

**KAUNO TECHNOLOGIJOS UNIVERSITETAS**

**KOMPIUTERIŲ KATEDRA**

Aurimas Paršonis

**MOBILI INFORMAVIMO SISTEMA**

**MAGISTRO BAIGIAMASIS DARBAS**

**Vadovas:** profesorius

*Egidijus Kazanavičius*

**KAUNAS, 2005**

## TURINYS

1. Analizės dalis .....	6
1.1 Informacinių sistemų analizės metodai ir priemonės .....	6
1.1.1 Integration Definition metodų šeima .....	7
1.1.2 Objektinio modeliavimo metodas.....	10
1.1.3 Objektiškai orientuoto proceso, aplinkos ir notacijos metodas.....	10
1.1.4 Unifikuota modeliavimo kalba .....	10
1.2 Literatūros šaltiniuose pateiktų sprendimų problemai spręsti lyginamoji analizė .....	12
1.2.1 Dienotvarkės ir planavimo įrankis „Appointment Scheduler“.....	12
1.2.2 Kalendorius „Web Calendar“ .....	12
1.2.3 Dienotvarkės tvarkyklė „Flex 7“ .....	13
1.2.4 Dienotvarkės planavimo sistema „Appointment Scheduler“ .....	13
1.2.5 Internetu pagrįsta programa „Super Calendar“ .....	14
1.2.6 Profesionali dienotvarkės valdymo sistema „Weblicon“ PIM.....	14
1.2.7 Išvados. ....	16
1.3 Organizacijos veiklos analizė.....	16
1.3.1 Sistemos kokybės kriterijai .....	16
1.3.2 Projektavimo metodai ir priemonės.....	17
1.4 Organizacijos veiklos analizė.....	17
1.4.1 Analizės tikslas .....	17
1.4.2 Veiklos tikslų modelis.....	17
1.4.3 Veiklos procesų modelis .....	18
1.4.4 Studento ir dėstytojo veiklos procesas.....	19
1.4.5 Veiklos objektų modelis.....	19
1.4.6 Veiklos procesų modelis .....	20
1.5 Analizės išvados .....	20
2. Sistemos struktūra ir realizacija .....	22
2.1 Kompiuterizuojamos sistemos varianto parinkimas.....	22
2.2 Vartotojų panaudojimo atvejų modelis.....	22
2.3 Specifikacijos sistemos panaudojimo atvejams .....	23
2.3.1 Naujo vartotojo registracijos panaudojimo atvejis .....	23
2.3.2 Studentų ir dėstytojų informavimo panaudojimo atvejis .....	23
2.3.3 Asmeninės informacijos koregavimo panaudojimo atvejis .....	24

2.3.4 Vartotojo ištrynimo panaudojimo atvejis.....	24
2.4 vartotojo ir sistemos sekų modelis .....	25
2.5 Dėstytojo SMS pranešimo sukūrimo sekų modelis.....	27
2.6 Sistemos elgsenos modelis, kai vartotojas dėstytojas.....	28
2.7 Sistemos elgsenos modelis, kai vartotojas studentas.....	29
2.8 Dalykinės srities statinis modelis .....	30
2.8.1 Dėstytojo klasių modelis .....	31
2.8.2 Studento klasių modelis .....	31
2.9 Vartotojų interfeiso modelis.....	32
2.10 Duomenų bazės modelis .....	33
2.11 Duomenų bazės lentelių aprašymas.....	34
2.12 Sistemos panaudojimo atvejis .....	35
3. Išvados.....	37
4. Literatūra.....	38
5. Terminų ir santrumpų žodynas .....	40
Priedai.....	42

## **Išvadas**

Šiomis dienomis žmonės tampa vis labiau mobilesni, jie nuolat keliauja, keičiasi informacija, kontaktuoja su kitais žmonėmis nebūdami nuolatinėje darbo vietoje. Informacijos srautas vis labiau auga, nuolat kinta jos gavimo bei perdavimo būdai, todėl labai svarbu ją pateikti bei gauti kuo greičiau ir kuo patogesniu būdu, nes nuo to priklauso daugelio žmonių tolesni veiksmai. Dėl šių priežasčių vis daugiau informacijos valdymo sistemų yra prieinamos per internetą bei mobiliąsias technologijas, kurios nuolat tobulinamos ir kuriamos vis naujesnės, stengiantis patenkinti vis didėjančius vartotojų poreikius. Šios komunikavimo priemonės tampa neatskirama kasdienio gyvenimo dalimi.

Kauno technologijos universitete studentai pakankamai dažnai susiduria su problema – dėstytojas neatėjo į paskaitą, nes buvo pakviestas į susirinkimą, ar perkėlė susitikimo laiką. Iki šiol dažniausiai ši problema būdavo sprendžiama pakabinant skelbimą apie pasikeitimus fakulteto skelbimų lentoje. Dėl šios ir panašių problemų yra gaištamas studentų laikas, kurie nežino dėstytojo tvarkaraščio, juo labiau, kada jiems skiriami susirinkimai ar iškyla kitokios iš anksti nenumatytos problemos, dėl kurių neįvyksta paskaita ar susitikimas.

Siekiant išvengti tokių nesusipratimų ir informuoti studentus bei dėstytojus apie dienotvarkę ir jos pasikeitimus, reikalinga greitai, patogiai bei kokybiškai pateikti informaciją. Vienas iš sprendimo būdų – internetinėmis bei mobiliomis technologijomis pagrįsta mobili informavimo sistema, padedanti informuoti suinteresuotus asmenis iš bet kurio kompiuterio, turinčio prieigą prie interneto.

Tyrimo sritis. Šis magistrinis darbas pagrįstas internetinėmis, mobiliomis technologijomis bei duomenų baze. Baigta sistema bus patalpinta į internetą. Šios sistemos tipas – vartotojas visus veiksmus atlieka naudodamasis interneto naršykle.

Darbo objektas. Šio magistrinio darbo objektas – internetu pagrįsta mobili informavimo sistema. Tai galinti veikti atskirai nuo Kauno Technologijos Universiteto sistemos, ar būti integruota į ją sistema. Jos paskirtis – studentams pateikti informaciją apie dėstytojus bei jų tvarkaraščius, o dėstytojams – koreguoti savo dienotvarkes, informuoti apie pasikeitimus norimus asmenis. Vartotojas su sistema dirbs interneto naršyklės pagalba.

Problema. Šiuo metu Kauno Technologijos Universitete nėra nei vienos mobilios informavimo sistemos, bei tokios, per kurią studentai ar dėstytojai galėtų sužinoti dėstytojų tvarkaraščius, jų laisvą laiką, nusiųsti pranešimą elektroniniu paštu ar trumpąja SMS žinute. Informacija apie dėstytojų tvarkaraščius, jų elektroniniai pašto adresai, jų fakultetai, katedros ir

kabinetai yra žinomi ir prieinami visiems, tačiau ši informacija nėra susisteminta, sudėta į vieną vietą ir patogiai bei greitai pasiekama.

*Darbo aktualumas.* Sukūrus šią sistemą, vartotojai turės galimybę pasiekti visą reikalingą informaciją iš vienos vietos. Dėstytojai galės prisijungti prie sistemos ir koreguoti savo tvarkaraštį, dienotvarkę, paskirti susