis, pirmasis lygmuo

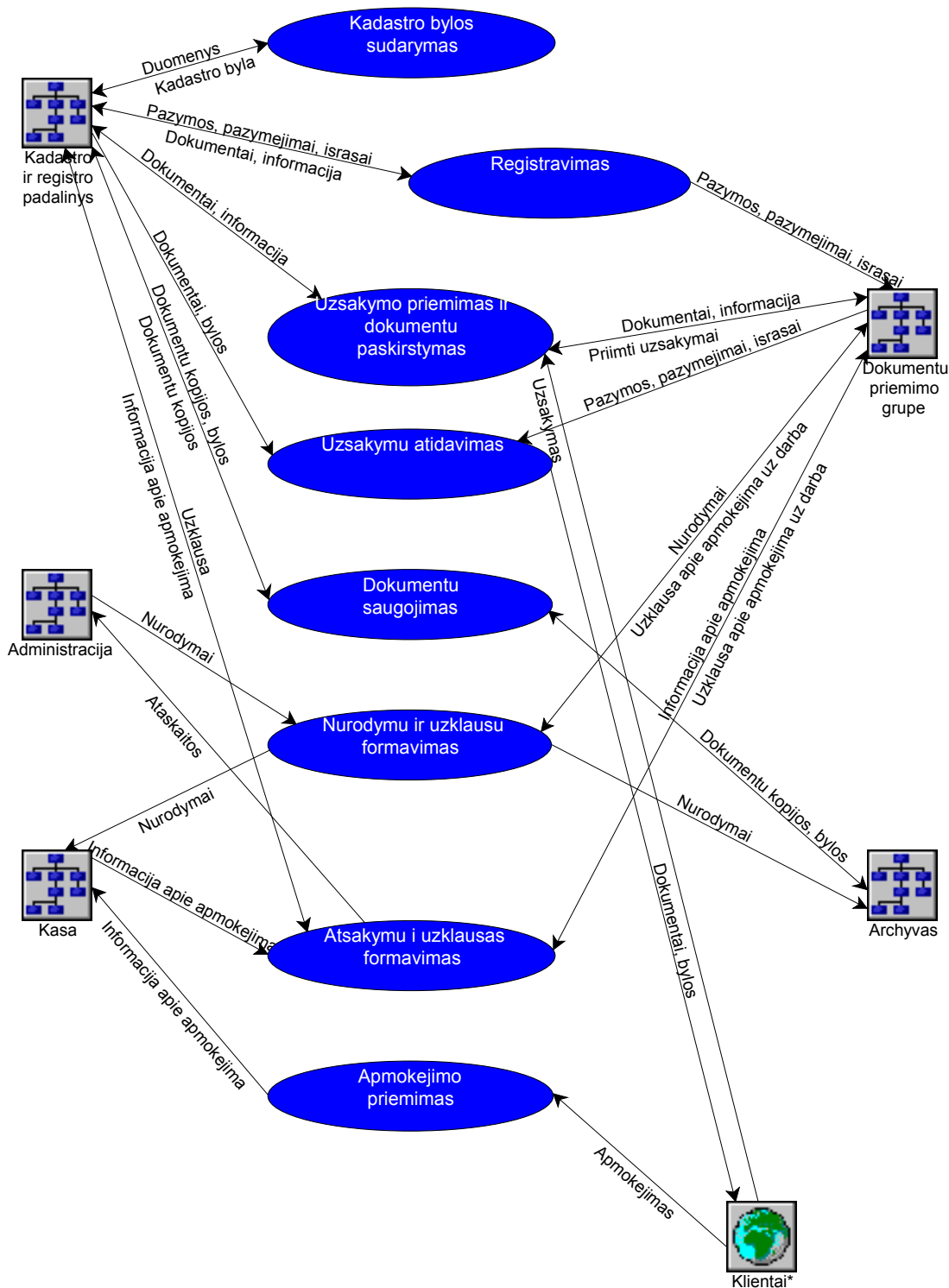
4.1.3. Darbų sekos modelis, antrasis lygmuo



14.pav. Darbų sekos modelis, antrasis lygmuo

4.1.4. Panaudojimo atvejų modelis

Biznio panaudojimo atvejų diagrama apibrėžia ir analizuoja svarbesnes biznio sferas, biznio procesų ar veiklos sąveikas tiek orfžganizacijos viduje tiek ir išorėje. Tai padeda apžvelgti kaip dauguma biznio zonų yra susijusios su išorinėmis biznio esybėmis, veiklomis ir sistemomis.



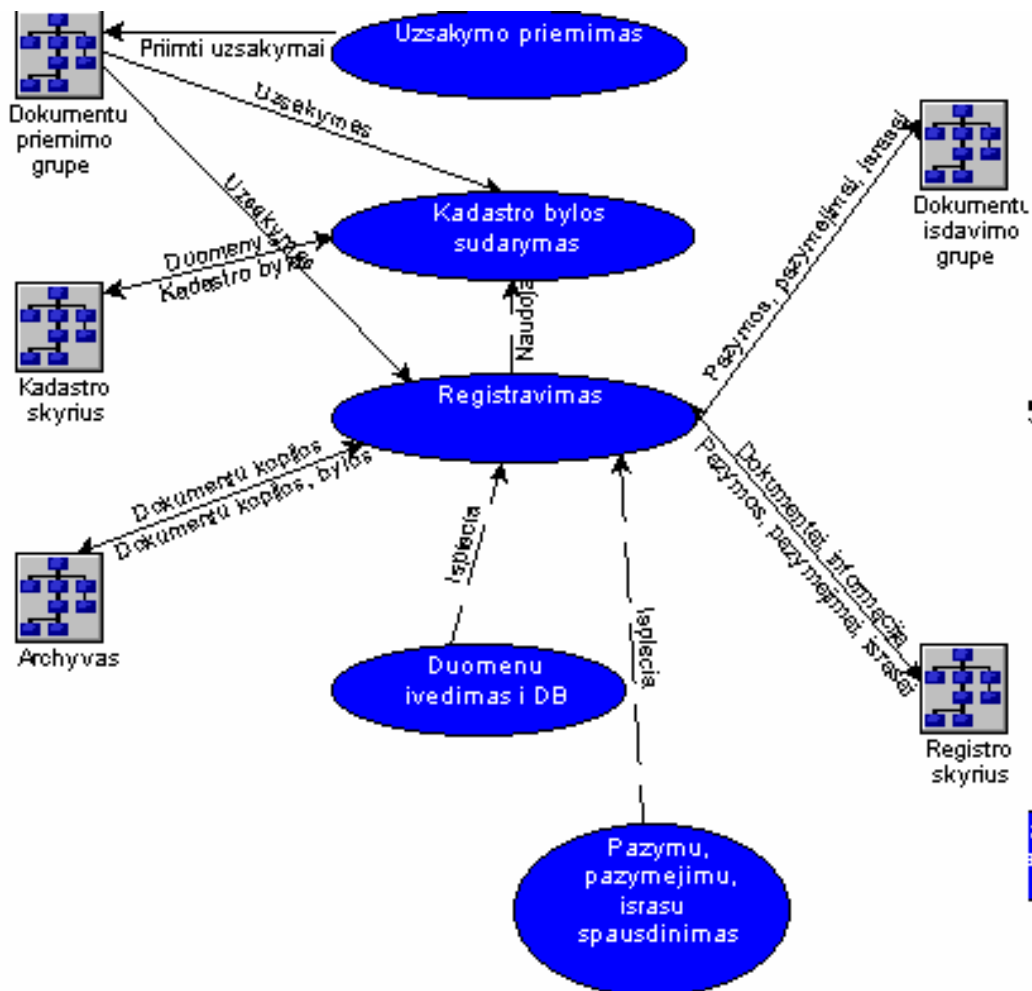
15.pav. Panaudojimo atvejų modelis

3.lentelė.Pagrindinių uždavinių aprašymas:

Uždavinys	Aprašymas
Žemės sklypų registravimo uždavinys	<p>Registro grupės darbuotojai gauna informaciją iš priėmimo skyriaus ir žemėtvarkos skyriaus. Į registro duomenų bazę jie įveda tokius duomenis: Žemės sklypo plotas, žemės ūkio naudmenų našumo balas, žemės sklypo vertė, žemės sklypo adresas, žemės sklypo savininkas – vardas, pavardė, asmens kodas, juridinio dokumento pavadinimas, data ir numeris. Registruodami naują žemės sklypą, registro darbuotojai sukuria šio sklypo unikalų numerį. Šis numeris sudarytas taip: Kadastrinės vietovės numeris + kadastrinio bloko numeris + sklypo numeris bloke. Informaciją apie vietovės ir bloko numerius registro skyrius gauna iš rajono žemėtvarkos skyriaus. Sklypo numerį bloke suteikia registro skyriaus darbuotojas įveda jį į duomenų bazę ir užrašo ant žemės sklypo plano. Iš statinių registro paimami duomenys apie statinius, esančius ant ž. sklypo Registro skyriaus darbuotojai, naudodamiesi duomenų baze, rengia reikalingus dokumentus (pažymas sandoriui, pažymėjimus ir kt.)</p>
Statinių registravimo uždavinys	<p>Šio registro skyriaus darbuotojai informaciją gauna iš priėmimo skyriaus, žemės sklypų registro skyriaus ir iš kadastro skyriaus. Į duomenų bazę jie įveda duomenis apie statinių teisinę priklausomybę (savininkas – vardas, pavardė, asmens kodas, juridinio dokumento pavadinimas, data ir numeris), duomenis apie žemės sklypą (ž. sklypo kadastrinį numerį). Suteikia statiniui unikalų numerį. Skyriaus darbuotojai, naudodamiesi duomenų baze, rengia reikalingus dokumentus (pažymas sandoriui, pažymėjimus ir kt.)</p>
Kadastrinių matavimų uždavinys	<p>Kadastro skyriaus darbuotojai, atlikę matavimus, braižo pastatų planus, skaičiuoja plotus, vertes. Brėžiniai braižomi, naudojant <i>AutoCad</i> programą. Statinių vertės skaičiuojamos, įvertinant statybos metus, pastato nusidėvėjimą, rinkos koeficientus konkrečiai vietai, medžiagas, panaudotas statyboje. Šiuos duomenis naudoja statinių registro skyrius.</p>
Užsakymų vykdymo uždavinys	<p>Klientai užsakymus gali pateikti dviem būdais: 1. atėję į klientų aptarnavimo skyrių, 2. per internetą. Jei klientas gali prieiti prie interneto, jis gali pateikti užsakymą internetu per Kadastro ir Registro valstybės įmonės interneto</p>

	svetainę. Ten jam iš karto yra pateikiamas meniu, ką klientas gali užsisakyti. Išsirinkęs, klientas turi pateikti informaciją apie save, kad būtų galima pradėti vykdyti užsakymą.
--	--

Use Case modelis, detalizuojantis registravimo procesą:



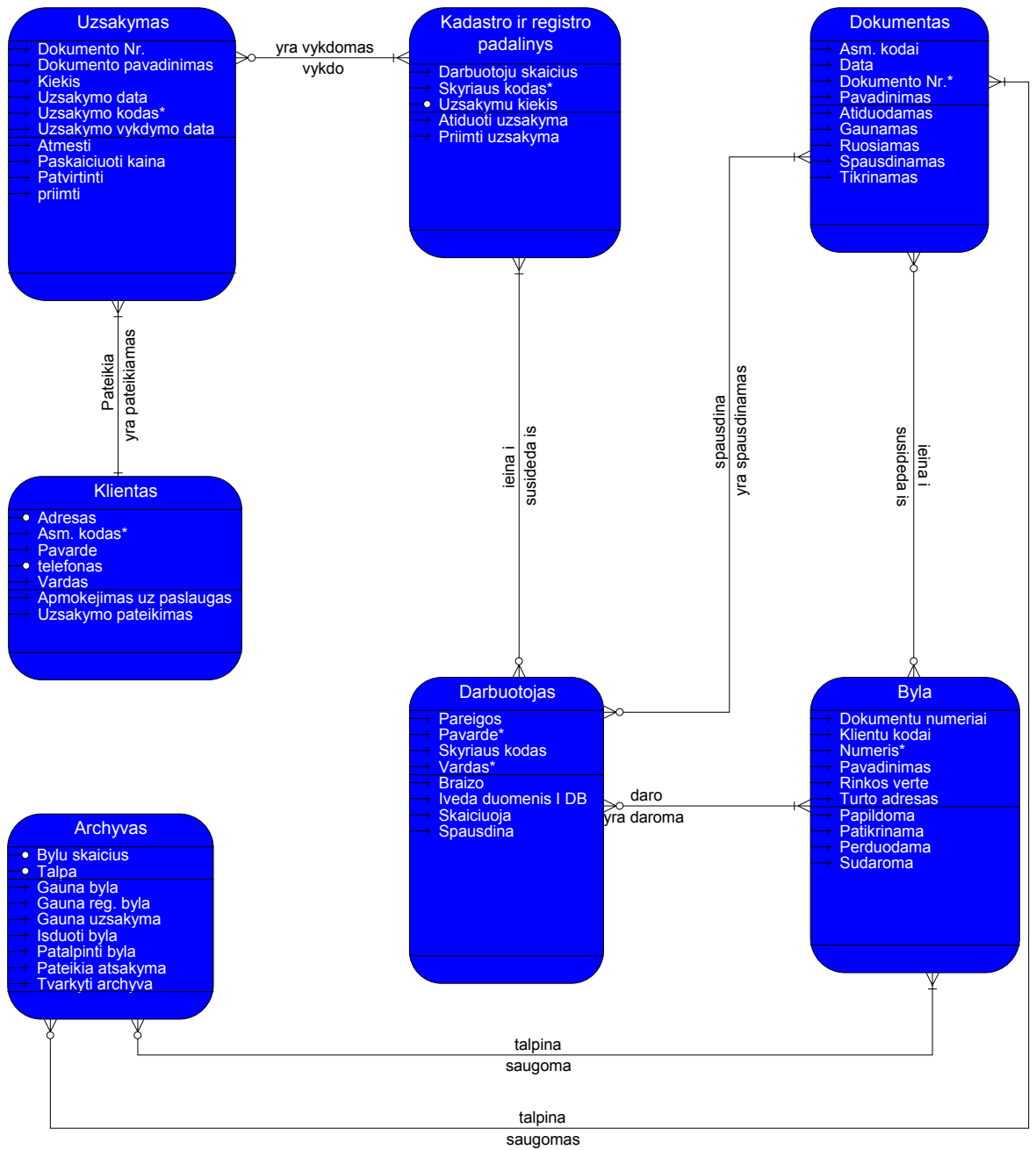
16.pav. Use Case modelis, detalizuojantis registravimo procesą.

4.lentelė.

Uždavinys	Aprašymas
Žemės sklypų registravimo uždavinys	Registro grupės darbuotojai gauna informaciją iš priėmimo skyriaus: Žemės sklypo plotas, žemės ūkio naudmenų našumo balas, žemės sklypo vertė, žemės sklypo adresas, žemės sklypo savininkas – vardas, pavardė, asmens kodas, juridinio dokumento pavadinimas, data ir numeris. Registruodami naują žemės sklypą, registro darbuotojai sukuria šio sklypo unikalų numerį. Šis numeris sudarytas taip: Kadastrinės vietovės numeris + kadastrinio bloko numeris + sklypo numeris bloke. Iš statinių registro paimami duomenys apie statinius, esančius ant ž. sklypo. Registro skyriaus darbuotojai, naudodamiesi duomenų baze, rengia reikalingus dokumentus (pažymas sandoriui, pažymėjimus ir kt.).
Duomenų įvedimas į DB	Į registro duomenų bazę įvedami tokie duomenys: Žemės sklypo plotas, žemės ūkio naudmenų našumo balas, žemės sklypo vertė, žemės sklypo adresas, žemės sklypo savininkas – vardas, pavardė, asmens kodas, juridinio dokumento pavadinimas, data ir numeris. Jei yra statinių, įvedama: statinio pavadinimas, pažymėjimas plane, paskirtis.
Pažymų, pažymėjimų, išrašų spausdinimas	Iš duomenų bazėje turimų duomenų suformuojami informaciniai išrašai ir atspausdinami.

4.1.5. Objektų modelis

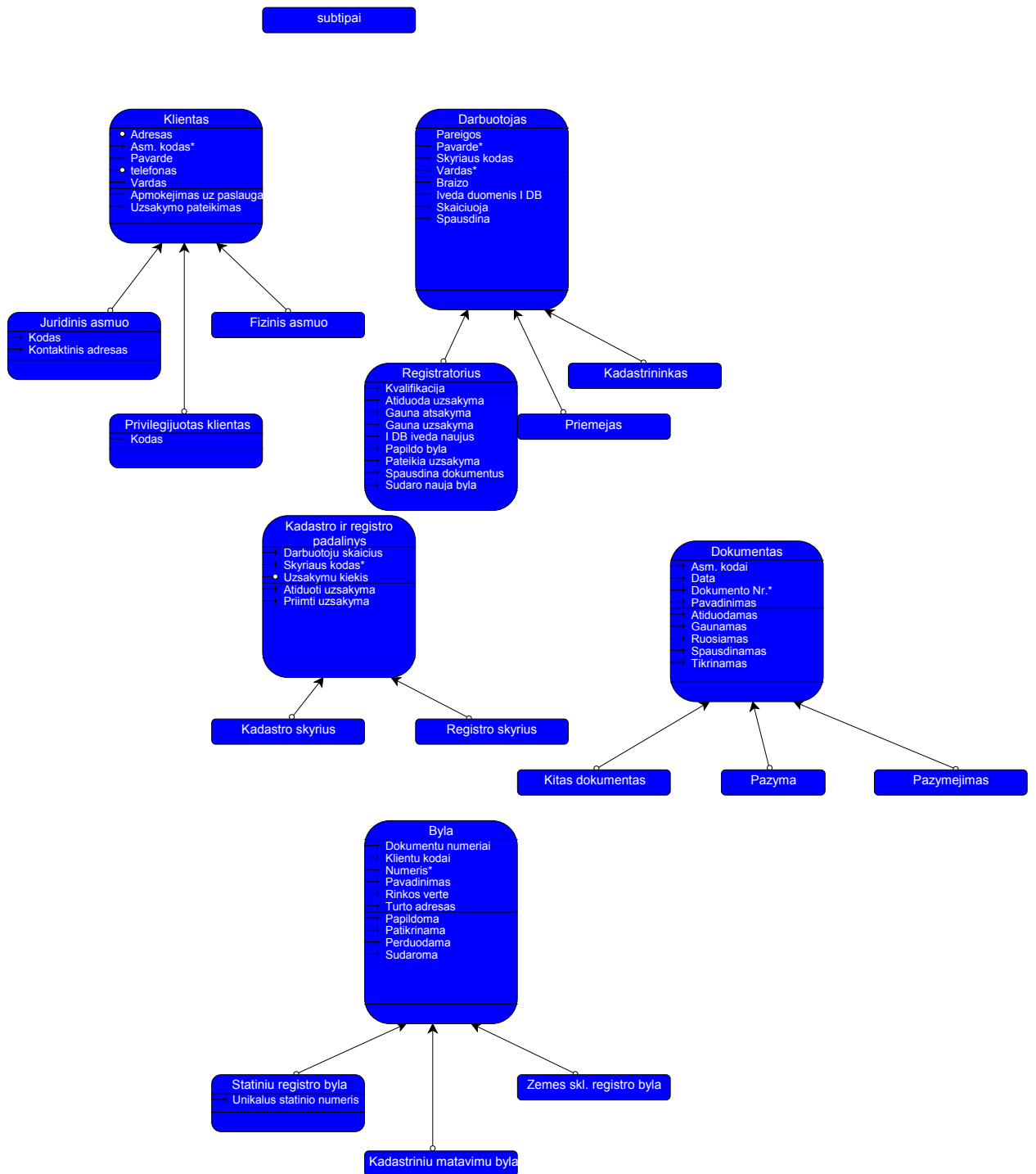
Biznio objektų modelis leidžia vartotojui apibrėžti detalią biznio objektų informaciją (savybes). Objektų modelis sudaro pagrindinį biznio objektų komponentų aprašą. Objektų modelis parodo kiekvieno objekto savybes : atributus ir metodus arba funkcijas.



17.pav. Objektų modelis

4.1.6.Subtipų modelis

Objektų tipai toliau apibrėžiami subtipų modelyje esančiuose objektuose.



18.pav.Subtipų modelis

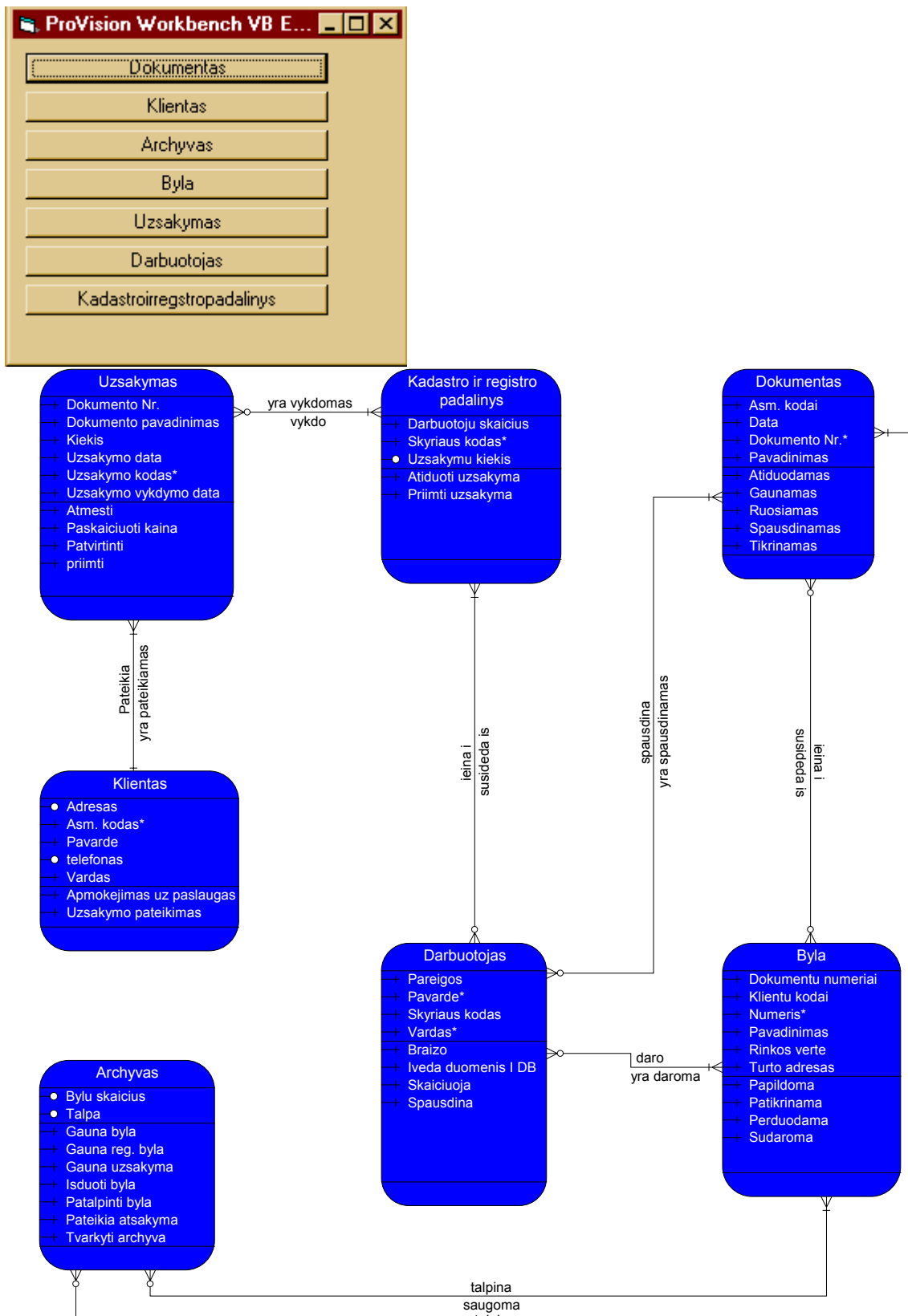
4.1.7. IS prototipo generavimo rezultatai

Vartotojo sąsajos langų prototipai

Vartotojo sąsajos langų prototipai sugeneruojami *Provision Workbench 3.1* paketu pagal šiuo paketu sudarytas IS specifikacijas.

Pagrindinis šaltinis generuojamiems vartotojo sąsajos langų prototipams – biznio objektų modelyje apibrėžiami objektai, jų metodai, atributai bei ryšiai tarp objektų .

19 paveiksle pateikiamas sąsajos lango prototipas, kuris yra pagrindinis sąsajos langas – jame yra visų biznio objektų pavadinimai, pateikiami ant formos mygtukų. Pasirinkus reikiamą mygtuką – iškviečiamas konkretus tą biznio objektą aprašantis sąsajos langas.



19.pav. Pagrindinis sąsajos langas

Toliau pateikiami kiti vartotojų sąsajos langų prototipai :

The screenshot shows a window titled "Dokumentas" with a red title bar. It contains four input fields on the left: "Asm. kodai", "Pavadinimas", "Data", and "Dokumento Nr.". To the right of these fields are three buttons: "saugomas_Archyvas", "ieina i_Byla", and "yra spausdinamas_Darbuotojas". At the bottom of the window, there is a row of five buttons: "Gaunamas", "Ruosiamas", "Spausdinamas", "Atiduodamas", and "Tikrinamas".

The screenshot shows a window titled "Klientas" with a red title bar. It contains five input fields on the left: "Asm. kodas", "Vardas", "telefonas", "Adresas", and "Pavarde". To the right of these fields is one button: "Pateikia_Uzsakymas". At the bottom of the window, there is a row of two buttons: "Uzsakymo pateikimas" and "Apmokejimas uz paslaugas".

Byla

Dokumentu numeriai	<input type="text"/>	yra daroma_Darbuotojas
Klientu kodai	<input type="text"/>	saugoma_Archyvas
Rinkos verte	<input type="text"/>	susideda is_Dokumentas
Turto adresas	<input type="text"/>	
Numeris	<input type="text"/>	
Pavadinimas	<input type="text"/>	

Uzasakymas

Uzasakymo kodas	<input type="text"/>	yra pateikiamas_Klientas
Dokumento Nr.	<input type="text"/>	yra vykdomas_Kadastro ir registro padalinys
Kiekis	<input type="text"/>	
Dokumento pavadinimas	<input type="text"/>	
Uzasakymo data	<input type="text"/>	
Uzasakymo vykdymo data	<input type="text"/>	

20.pav. Vartotojų sąsajos langų prototipai

4.2. Reikalavimų dokumentas

Reikalavimų išgavimo planas

Reikalavimų išgavimui naudosime tokį planą:

- Vartotojų apklausa, bei panašaus pobūdžio programinės įrangos privalumų ir trūkumų analizė.
- Nefunkcinių reikalavimų nustatymas.
- Prototipų kūrimas.

Vartotojo reikalavimai

- Palengvinti naujų duomenų įvedimą į DB
- Palengvinti duomenų paiešką duomenų bazėje
- Sutrumpinti duomenų įvedimo ir paieškos laiką
- Prie tekstinių duomenų pateikti grafinę informaciją
- Suteikti priėjimą prie DB ir kitų įmonės skyrių darbuotojams
- IS turi būti suderinta su Windows 95, Windows 98, Windows NT ir Windows 2000 operacinėmis sistemomis.
- Naujo Nekilnojamojo turto vieneto (žemės sklypo) registravimas (tekstinių duomenų apie naujai suformuotą žemės sklypą įvedimas)
- Nuosavybės, nuomos ir kt. daiktinių teisių žemės sklypui registravimas.
- Pažymėjimo apie Nekilnojamojo turto registre įregistruotą žemės sklypą ir teises į jį spausdinimas.
- Pažymos sandoriui spausdinimas.
- Turto paieška pagal :
 1. registro numerį,
 2. pavardę, asmens kodą,
 3. unikalų žemės sklypo kodą,
 4. turto adresą.
- Apmokymas. Užsakovas reikalauja apmokyti darbuotojus dirbti su nauja programine įranga. Taip pat vartotojai pagalbą galės gauti ir elektroniniu paštu : hakari@kada.lt

Nefunkciniai reikalavimai

Išnagrinėjus panašaus tipo programinę įrangą bei jos reikalavimus sistemoms, planuojami programinės bei aparatūrinės įrangos reikalavimai būtų tokie:

Aparatūrinė įranga:

IBM PC.

Programinė įranga:

- Programos greitis priklausys nuo turimos techninės įrangos.
- Planuojamas programos dydis: iki 50Mb.
- OS: Windows 9x, ME, NT, 2000, XP.

Viena iš labiausiai paplitusių operacinių sistemų.

- Duomenų bazės sukurtos SQL kalba pagal reliacinių duomenų bazių standartus. Duomenų bazių valdymas turi vykti per ODBC interfeisą. Grafinė informaciją saugoma BMP, JPG, JPEG formatais
- Minimalūs reikalavimai aparatūrai:

IBM PC arba juos atitinkantys kompiuteriai su sekančiomis charakteristikomis :

- Intel Pentium 166MHz procesorius
- 32MB SDRAM
- 4MB kietasis diskas

Prototipai

Programinės įrangos projektavimo procese labai svarbus yra prototipų vaidmuo. Prototipai padeda išsiaiškinti papildomus sistemos funkcinius ir nefunkcinius reikalavimus, pateikia pradinį sistemos vaizdą. Analizuojant egzistuojančius IS kūrimo būdus (*ProVision Workbench paketa*, IS kūrimą, naudojant komponentinį modelį ir kt.), buvo sukurti keli prototipai papildomų reikalavimų išgavimui, bei vartotojo sąsajos pobūdžio nustatymui.

Sistemos reikalavimų specifikacija

Įvertinus vartotojų reikalavimus bei įrangos analizės rezultatus, gaunami reikalavimai sistemai.

Interfeiso reikalavimų specifikacija

1. Vartotojo sąsaja: duomenų bazė turi specialią vartotojo sąsają. Vietinis tinklas turi standartinę Windows sąsają

2. GUI – grafinė varotojo sąsaja (menu scenarijus, langai): realizuota pagrindinio menu pagalba, iš kurio galima pereiti į smulkesnius menu ir pasirinkti reikiamas atlikti operacijas:

- Suformuoti reikiamus sąrašus,
- Papildyti sąrašus,
- Gauti apdorotą informaciją, reikalingą tolimesniuose veiksmuose bei nustatyti ankstesnio etapo atliktus veiksmus,
- Peržiūrėti duomenis,
- Formuoti ataskaitas.

3. Komandos realizuojamos menu pagalba ir mygtukais.

4. Atlikus operaciją galima grįžti į pradinį menu langą.

5. Diagnostika (klaidų pranešimų aprašymai): jei klaida gali būti nustatyta sistemos vartotojui pranešama žinutėmis apie klaidingų duomenų įvedimą.

6. Techninės įrangos sąsajos – IS turi sąsajas, leidžiančias pasinaudoti bendru tinkle spausdintuvu bei techniškai užtikrintas ryšys su Internetu.

7. Komunikacinės sąsajos: tinkle turi būti įdiegta programinė įranga, leidžianti skyrių darbuotojams komunikuoti realiame laike, tai pagreitina informacijos pasikeitimą tarp organizacijos darbuotojų, leidžia greičiau spręsti iškilusias problemas.

8. Programinės įrangos sąsajos: turi būti darbuotojų kompiuteriuose instaliuota programinė įranga, leidžianti naudotis Internetu, kitaip nebus išnaudojamos visos duomenų bazės funkcijos ir galimybės.

9. Pranešimų formulavimas. Pranešimai formuluojami lietuvių kalba, jie turi būti trumpi, suprantami kiekvienam IS vartotojui, neturinčiam specialaus pasiruošimo dirbti kompiuteriu.

10. Operacinės sistemos naudojimas. IS naudoja funkcijas, bendras operaciniams sistemoms Windows 95, Windows 98, Windows NT ir Windows 2000.

11. Duomenų bazių interfeisas. Duomenų bazės turi būti sukurtos SQL kalba pagal reliacinių duomenų bazių standartus. Duomenų bazių valdymas turi vykti per ODBC interfeisą. Grafinė informaciją saugoma BMP, JPG, JPEG formatais

12. Kompiuterinio ryšio interfeisas ir darbas kompiuterių tinkle. Kompiuterinio ryšio interfeisas organizuojamas ETHERNET tinklo standarto IEE 802.3 pagrindu, naudojant TCP/IP protokolų steką. Darbas kompiuterių tinkle organizuojamas priklausomai nuo esamos operacinės sistemos.

Tikslumo reikalavimų specifikacija

1. Duomenų vaizdavimo tikslumas.

Simbolinio tipo duomenys neturi viršyti 200 ženklų.

2. Leistinos paklaidos.

Nėra

Našumo reikalavimų specifikacija

1. Atminties ir įrenginių naudojimas.

Reikalavimuose nurodytas funkcijas korektiškai vykdyti terminaluose nenaudoti daugiau nei 8 MB operatyvinės atminties. Operatyvinės atminties naudojimas serveryje neturi viršyti 32 MB operatyvinės atminties.

Duomenų bazės serveryje neturi užimti daugiau 2 GB išorinės atminties.

IS turi pajėgti reaguoti į 10 užklausų vienu metu.

2. Reakcijos laikas.

IS turi pateikti atsakymą į užklausą ne daugiau kaip per 3 sekundes.

Duomenų bazių pildymas neturi viršyti 3 sekundžių.

Sistemos patikimumo specifikacija

1. Patikimumo matavimas.

IS funkcionalumo praradimo trukmė neturi viršyti dviejų minučių per vieną valandą.

2. Klaidų ir ypatingų situacijų apdorojimas.

Ypatingos situacijos arba klaidos atveju IS turi išduoti klaidos pranešimą.

3. Trūkių apdorojimas ir sistemos veikimo atkūrimas.

IS turi būti sukurta iš atskirų modulių. Įvykus klaidai viename iš modulių, likę neturi prarasti funkcionalumo. Įvykus trikiams, IS veiklos atnaujinimas vykdomas sistemos administratoriaus pagalba.

Kiti nefunkciniai sistemos atributai

1. Saugumas - būtina apsauga nuo išorinių įsilaužėlių firewall ("ugnies siena").

Paskiriami organizacijos darbuotojams slaptažodžiai individualiam darbui.

2. Patikimumas - būtinas duomenų kopijų kūrimas.

3. Reikalavimai IS palaikymui - konsultacijos pageidavimas iš informacijos sistemų specialisto dėl iškilusių neaiškumų.

4. IS išplėtimo reikalavimai – galimybė susijungti į bendrą organizacijos IS, apjungiant ir kitų skyrių operacijas.

5. Pakartotino panaudojimo reikalavimai – programą galima panaudoti pakartotinai tik sėkmingai įdiegus programą iš naujo išnagrinėjus, ar nepasikeitė organizacijos struktūra ir duomenų srautai.

6. Taikomųjų programų suderinamumas – reikalingas suderinimas su Microsoft ir kita programine įranga.

7. Reikalavimai servisui - kas tam tikrą laiko tarpą turi būti atliekamas sistemos saugumo ir stabilumo patikrinimas.

Sistemos aptarnavimo ir priežiūros reikalavimų specifikacija

IS administravimui naudoti standartinius duomenų bazių valdymo bei lokalaus tinklo administravimo priemones.

Sistemos modernizavimo reikalavimų specifikacija

1. Reikalavimai tarpinei programinei įrangai.

Nėra.

2. Perkeliamumo reikalavimai.

IS perkeliamumas į kitą kompiuterinę platformą nevykdomas.

IS turi būti perkeliama į naujesnes ar patobulintas Windows operacinių sistemų versijas. Į kitas operacines sistemas perkeliamumas nevykdomas.

Sistemos saugos reikalavimų specifikacija

1. Vartotojų registracija ir tapatybės nustatymas.

Vartotojas turi pateikti IS savo identifikacijos vardą bei slaptažodį. Jeigu vardas ar slaptažodis nėra teisingas, IS turi pateikti pranešimą, prašantį pakartoti vardo ir slaptažodžio įvedimą.

2. Vartotojų klasės ir teisės.

Vartotojai skirstomi į tris klases:

- *“Full Control”* (Administratorius)
- *“Read/Write/Modify All”*
- *“Read Only”*

“Read Only” – turi teisę pateikti užklausas IS apie ž. sklypus, savininkus ir kt., peržiūrėti informaciją bei ją spausdinti;

“Read/Write/Modify All” – turi teisę peržiūrėti ir koreguoti DB

“Full Control” (Administratorius) – turi visas aukščiau išvardintas teises bei vykdyti IS priežiūrą.

Operacinis scenarijus (sistemos veiksmų scenarijus iš vartotojo pozicijų)

Vartotojui sukurtas pagrindinis meniu. Duomenų įvedimas atliekamas lentelėms sukurtomis vaizdžiomis formomis, turinčiomis papildomų atributų: mygtukų, iškrentančių sąrašų su galimomis laukų reikšmėmis ir kt. Formų langai atitinkamai apipavidalinti, laukų reikšmės jame išdėstomos laisva tvarka ir vaizduojamos vartotojui patogiu formatu. Duomenų įvedimui skirtos formos, sudarytos iš vienos ar kelių DB lentelių. Rezultatai (duomenys) vartotojo pageidavimu gali būti spausdinami ataskaitose.

4.3. Projekto grafikų ir išlaidų planas

Projekto išlaidos:

Projekto išlaidos yra minimalios, kadangi projektas vykdomas kaip magistrinis – mokslinis darbas. Daugumoje projekto išlaidas sudaro laikas skirtas projektui kurti bei programai rašyti.

Projekto grafikas:

5.lentelė. Projekto grafikas

Kūrimo stadija	Stadijos pabaigos data	Pristatymai	Pristatymo pabaigos data
Planavimas	2002 11 16	Projekto planas	2002 11 15
Reikalavimų apibrėžimas (nustatymas)	2002 12 12	Reikalavimų specifikacijos projektas Sistemos specifikacijos projektas Sistemos testavimo planas Reikalavimų specifikacija (galutinė)	2002 12 02 2002 12 02 2002 12 09 2002 12 12
Projektavimas	2003 04 30	Darbuotojų apmokymo plano projektas Programos ir duomenų bazės specifikacija Sistemos specifikacija (galutinė)	2003 02 01 2003 03 01 2003 04 30
Programavimas	2003 06 30	Programinė įranga Sistemos testavimo planas Instrukcijos vartotojams Eksploatacijos dokumentacija	2003 05 15 2003 05 25 2003 06 15 2003 06 30
Integravimas ir testavimas	04/15/00	Testo ataskaita Apmokymo planas (galutinis) Vartotojo instrukcijos (galutinės)	2003 07 30 2003 08 08 2003 08 12
Instaliavimas	2003 10 15	Aptarnavimo (eksploatacijos) planas Diegimo testavimo ataskaita	2003 09 30 2003 10 15

Kūrimo stadija	Stadijos pabaigos data	Pristatymai	Pristatymo pabaigos data

4.4. Rizikos mažinimas, kontrolė ir valdymas

Rizikos mažinimo, kontrolės ir valdymo apžvalga

Reikia siekti, kad programinė įranga būtų be defektų ir klaidų, tačiau kartais to pasiekti yra labai sunku arba ir neįmanoma. Todėl reikia turėti rizikos valdymo planą, kad būtų galima numatyti riziką, kuri gali įtakoti programinės įrangos kūrimą ir vystymą. Tikslas yra numatyti riziką, kad būtų galima kontroliuoti ir valdyti ją.

Norint numatyti visus galimus rizikos faktorius, klientas ir programinės įrangos kūrėjas turi glaudžiai bendradarbiauti. Svarbu, kad užsakovas daugiau savo laiko bendradarbiavimui skirtų pradinėse projektavimo fazėse. Jei užsakovas nusprendžia padaryti pakeitimus programinėje įrangoje ar reikalavimuose, tai turi didelės įtakos PĮ kūrimui ir vystymui.

Rizikos valdymo organizacinis vaidmuo

Kiekvienas, kuris yra susijęs su kuriama PĮ, turi prisiimti atsakomybę dėl rizikos valdymo. Jei kiekvienas, kuris dalyvauja PĮ kūrime, atkreips ypatingą dėmesį į visas PĮ kūrimo detales (ypač pradinėse stadijose), tai daugelio rizikos faktorių gali būti išvengta.

- Kuriant PĮ galima išvengti rizikos dar kartą patikrinant PĮ kūrimo grafiką, produkto apimtį, skaičiavimus (kainos, laiko).
- Užsakovas gali padėti išvengti rizikos pateikdamas visą būtiną informaciją ankstyvosiose kūrimo stadijose.
- Rangovas gali išvengti rizikos, surinkdamas visą informaciją apie jiems pateiktą ir prieinamą įrangą.
- Užsakovas gali išvengti rizikos padarydamas visus veiklos pakeitimus prieš reikalavimų programinei įrangai pateikimą.

Rizikos aprašymas

Ši dalis aprašo riziką, kuri gali atsirasti šio projekto kūrimo metu.

Rizikos tipų aprašymas

Poveikio įmonės veiklai rizika:

Tai yra rizika susijusi su tuo ar kuriama sistema atitiks kliento reikalavimus ir ar teigiamai paveiks kompiuterizuojamos veiklos sritį.

Kliento rizika:

Ar užsakovas bus pakankamai suinteresuotas padėti kuriant programą. Jei užsakovas nebus suinteresuotas, nepakankamai gerai ir tiksliai pateiks reikalavimus, tai tada nepavyks sukurti kokybiškos programinės įrangos.

Kūrimo rizika:

Jei klientas nepateiks tikslius reikalavimus, sistemos kūrimas nepavyks. Užsakovas turi skirti pakankamai laiko ir resursų programinės įrangos kūrėjams. Jei visi būtini resursai nebus pateikti, PĮ kūrėjų šansai sukurti gerą produktą stipriai sumažėja.

Darbuotojų rizika:

Čia sugebėjimas sukurti veikiančią sistemą visiškai priklauso nuo personalo kvalifikacijos ir patirties. Jei darbuotojai nėra pakankamai kvalifikuoti ar neturi pakankamai patirties dirbti su reikalaujama programavimo technologija, tai darbų atlikimo datos bus nukeliamos vėlesniam laikui, kol visas projektas netilps į užsibrėžtus laiko apribojimus. Tai yra vienas pagrindinių rizikos faktorių PĮ kūrimo metu.

Proceso rizika:

Tai rizika susijusi su produkto kokybe. Sukurta PĮ gali neatitikti užsakovo numatytų standartų. Tai gali įvykti dėl užsakovo nesugebėjimo pateikti tikslų veiklos ir reikalavimų aprašymą arba dėl rangovo nesugebėjimo tinkamai suprasti reikalavimus arba dėl netinkamai parinktos projektavimo technologijos ar netinkamos darbuotojų kvalifikacijos.

Produkto dydžio rizika:

Gali įvykti nesusipratimas tarp kliento ir kūrimo komandos nustatant kuriamos sistemos apimtis. Tai gali sutrukdyti laiku įvykdyti užsakymą. Klaidingai įvertinus sistemos dydį, bus nepakankamai skirta darbuotojų ir resursų.

Technologijos rizika:

Gali būti naudojamos technologijos, kurios jau pasenusios arba greit bus pasenusios. Tokia PĮ būtų naudinga tik trumpai ir užsakovui po kiek laiko vėl reikės naujos PĮ. Kadangi technologijos keičiasi greitai, reikia atkreipti dėmesį į šią riziką.

Rizikos tikimybė ir jos poveikis

6.Lentelė.Rizika surūšiuota pagal tikimybę ir poveikį

Kategorija	Rizika	Tikimybė	Poveikis
Darbuotojų rizika	nepakankama kvalifikacija ir patirtis	40 %	1
Proceso rizika	Žema produkto kokybė	35 %	1
Produkto dydžio rizika	produkto apimtis gali būti neteisingai suskaičiuota	30 %	2
Kūrimo rizika	nepakankami resursai	30 %	2
Kliento rizika	nepakankamas kliento bendradarbiavimas	20 %	3
Technologijos rizika	pasenusi technologija	10 %	2
Poveikio įmonės veiklai rizika	Produktas gali pabloginti įmonės veiklą	10 %	3

Poveikio reikšmės : Apibūdinimas:

- 1 Katastrofiškas
- 2 Pavojingas
- 3 Nedidelis

Rizikos mažinimas, kontrolė ir valdymas

Šioje dalyje detaliai aprašomi rizikos mažinimo, valdymo ir kontrolės būdai.

Produkto dydis

Norint išvengti šios rizikos reikia sutartyje numatyti išlygas:

- padarius pakeitimus reikalavimuose, bus perskaičiuojama projekto apimtis
- sistemos apimtis tam tikrais intervalais bus perskaičiuojama ir jei atsiras naujų veiksnių, skaičiavimuose bus daromi atitinkami pakeitimai
- skaičiavimų rezultatai kūrimo metu gali keistis.

Reikia pastoviai sekti kokios PĮ funkcijos yra būtinos o kurios ne. Sekti DB dydį.

Kokios lentelės ir kurie laukai lentelėje reikalingi.

Jeigu buvo neteisingai įvertinta apimtis (apimtis pasirodė esanti didesnė negu manyta), galima:

padidinti darbo valandų, skirtų projektui, skaičių;
produktą pristatyti evoliucinio prototipo būdu;
sumažinti produkto funkcionalumą.

Poveikis įmonės veiklai

Norint sumažinti šią riziką reikės skirti daugiau laiko bendravimui su vartotojais, kad suprasti jų poreikius. Tuo būdu bus galima gauti daugiau reikalingos informacijos.

Reikės pasistengti gerai suprasti sistemos veiklos aplinką. Reikės supažindinti užsakovą su jau specifiškais reikalavimais, kad išsiaiškinti ar nėra nesupratimų.

Jeigu klaida įvyko, reikės organizuoti susitikimus su vartotojais, išsiaiškinti problemas ir sudaryti planą, kaip pagerinti ar ištaisyti PĮ.

Kliento (vartotojo) rizika

Norint išvengti šios rizikos reikia sudaryti susitikimų su klientu grafiką. Susitikimų metu supažindinti klientą su jau sukurtomis PĮ fazėmis. Sudaryti raštišką susitarimą su klientu, kuriame turi būti numatyti atsakingi asmenys. Jei bus nesilaikoma grafiko, informuoti užsakovo vadovybę. Jei vartotojai per mažai domisi, reikia užduoti jiems klausimus, kad išsiaiškinti ar jiems viskas aišku.

Proceso rizika

Reikia nustatyti sistemos kūrimo gaires kiekvienoje stadijoje. Numatyti ir apibrėžti visus standartus. Tam tikrais laiko tarpais peržiūrėti suprojektuotus sistemos fragmentus.

Technologijos rizika

Reikia išnagrinėti šiuo metu naudojamą technologiją ir naudoti pačią naujausią.

Kūrimo metu taip pat sekti ar nepasikeitė PĮ kūrimo technologijos. Pastebėjus naujas technologijas, jas reikia pritaikyti sistemos projektavime, jei tai nepareikalaus žymių projekto pakeitimų.

Kūrimo rizika

Reikia įsitikinti, kad skirto biudžeto užteks, norint panaudoti pažangiausias technologijas ir įrangą. Jei neužtenka lėšų, reikia projekto grafiką perdaryti taip, kad daugiau laiko liktų programavimo stadijai.

Personalo rizika

Reikia įsitikinti ar darbuotojų kvalifikacijos ir patirties pakanka. PĮ kūrimo metu stebėti ar personalas tam tikruose stadijose nesusiduria su sunkumais ir jei reikia suteikti pagalbą.

4.5. Testavimo strategija

Šiame skyriuje aprašysime testavimo strategiją. Naudosime keturis skirtingus metodus PĮ testavimui.

1. Komponentų testavimas

Kiekvienas sistemos komponentas testuojamas individualiai. Naudojamas "Baltos dėžės" testavimas. Komponentai testuojami įvedant duomenis ir peržiūrint išvedimo duomenis, ieškant klaidų. Reikia įsitikinti ar visi komponentai veikia be sutrikimų.

Pirminį testavimą atliks PĮ kūrėjas. Vėliau testavimas bus perduotas nepriklausomiems testuotojams.

2. Integravimo testavimas

Šis metodas numato PĮ įdiegimą pas užsakovą ir programos paleidimą. Stebėsime ar nėra kolizijos tarp mūsų PĮ ir kliento turimos PĮ. Stebėsime ar nėra sutrikimų kai kelios programos vienu metu veikia tinkle. Reikia įsitikinti ar visi duomenys korektiškai išsaugomi, ar nėra

duomenų praradimo ir DB anomalijų. Pradėsime nuo pagrindinių meniu punktų ir testuosime visas programos funkcijas.

3. Patvirtinimo testas

Šis metodas apima bendravimą su klientu, kad įsitikinti ar PĮ yra tinkama klientui. Norime įsitikinti, kad klientas gavo viską ko tikėjosi iš sistemos. Peržiūrėsime reikalavimų dokumentą, kad išsiaiškinti ar nėra konfliktų ar nesusipratimų su klientu dėl PĮ komponentų. Naudosime “juodos dėžės” testavimą. Turėsime keletą testavimo atvejų. Įvesime duomenis ir gautus rezultatus palyginsime su duomenimis, numatytais iš anksto. Tuo atveju jei čia atsiras problemų, sudarysime trūkumų sąrašą. Klaidoms ištaisyti reikės derėtis su klientu dėl papildomo laiko.

4. “High- order” testavimas

Šis metodas apima keletą skirtingų testų tipų. Testuosime keletą skirtingų sąlygų skirtingais metodais.

- Atsistatymo testavimas

Čia mes domimės PĮ sugebėjimu atkurti prarastus duomenis. Mes norime įsitikinti, kad programa yra atspari klaidoms ir nepraranda duomenų sistemos sustabdymo metu.

- Apsaugos testavimas

Šio metodu tikriname ar apsaugos sistema veikia ir ar kas nors gali nesankcionuotai prieiti prie duomenų bazės. Tai yra vienas iš svarbiausių testų, kadangi sistema turi neleisti nesankcionuoto priėjimo prie DB.

- Stresinis testavimas

Šiame teste stebėsime kaip sistema reaguoja į didelį apkrovimą kai ja vienu metu naudojami daug vartotojų. Sistema neturėtų “užstrigti” dėl ekstremalaus naudojimo.

- Vykdyto testavimas

Vykdyto apribojimai yra nustatomi PĮ projektavimo fazėje. Šie apribojimai padeda nustatyti sistemos efektyvumą. Šie apribojimai taip pat padeda sumažinti vartotojo streso lygį.

Testavimo rezultatų saugojimas

Naudosime Microsoft Excel lentelę testavimo duomenims saugoti. Žemiau pateikiamas tokios lentelės pavyzdys.

Microsoft Excel - Test_rez.xls

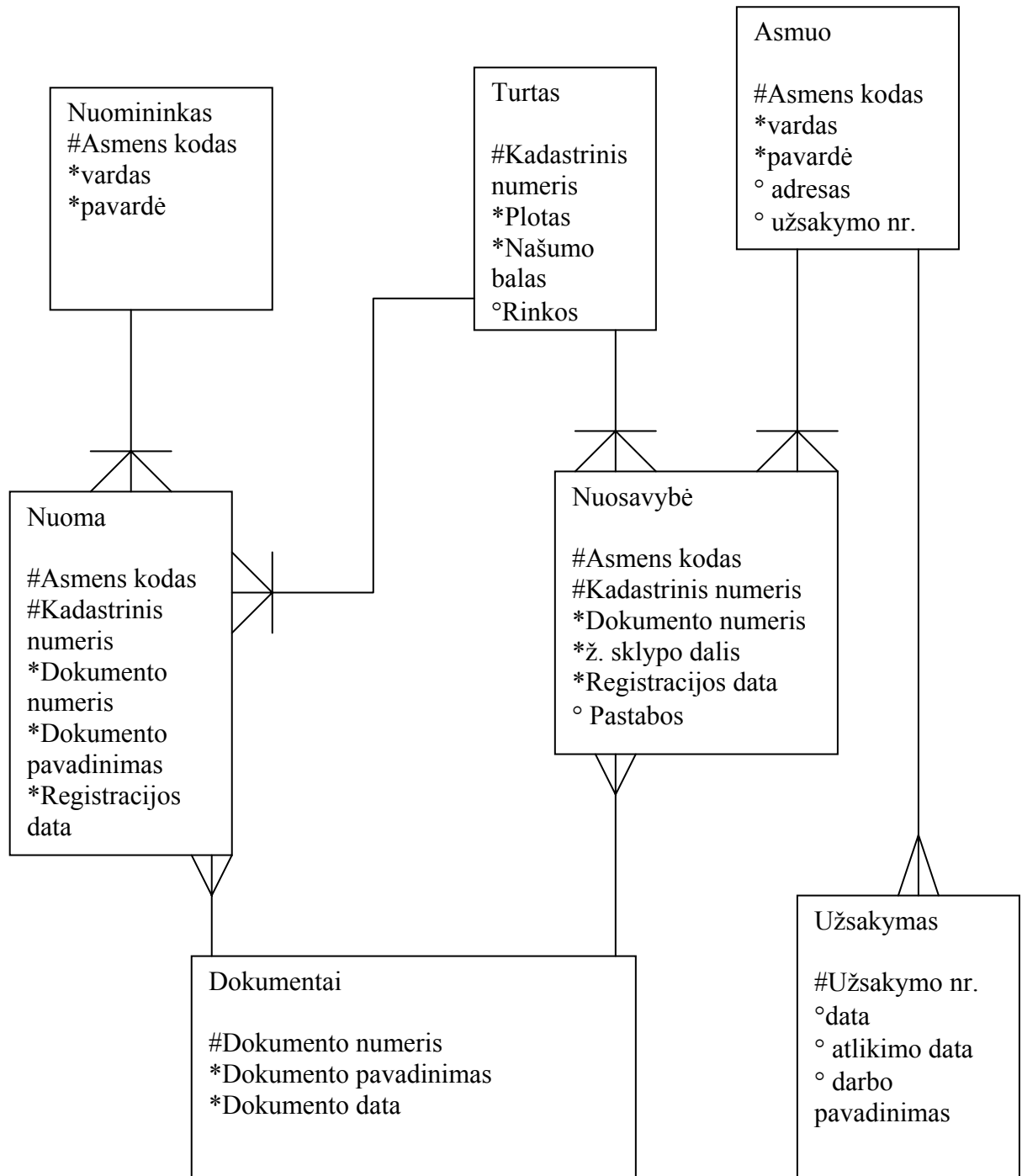
File Edit View Insert Format Tools Data Window Help

C3 = Dokumento pavadinimas korektiškai įrašytas formoje

	A	B	C
1	Testo Nr.	Testo aprašymas	Rezultatai
2	1	Laikas: Laiko tikslumas	Laiko funkcija veikia. Tisklus laikas yra pateikiamas
3	2	Ar dokumento pavadinimas korektiškai įrašytas formoje	Dokumento pavadinimas korektiškai įrašytas formoje
4			
5			
6			
7			
8			

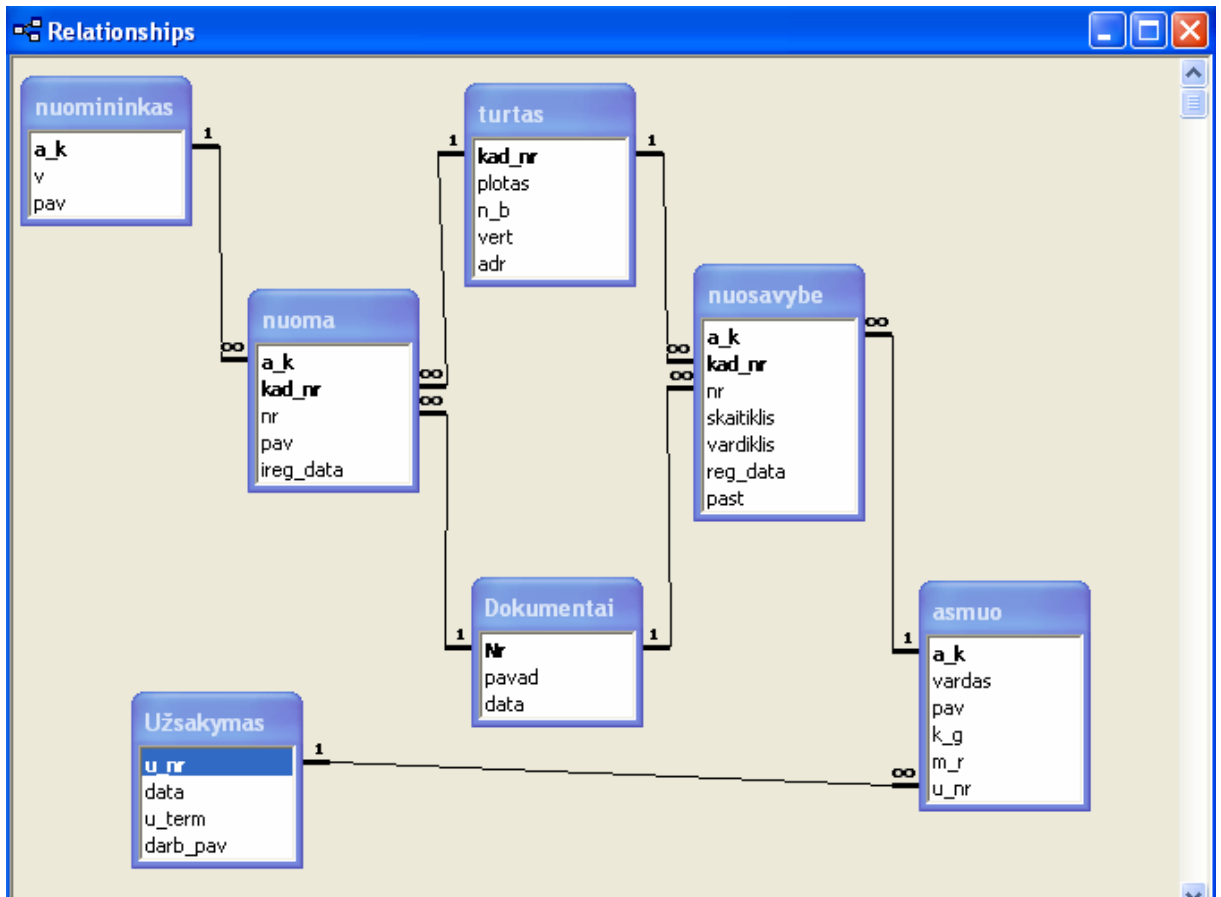
21. pav. Testavimo rezultatų saugojimas

4.6. ER diagrama



22.pav.ER diagrama

4.7. Reliacinis modelis



23.pav. Reliacinis modelis

DB lentelės

Lentelė: Asmuo

Lentelės paskirtis – saugoti duomenis apie ž. sklypų savininkus. Lentelė turi paprastą pirminį raktą – a_k.

Lentelės laukai:

a_k - pirminis raktas. Lauko tipas **Text**, kurio dydis yra 11 simbolių. Pavadinimas (**Caption**) “Asmens kodas”. Įvesta reikšmės tikrinimo taisyklė (**Validation Rule**) Like "3#####" Or Like "4#####", kuri apriboja, kad AK reikšmė prasidėtų skaitmeniu 3 arba 4 ir neviršytų 11 skaitmenų kode. Įspėjimo tekstas (**Validation Text**) –“ Asm. kodas turi prasidėti "3" arba "4" ir turėti 11skaitmenų. Lauko savybė (**Required**) nustatyta reikšmei **Yes** – būtina įvesti reikšmę, o savybės **Indexed - Yes (No Duplicates)** – rakto lauko reikšmė negali kartotis.

vardas - lauko tipas **Text**, kurio dydis 16 simbolių. Pavadinimas (**Caption**) – “Vardas”. Lauko savybė **Required** nustatyta reikšmei **Yes**, t.y. privaloma įvesti reikšmę, o savybė **Indexed - Yes (Duplicates OK)** – rakto lauko reikšmė gali kartotis.

Reikšmių įvedimas palengvinamas nustatant įvedimo kaukę (**Input Mask**) >L<aaaaaaaaaaaaaaaa, kuri palengvina vardo įvedimą (pirmą vardo raidę įvedus mažąja raide, automatiškai pakeičia didžiąja).

pav - lauko tipas **Text**, kurio dydis yra 21 simbolis. Pavadinimas (**Caption**) – “Pavardė”. Lauko savybė **Required** nustatyta reikšmei **Yes**, t.y. privaloma įvesti reikšmę, o savybė **Indexed - Yes (Duplicates OK)** – rakto lauko reikšmė gali kartotis.

Reikšmių įvedimas palengvinamas nustatant įvedimo kaukę (**Input Mask**) >L<aaaaaaaaaaaaaaaaaaaaa, kuri palengvina pavardės įvedimą (pirmą raidę įvedus mažąja raide, automatiškai pakeičia didžiąja).

k_g - lauko tipas **Text**, kurio dydis yra 25 simboliai. Pavadinimas (**Caption**) – “Kaimas/Gatvė”. Lauko savybė **Required** nustatyta reikšmei **No**, t.y. neprivaloma įvesti reikšmę, o savybė **Indexed - Yes (Duplicates OK)** – rakto lauko reikšmė gali kartotis.

m_r - lauko tipas **Text**, kurio dydis yra 15 simbolių. Pavadinimas (**Caption**) – “Miestas/Rajonas”. Lauko savybė **Required** nustatyta reikšmei **No**, t.y. neprivaloma įvesti reikšmę, o savybė **Indexed - Yes (Duplicates OK)** – rakto lauko reikšmė gali kartotis.

u_nr – lauko tipas **Number**.

Lentelė: Turtas

Lentelės paskirtis – saugoti duomenis apie žemės sklypus. Lentelė turi paprastą pirminį raktą – kad_nr.

Lentelės laukai:

kad_nr - pirminis raktas. Lauko tipas **Text**, kurio dydis yra 9 simboliai. Laukas turi įvedimo kaukę (**Input Mask**) 0000\00\000, kuri apriboja kadastrinio numerio įvedimą. Nustato, numerį turi sudaryti dešimtainiai skaitmenys be ženklo. Galimi kadastr. numerio reikšmių variantai:

8101-01-001; 1111-11-111; 1122-00-000;

Lauko pavadinimas (**Caption**) - “Kadastrinis numeris”. Lauko savybė (**Required**) nustatyta reikšmei **Yes** – būtina įvesti reikšmę, o savybės **Indexed - Yes (No Duplicates)** – raktinio lauko reikšmė negali kartotis.

plotas - lauko tipas **Number**, o dydis **Single**. Dešimtinių pozicijų skaičius – 2. Lauko pavadinimas (**Caption**) – “Plotas”. Įvesta reikšmės tikrinimo taisyklė (**Validation Rule**) >0 And ≤ 100 , kuri apriboja, kad “plotas” reikšmė būtų didesnė už 0 ir neviršytų 100. Įspėjimo tekstas (**Validation Text**) – “Ribos [0;100]. Kitos savybės suteiktos pagal nutylėjimą.

n_b - lauko tipas **Number**, o dydis **Byte**. Dešimtinių pozicijų skaičius – 2. Lauko pavadinimas (**Caption**) – “Našumo balas”. Įvesta reikšmės tikrinimo taisyklė (**Validation Rule**) ≥ 0 And ≤ 100 , kuri apriboja, kad reikšmė būtų didesnė arba lygi 0 ir neviršytų 100. Įspėjimo tekstas (**Validation Text**) – “Ribos [0;100]. Kitos savybės suteiktos pagal nutylėjimą.

vert - lauko tipas **Currency**, format **Fixed**. Dešimtinių pozicijų skaičius – 2. Lauko pavadinimas (**Caption**) – “Sklypo rinkos kaina”.

adr - lauko tipas **Text**, kuriuo dydis 30 simbolių. Lauko pavadinimas (**Caption**) – “Kaimas”. Lauko savybė **Required** nustatyta reikšmei **Yes**, t.y. privaloma įvesti reikšmę, o savybė **Indexed - Yes (Duplicates OK)** – rakto lauko reikšmė gali kartotis.

Lentelė: Dokumentai

Lentelės paskirtis – saugoti duomenis apie dokumentus. Lentelė turi paprastą pirminį raktą – Nr.

Lentelės laukai:

Nr – raktinio lauko tipas **Text**, kurio dydis 10 simbolių. Lauko pavadinimas (**Caption**) – “Dokumento numeris”. Lauko savybė (**Required**) nustatyta reikšmei **Yes** – būtina įvesti reikšmę, o savybės **Indexed - Yes (No Duplicates)** – rakto lauko reikšmė negali kartotis.

data - lauko tipas **Date/Time**, format – **Short Date**. Laukas turi įvedimo kaukę (**Input Mask**) 00.00.00;0; kuri ne tiek apriboja kiek palengvina datos įvedimą. Lauko pavadinimas (**Caption**) – “Dokumento data”.

pavad - lauko tipas **Text**, kurio dydis 50 simbolių. Lauko pavadinimas (**Caption**) – “Dokumento pavadinimas”. Lauko savybė (**Required**) nustatyta reikšmei **Yes** – būtina įvesti reikšmę, o savybės **Indexed - Yes (Duplicates OK)** – rakto lauko reikšmė gali kartotis.

Lentelė: Nuosavybė

Lentelės paskirtis – saugoti duomenis apie asmeniui priklausančius sklypus, apie dokumentus. Lentelė turi sudėtinį raktą: a_k+kad_nr.

Lentelės laukai

a_k – raktinio lauko tipas **Text**, kurio dydis 11 simbolių. Lauko pavadinimas (**Caption**) – “Asmens kodas”. Įvesta reikšmės tikrinimo taisyklė (**Validation Rule**) “Like "3#####" Or Like "4#####” , kuri apriboja, kad a_k reikšmė prasidėtų skaitmeniu 3 arba 4 ir turėtų 11 skaitmenų. Įspėjimo tekstas (**Validation Text**) –“ Asm. kodas turi prasidėti "3" arba "4" ir turėti 11 skaitmenų”. Savybės **Indexed** reikšmė **Yes (Duplicates OK)** – raktinio lauko reikšmės gali kartotis (dalis sudėtinio rakto), savybė (**Required**) nustatyta reikšmei **Yes**.

kad_nr - lauko tipas **Text**, kurio dydis yra 9 simboliai. Laukas turi įvedimo kaukę (**Input Mask**) 0000\~00\~000, kuri apriboja kadastrinio numerio įvedimą Pavadinimas (**Caption**) “Kadastrinis numeris”. Lauko savybė (**Required**) nustatyta reikšmei **Yes** – būtina įvesti reikšmę, o savybės **Indexed** - **Yes (Duplicates OK)** – raktinio lauko reikšmė gali kartotis (dalis sudėtinio rakto).

Nr - lauko tipas **Text**, kurio dydis yra 10 simbolių. Pavadinimas (**Caption**) “Dokumento numeris”. Lauko savybė (**Required**) nustatyta reikšmei **Yes** – būtina įvesti reikšmę, o savybės **Indexed** - **Yes (Duplicates OK)** – lauko reikšmė gali kartotis. Kitos savybės suteikiamos pagal nutylėjimą

skaitiklis - lauko tipas **Number**

vardiklis - lauko tipas **Number**

reg_data - lauko tipas **Date/Time**, formatas (**Format**) – **General Date**. Lauko pavadinimas (**Caption**) – “Registro įsigaliojimo data”. Lauko reikšmė pagal nutylėjimą – dvi dienos į priekį nuo dabartinės datos.

Lauko savybė (**Required**) nustatyta reikšmei **Yes** – būtina įvesti reikšmę, o savybės **Indexed** - **Yes (Duplicates OK)** – lauko reikšmė gali kartotis

past - **Memo** tipo laukas, kuriame fiksuojamos pastabos apie žemės sklypą, asmenį ir kt.. Užpildyti lauką nėra būtina: **Required** savybės reikšmė pagal nutylėjimą – **No**. Lauko pavadinimas (**Caption**) –“Pastabos”.

Lentelė: nuomininkas

Lentelės paskirtis – saugoti duomenis apie nuomininkus. Lentelė turi paprastą pirminį raktą – a_k.

Lentelės laukai:

a_k - pirminis raktas. Lauko tipas **Text**, kurio dydis yra 11 simbolių. Pavadinimas (**Caption**) “Asmens kodas”. Įvesta reikšmės tikrinimo taisyklė (**Validation Rule**) Like "3#####" Or Like "4#####” , kuri apriboja, kad AK reikšmė prasidėtų skaitmeniu 3 arba 4 ir

neviršytų 11 skaitmenų kode. Įspėjimo tekstas (**Validation Text**) –“ Asm. kodas turi prasidėti "3" arba "4" ir turėti 11 skaitmenų.

v - lauko tipas **Text**.

pav - lauko tipas **Text**.

Lentelė: nuoma

Lentelės paskirtis – saugoti duomenis apie nuomą sklypus, apie dokumentus. Lentelė turi sudėtinį raktą: a_k+kad_nr.

Lentelės laukai

a_k – raktinio lauko tipas **Text**, kurio dydis 11 simbolių. Lauko pavadinimas (**Caption**) – “Asmens kodas” . Įvesta reikšmės tikrinimo taisyklė (**Validation Rule**) “Like "3#####” Or Like "4#####” ” , kuri apriboja, kad a_k reikšmė prasidėtų skaitmeniu 3 arba 4 ir turėtų 11 skaitmenų. Įspėjimo tekstas (**Validation Text**) –“ Asm. kodas turi prasidėti "3" arba "4" ir turėti 11 skaitmenų ”. Savybės **Indexed** reikšmė **Yes (Duplicates OK)** – raktinio lauko reikšmės gali kartotis (dalis sudėtinio rakto), savybė (**Required**) nustatyta reikl̄ mei **Yes**.

kad_nr - lauko tipas **Text**, kurio dydis yra 9 simboliai. Laukas turi įvedimo kaukę (**Input Mask**) 0000\ -00\ -000, kuri apriboja kadastrinio numerio įvedimą Pavadinimas (**Caption**) “Kadastrinis numeris”. Lauko savybė (**Required**) nustatyta reikšmei **Yes** – būtina įvesti reikšmę, o savybės **Indexed** - **Yes (Duplicates OK)** – raktinio lauko reikšmė gali kartotis (dalis sudėtinio rakto).

nr - lauko tipas **Text**,

pav - lauko tipas **Text**,

reg_data - lauko tipas **Date/Time**, formatas (**Format**) – **General Date**. Lauko savybė (**Required**) nustatyta reikšmei **Yes** – būtina įvesti reikšmę, o savybės **Indexed** - **Yes (Duplicates OK)** – lauko reikšmė gali kartotis

Lentelė: Užsakymas

Lentelės paskirtis – saugoti duomenis apie užsakymus. Lentelė turi paprastą pirminį raktą – u_nr.

Lentelės laukai:

u_nr – lauko tipas **AutoNumber**,

data - lauko tipas **General Date**,

u_term - lauko tipas **Long Date**,

darb_pav - lauko tipas **Text**.

4.8.Vartotojo sąsajos projektas

Formos

Duomenų įvedimą palengvina lentelėms sukurtos vaizdžios formos, turinčios papildomų atributų: mygtukų, iškrentančių sąrašų su galimomis laukų reikšmėmis ir kt. Formų langai atitinkamai apipavidalinti, laukų reikšmės jame išdėstomos laisva tvarka ir vaizduojamos vartotojui patogiu formatu. Duomenų įvedimui skirtos formos, sudarytos iš vienos ar kelių DB lentelių yra: “Dokumentai”, “Asmuo”, “Turtas”, “Registravimas”, “Nuoma”, “Nuomos registravimas”, “Užsakymas”.

Forma: Turtas

Paprasta forma, sukurta lentelės “Turtas” duomenims įvesti. Formoje panaudoti įrankių juostos **Toolbox** elementai: Image, Command button, Combo box., leidžiančių atlikti įvairius veiksmus su formos komponentais bei pačia forma.

Forma: Asmuo

Paprasta forma, sukurta lentelės “Savininkas” duomenims įvesti. Formoje panaudoti įrankių juostos **Toolbox** elementai: **Command button**. Forma turi visus penkis lentelės laukus duomenims įvesti.

Forma yra viena iš pradinių, į kurias pirmiausia įvedami duomenys.

Forma: Dokumentai

Paprasta forma, skirta lentelės “Dokumentai” duomenims suvesti bei vaizduoti. Paimti visi lentelės laukai. Formoje panaudoti įrankių juostos **Toolbox** elementai: **Command button**, **Combo box.**, leidžiančių atlikti įvairius veiksmus su formos komponentais bei pačia forma.

Forma: Registravimas

Forma, skirta suvesti duomenis apie sklypo registravimo datą ir pastabas
Sudėtinė forma, kurią kuriant panaudotos lentelės “Asmuo” ir “Nuosavybė”. Šiuo atveju pagrindinė forma sudaryta iš lentelės “Asmuo”. Pavaldi forma (**Subform**) sudaryta iš lentelės “nuosavybė” .

Forma: Nuomininkas

Paprasta forma, skirta lentelės “Nuomininkas” duomenims suvesti bei vaizduoti. Paimti visi lentelės laukai. Formoje panaudoti įrankių juostos **Toolbox** elementai: **Command button**, **Combo box**., leidžiančių atlikti įvairius veiksmus su formos komponentais bei pačia forma.

Forma: Užsakymas

Paprasta forma, skirta lentelės “Užsakymas” duomenims suvesti bei vaizduoti. Paimti visi lentelės laukai. Formoje panaudoti įrankių juostos **Toolbox** elementai: **Command button**, **Combo box**., leidžiančių atlikti įvairius veiksmus su formos komponentais bei pačia forma.

Forma: nuomos registravimas

Forma, skirta suvesti duomenis apie sklypo nuomos registravimo datą ir pastabas.

Sudėtinė forma, kurią kuriant panaudotos lentelės “nuomininkas” ir “nuoma”. Šiuo atveju pagrindinė forma sudaryta iš lentelės “nuomininkas”. Pavaldi forma (**Subform**) sudaryta iš lentelės “nuoma” .

Užklausa

Užklausa: visi_savinin

Tai paprasta užklausa iš lentelės ‘Asmuo’ išrenkanti visų savininkų vardus, pavardes ir asmens kodus.

Užklausa: vidut_vertė

Užklausa parodo savininko vardą, pavardę, jų turimų sklypų plotą, vertę ir paskaičiuoja vidutinę 1ha vertę. Užklausa taip pat atrenka sklypus didesnius už 0 ir mažesnius už 50ha.

Užklausa: pasirink_plota

Užklausa išrenka savininkus, kurių turimų sklypų plotas atitinka pasirinktąjį.

Užklausa: Kiek_kaime_skl

Kryžminė užklausa parodo koks savininkas kiek sklypų turi atitinkamame kaime.

Ataskaitos

Ataskaita : Savininkas

Ataskaitoje pateikta savininkų vardai, pavardės ir turimų sklypų plotai bei vertės.

Ataskaita : Pažymėjimas

Ataskaitoje pateikta vieno savininko vardas, pavardė ir turimų sklypų plotai , vertės.

Ataskaita : Pažyma

Ataskaitoje pateikta vieno savininko vardas, pavardė ir turimų sklypų plotai , vertės.

Ataskaitoje spausdinamas pažymos numeris.

Duomenų kontrolės logika

Duomenų kontrolė leidžia išvengti klaidų įvedant duomenis, pateikiant užklausas. Tuo palengvinamas vartotojo darbas – jam parodomas įspėjimo tekstas, IS neatsiranda tokių klaidų sąlygotų funkcionavimo sutrikimų.

Pagrindinė kontrolė skiriama privalomam raktinių laukų įvedimo tikrinimui. Reikalaujama iš vartotojo būtinai įvesti duomenis, kurie yra raktinių laukų reikšmės.

Pavyzdžiui:

Lentelėje “Asmuo” įvesta reikšmės tikrinimo taisyklė (**Validation Rule**) Like "3#####"
Or Like "4#####", kuri apriboja, kad AK reikšmė prasidėtų skaitmeniu 3 arba 4 ir neviršytų 11 skaitmenų kode. Įspėjimo tekstas (**Validation Text**) –“ Asm. kodas turi prasidėti "3" arba "4" ir turėti 11 skaitmenų. Lauko savybė (**Required**) nustatyta reikšmei **Yes** – būtina įvesti reikšmę, o savybės **Indexed - Yes (No Duplicates)** – rakto lauko reikšmė negali kartotis.

Lentelėje “Turtas” **kad_nr** - pirminis raktas. Lauko tipas **Text**, kurio dydis yra 9 simboliai. Laukas turi įvedimo kaukę (**Input Mask**) 0000\ -00\ -000, kuri apriboja kadastrinio numerio įvedimą. Nustato, numerį turi sudaryti dešimtainiai skaitmenys be ženklo. Galimi kadastr. numerio reikšmių variantai:

8101-01-001; 1111-11-111; 1122-00-000;

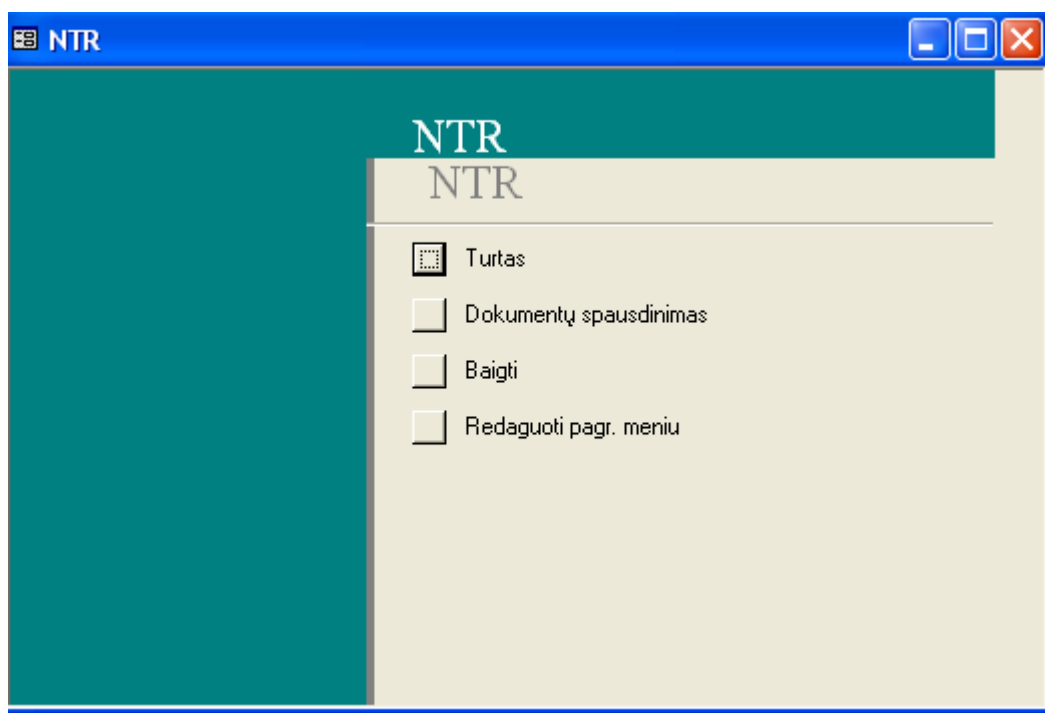
Laukas– “Plotas”. Įvesta reikšmės tikrinimo taisyklė (**Validation Rule**) >0 And <=100, kuri apriboja, kad “plotas” reikšmė būtų didesnė už 0 ir neviršytų 100. Įspėjimo tekstas (**Validation Text**) –“ Ribos [0;100].

Pagrindinė forma : Kadastro duomenų bazė

Pagrindinė forma palengvina vartotojo darbą su MS Access duomenų baze.

Forma turi mygtukus “Turtas”, “Dokumentų spausdinimas”, kurių pagalba patenkama į papildomus meniu langus, skirtus naujų duomenų įvedimui, dokumentų spausdinimui.

Mygtukas “Redaguoti pagr. meniu” yra skirtas pačios pagrindinės formos koregavimui ir tobulinimui bei pritaikymui vartotojo poreikiams.



24. pav. Pagrindinė forma

4.9. Vartotojo dokumentacija

Kadangi sistema dar nėra visiškai išbaigta, todėl čia pateiksime preliminarią vartotojo dokumentaciją:

Funkcinis sistemos aprašymas:

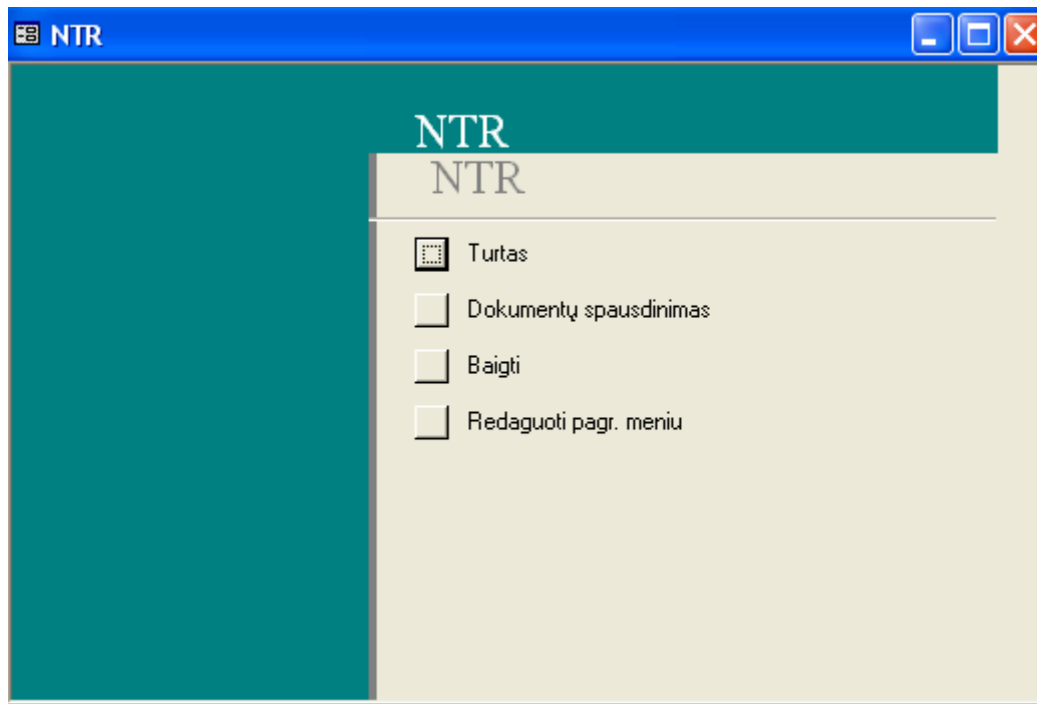
Sistema skirta nekilnojamajam turtui (žemės sklypams) registruoti. Ji leidžia greitai ir paprastai įvesti duomenis apie sklypą, gauti suvestinius duomenis, spausdinti pažymas.

Pagrindinės sistemos galimybės:

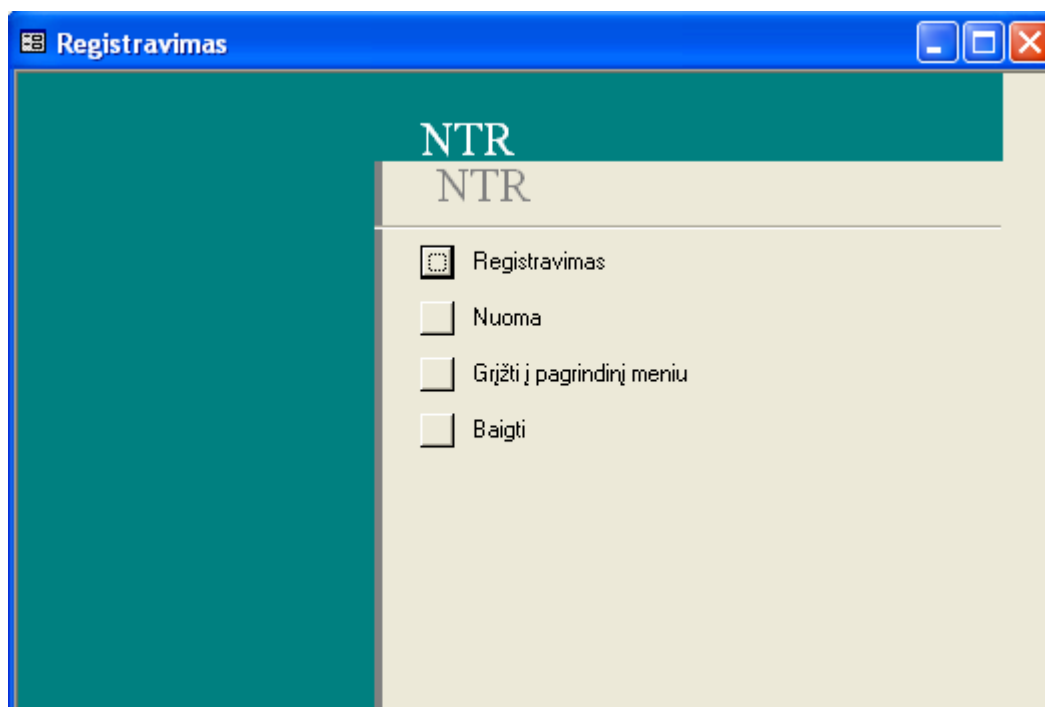
1. Informacijos perdavimas iš vieno etapo į kitą vietinio tinklo ir DB pagalba.
2. Duomenų baze gali naudotis visų skyrių darbuotojai. Duomenų bazėje laikoma sukurta ir gaunama informacija. Kiekvienas skyrius vietinio tinklo pagalba gali perduoti duomenis ir informaciją į bendrą duomenų bazę. Vietinio tinklo pagalba gali būti sujungti du ar daugiau skyriaus darbuotojų.
3. Pakankamas tinklo ir duomenų bazės greitis, duomenų bazės valdymo paprastumas.
4. Nesudėtinga sąsaja su vartotoju, leidžia išvengti klaidų ir informacijos praradimo duomenų bazėje dėl vartotojo neapdairumo.
5. Programa atlieka:
 - Naujo Nekilnojamojo turto vieneto (žemės sklypo) registravimas (tekstinių duomenų apie naujai suformuotą žemės sklypą įvedimas)
 - Nuosavybės, nuomos ir kt. daiktinių teisių žemės sklypui registravimas.
 - Pažymėjimo apie Nekilnojamojo turto registre įregistruotą žemės sklypą ir teises į jį spausdinimas.
 - Pažymos sandoriui spausdinimas.
 - Apribojimų (turto areštas, įkeitimas ir kt.) turtui įvedimas į duomenų bazę.
 - Turto paieška pagal :
 1. registro numerį,
 2. pavardę, asmens kodą,
 3. unikalų žemės sklypo kodą,
 4. turto adresą.

Kaip naudotis programa (registravimo pavyzdys) :

1. Paleidžiame failą *ntr_1.mdb*.
2. Atsidaro pagrindinio meniu langas :



3. Spaudžiame mygtuką „Turtas“. Pereiname į pagalbinio meniu langą.



3. Spaudžiame mygtuką „Registravimas“. Atsidaro forma „Turtas“. Suvedame duomenis. Spaudžiame – „Išsaugoti“

turtas

Kadastrinis numeris	1234-12-123	Ištrinti įrašą
Plotas	9,15	
Išsumo balas	36	
Sklypo rinkos kain	10000,00	Baigti
Kaimas	Dainavos k.	

Išsaugoti **Pereiti į formą "Asmuo"** **Naujas įrašas**

Record: 1 of 1

4. Spaudžiame mygtuką „Pereiti į formą „Asmuo““ ir suvedame asmens duomenis. Spaudžiame – „Išsaugoti“.

asmuo

Asmens kodas	3999999999	Ištrinti įrašą
Vardas	Jonas	
Pavardė	Jonaitis	
Kaimas/Gatvė	xxxx	Baigti
Miestas/Rajonas	zzzz	

Išsaugoti **Pereiti į formą "Dokumentai"** **Naujas įrašas**

Record: 4 of 4

5. Analogiškai suvedame dokumentus.

6. Iš formos „Dokumentai“ pereiname į formą „Registravimas“. Vardas, Pavardė, Asmens kodas atsiranda automatiškai, įvedame sklypo kadastrinį numerį (galima pasirinkti iš sąrašo pateikiamo langelio apačioje), analogiškai įvedame dokumentą, įvedame asmens valdomą dalį (skaitiklį ir vardiklį).

7. Paspaudę mygtuką „Tikrinti“ galima patikrinti ar sklypo dalių suma nėra didesnė už vienetą.

nuosavybe Query1 Query : Select Query	
Kadastrinis numeris	Sum Of Dalis
1111-11-111	0,25
1234-12-123	1
2222-22-222	1,6666666667

4.10.Sistemos įdiegimas:

- Reikalavimai sistemai.
IBM PC 166 Mhz, 4MB HDD, 32 RAM, Windows operacinė sistema, pelė, klaviatūra.
- Nukopijuoti *nr_1.mdb* bylą į vartotojo sukurtą katalogą. Sukurti šaukinį pagrindinei paleidžiamajai bylai.

4.11.Produkto kokybės įvertinimas

PĮ sistemos kokybė bus vertinama sistemos testavimo metu, bei įvertinant vartotojų atsiliepimus (anketos vartotojams).

5. Išvados

Atlikus programinės įrangos analizę buvo išnagrinėti duomenų bazių programinės įrangos kūrimo būdai (CASE - kompiuterizuota programų inžinerija, Objektiškai orientuotas IS kūrimas, Informacijos sistemos kūrimas sudarant komponentinį modelį, klasių diagramos), duomenų bazėms keliami reikalavimai bei vartotojų poreikiai. Sudarytas žemės kadasro programinės įrangos sistemos projektas. Idėjos iliustracijai paketu *ProVision Workbench*, bei informacijos sistemos kūrimo sudarant komponentinį modelį būdu buvo sukurti programinės įrangos prototipai.

Vartotojo biznio veiklai atvaizduoti strateginėje perspektyvoje buvo sudaryti Use Case modeliai. Informaciniams ir materialiams srautams organizacijos viduje ir išorėje nustatyti, buvo sudarytos DFD – duomenų srautų diagramos, dabų sekų modeliai ir kt.

Šiai duomenų bazei kurti, atsižvelgiant į vartotojo poreikius, DB apimtį, kūrimo kaštus buvo pasirinkta DBVS *MS Access*, nes duomenų bazių valdymo sistema tampa vis populiareesnė ir Lietuvoje. *MS Access* yra unikali tiek savo galimybėmis, tiek ir tinkamumu labai skirtingiems vartotojams (nuo eilinio vartuotojo iki profesionalaus programuotojo).

Programa gerai dirba su operacinėmis sistemomis *Windows 9**, *Windows 2000*, *Windows XP*.

Galima kurti tiek autonomines, tiek tinklines taikomas programas, taip pat programas, skirtas dirbti sistemose „Klientas – serveris“. *MS Access* stabiliai dirba, turi daug funkcijų.

Sudariau sistemos projektą, parinkau technologines realizavimo priemones;

Sudariau programinį produktą, aprašiau jo savybes;

Nagrinėjau mokslinius leidinius darbo tematika, analizavau programinės įrangos technologijas, konsultavausi teoriniais ir inžineriniais klausimais.

6.Literatūra

1. Duomenų bazių projektavimas [interaktyvus]. [žiūrėta 2001-12-15], prieiga per internetą: <ftp://isd.ktu.lt/isd/Gudas/RS2000/DbirIS/>
2. How to draw Data Flow Diagrams [interaktyvus]. [interaktyvus]. [žiūrėta 2001-12-15], prieiga per internetą: <http://www.smartdraw.com/resources/centers/software/>
3. ProVision Workbench; UML; [interaktyvus]. [žiūrėta 2003-03-16], prieiga per internetą: <ftp://isd.ktu.lt/isd/Gudas/PS2003/OCASEtechn/>
4. Security, Privacy and Risk Management: 2002 and Beyond [interaktyvus]. [žiūrėta 2001-12-15], prieiga per internetą: [ftp://isd.ktu.lt/isd/Gudas/mag2000/GARTNER research reports/](ftp://isd.ktu.lt/isd/Gudas/mag2000/GARTNER_research_reports/)
5. Structuring System Requirements: Process Modeling [interaktyvus]. [žiūrėta 2002-01-10], prieiga per internetą: <http://www.best.uts.ohio-state.edu/Bpcourses?Bentley/HIMS/LectureNotes/Structun.htm/>
6. Volere Requirements Specification Template [interaktyvus]. [žiūrėta 2002-01-11], prieiga per internetą: <ftp://isd.ktu.lt/isd/Gudas/mag2000/HURTWITZgroupreports/>
7. Baronas R. Duomenų bazių sistemos: metodinė priemonė. Vilnius: TEV, 2002. 126p.
8. Gudas S. Veiklos analizė ir informacinių poreikių specifikavimas: mokomoji knyga. Kaunas: Naujasis LANKAS, 2002. 93P.
9. Minkevičius S. Microsoft Access XP: mokomoji knyga. Vilnius: PIKETAS, 2003. 147p.

7. Summary

Reengineering of State Land Cadastre data base

Currently there are many software systems for creating data bases widely used in the world.

There were made an analysis of existing software and customer needs in this work.

There were made an analysis of *Object-oriented programing, CASE – computer aided system engeenering, Component modeling, MS Access.*

To understand business environment I created Business Interaction Model, Workflow models, Use Case Model. For this purpose I used *ProVision Workbench* package.

I created data base project. I used *MS Access XP* because of its stability, multipurpose, potential, good working with *Windows 9*, Windows 2000, Windows XP* operational systems.

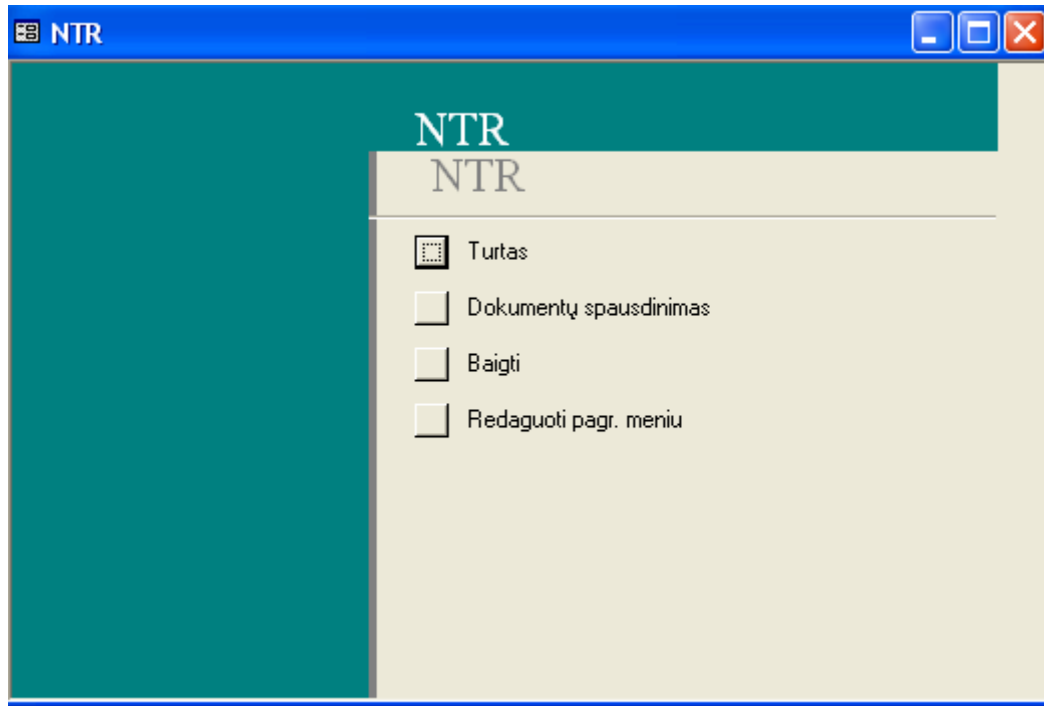
There were made an User Interface, including forms , reports, queries.

8. Terminų ir santrumpų žodynas

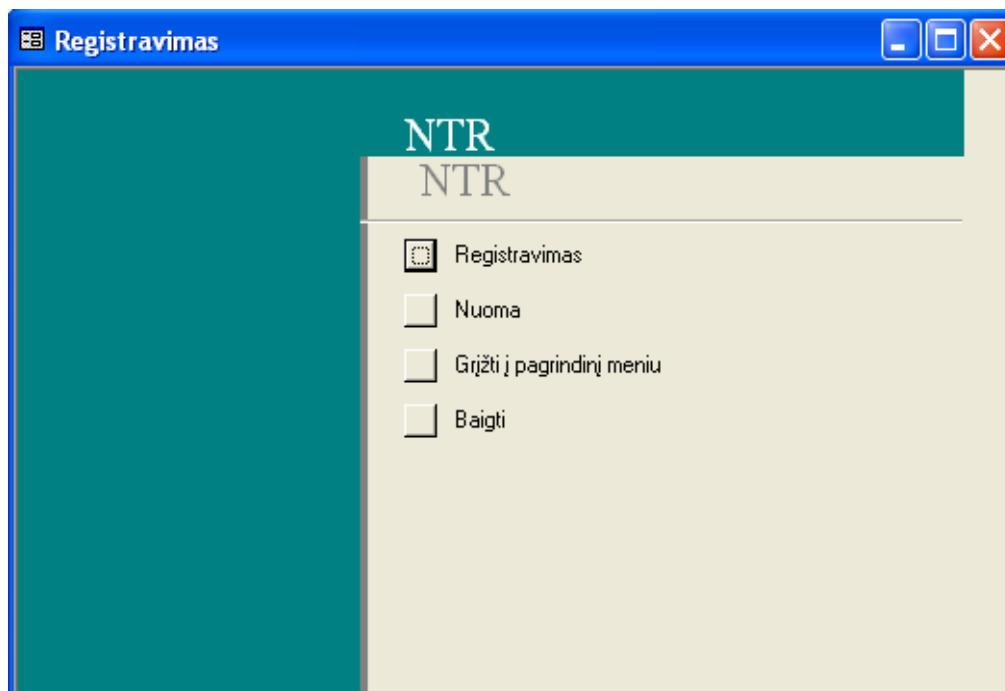
Pavadinimas	Paiškinimas
GC	Gyvavimo ciklas
CASE	Computer Aided System Engeneering – Kompiuterizuota programų inžinerija
OO	Objektiškai orientuotas
UML	Unifikuota modeliavimo kalba
DFD	Data Flow Diagrams – Duomenų srautų digramos
IS	Informacijos sistema
Use Case	Taimomųjų uždavinių modelis
DB	Duomenų bazė
VIA	Veiklos informacinė architektūra
GUI	Grafic user interface – Grafinė vartotojo sąsaja
ER digrama	Esybių ryšio diagrama
PĮ	Programinė įranga

9.Priedai

Priedas Nr.1. Pagrindinė forma



25.pav. Pagrindinė forma – „NTR“

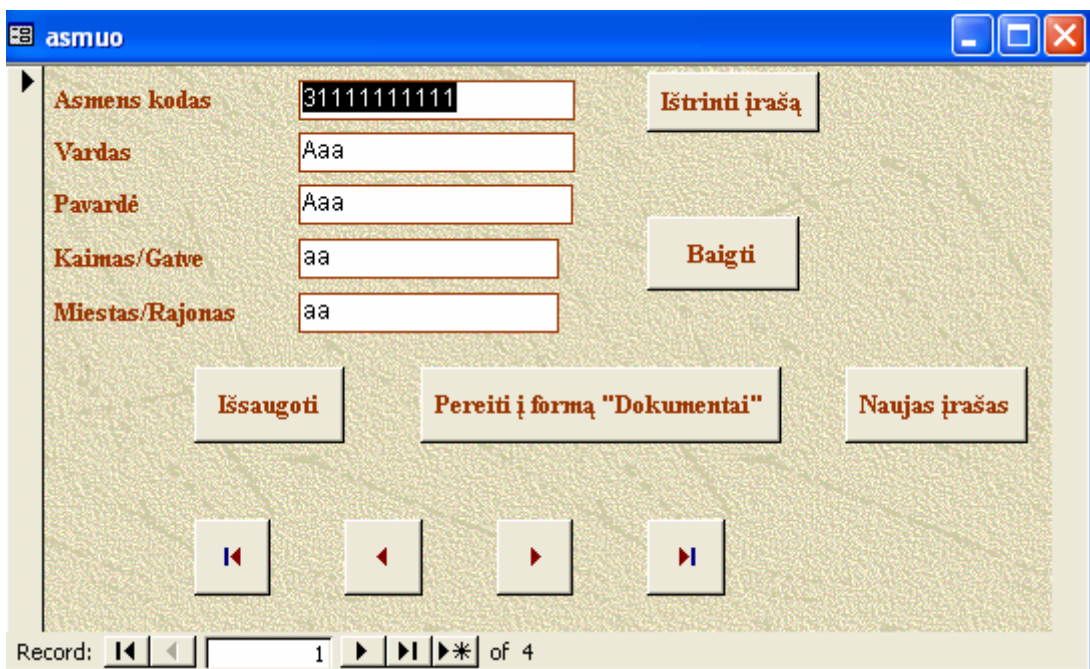


26.pav. Pagalbinė forma – „Registravimas“



27.pav. Pagalbinė forma – „Dokumentų spausdinimas“

Priedas Nr.2. Formos.



28.pav. Forma – „Asmuo“

Dokumentai

Dokumento numeris

Dokumento pavadinimas pasirinkite

Dokumento data

Išsaugoti Pereiti į formą "Registravimas" Naujas įrašas Baigti

Ištrinti įrašą

Record: 1 of 2

29.pav. Forma – „Dokumentai“

nuomininkas

Asmens kodas Vardas Pavardė

Ištrinti įrašą Naujas įrašas Nuomos registravimas Baigti

Record: 1 of 2

30.pav. Forma – „Nuomininkas“

nuomos registravimas

pav

nuoma

Asnens kodas	Kadastrinis numeris	Dokumento numeris	Dokumento pavadinimas	Registravimo data
<input type="text" value="3777777777"/>	<input type="text" value="2222-22-222"/>	<input type="text" value="3"/>	<input type="text" value="cc"/>	<input type="text"/>
	<input type="text" value="123412123"/> <input type="text" value="22222222"/>	<input type="text" value="3"/> <input type="text" value="4"/>	<input type="text" value="cc"/> <input type="text" value="pp"/>	
* <input type="text" value="3777777777"/>	<input type="text"/>	<input type="text" value="0"/>	<input type="text"/>	<input type="text"/>
	<input type="text" value="111111111"/>	<input type="text" value="3"/>	<input type="text" value="cc"/>	

Record: of 1

31.pav.Forma – „Nuomos registravimas“

asmuo 1

Vardas

Pavardė

Baigti

nuosavybe

Asnens kodas	Kadastrinis numeris	Dokumento Nr.	skaitiklis	vardiklis	Registro įsigaliojimo data	Pastabos
<input type="text" value="8111111111"/>	<input type="text" value="1111-11-111"/>	<input type="text" value="8"/>	<input type="text" value="1"/>	<input type="text" value="4"/>	<input type="text" value="2008.12.28 21:08:05"/>	<input type="text"/>
	<input type="text" value="11111111"/>	<input type="text" value="8"/>				
<input type="text" value="8111111111"/>	<input type="text" value="2222-22-222"/>	<input type="text" value="4"/>	<input type="text" value="1"/>	<input type="text" value="1"/>	<input type="text" value="2008.12.28 21:55:39"/>	<input type="text"/>
	<input type="text" value="222222222"/>	<input type="text" value="4"/>				

Record: of 2

Record: of 4

32.pav.Forma – „Registravimas“

turtas

Kadastrinis numer	111-11-111	Ištrinti įrašą
Plotas	1	
Našumo balas	36	
Sklypo rinkos kain		Baigti
Kaimas	aaa	

Išsaugoti Pereiti į formą "Asmuo" Naujas įrašas

Record: 1 of 3

33.pav. Forma – „Turtas“

Užsakymas

Užsakymo Nr.	
Data	2003.10.11
Atlikimo data	2003 m. lapkritis 11 d.
Užsakomo darbo pavad.	XXXXXXXXXXXXXXXX

Baigti Pašalinti įrašą Naujas įrašas

Record: 1 of 1

34.pav. Forma – „Užsakymas“

Priedas Nr.3. Užklauso

	Vardas	Pavardė	Asmens kodas
▶	Petras	Petraitis	31234567891
	Juozas	Imbrasas	33333333331
	Petras	Jonaitis	33333333333
	Petras	Umbrasas	34444444441
	Petras	Karosas	34444444442
	Petras	Pagirys	34444444445
	Petras	Rupeika	34444444446
	Henris	Fordas	34444444448
	Laim	Bimbiriene	44444444444
*			

35.pav. Užklausa - “visi_savinin”

	Pavardė	Vardas	Sklypo rinkos kaina	Plotas	1ha kaina
▶	Bimbiriene	Laim	3641,00	3	1 213,67
	Petraitis	Petras	3000,00	1	3 000,00
	Petraitis	Petras	4000,00	44	90,91
	Pagirys	Petras	5894,00	5	1 178,80
	Umbrasas	Petras	23496,00	6	3 916,00
	Karosas	Petras	2265,00	7	323,57
	Imbrasas	Juozas	2265,00	7	323,57
	Jonaitis	Petras	5894,00	5	1 178,80
	Rupeika	Petras	5894,00	5	1 178,80
	Fordas	Henris	53420,00	8	6 677,50
	Fordas	Henris	5894,00	5	1 178,80

36.pav. Užklausa - “vidut_verte”

Enter Parameter Value

=plotas

OK Cancel

pasirink_plota : Select Query

	Vardas	Pavardė	Plotas
▶	Petras	Pagirys	5
	Petras	Jonaitis	5
	Petras	Rupeika	5
	Henris	Fordas	5
*			

Record: 14 of 4

38.pav. Užklausa – “pasirnk_plota”

Kiek_kaime_skl : Crosstab Query

	Vardas	Pavardė	Laumenai	m	Pabaiskas	Siesikai	Vepriai	Žemaitkiemis
▶	Henris	Fordas					2	
	Juozas	Imbrasas				1		
	Laim	Bimbiriene		1				
	Petras	Jonaitis					1	
	Petras	Karosas				1		
	Petras	Pagirys					1	
	Petras	Petraitis			1			1
	Petras	Rupeika					1	
	Petras	Umbrasas	1					

Record: 14 of 9

39.pav.Užklausa – “kiek_kaime_skl”

Savininkas			Date: 07.01.03 19:04
Pavardė	Bimkiriene		
Sklypo rinkos kaina	Plotas	1ha kaina	
3641,00	3	1213,67	
Pavardė	Fondas		
Sklypo rinkos kaina	Plotas	1ha kaina	
5894,00	5	1178,80	
53420,00	8	6677,50	
Pavardė	Imbrasas		
Sklypo rinkos kaina	Plotas	1ha kaina	
2265,00	7	323,57	

40.pav.Ataskaita – “Savininkas”

asmuo

Pažymėjimas

Vardas

Pavardė

Žemės sklypo plotas

9,15

Kaimas **Dalis:**

Dainavos k. 1

1

Dokumento pavadinimas

cc

Dokumento numeris

3

41.pav.Ataskaita – „Pažymėjimas“

Pažyma

Numeris: 123456

<i>Asmens kodas</i>	<input type="text" value="399999999"/>
<i>Vardas</i>	<input type="text" value="Jonas"/>
<i>Pavardė</i>	<input type="text" value="Jonaitis"/>
<i>Dokumento numeris</i>	<input type="text" value="3"/>
<i>Dokumento pavadinimas</i>	<input type="text" value="cc"/>
<i>Dokumento data</i>	<input type="text" value="2000.11.11"/>
<i>Kadastrinis numeris</i>	<input type="text" value="1111-11-111"/>
<i>Plotas</i>	<input type="text" value="1"/>
<i>Sklypo rinkos kaina</i>	<input type="text"/>
<i>Kaimas</i>	<input type="text" value="aaa"/>
<i>skaitiklis</i>	<input type="text" value="1"/>
<i>vardiklis</i>	<input type="text" value="4"/>
<i>Registro įsigaliojimo data</i>	<input type="text" value="2003.12.28 21:08:05"/>
<i>Asmens kodas</i>	<input type="text" value="399999999"/>
<i>Vardas</i>	<input type="text" value="Jonas"/>
<i>Pavardė</i>	<input type="text" value="Jonaitis"/>
<i>Dokumento numeris</i>	<input type="text" value="3"/>

42.pav.Ataskaita – „Pažyma“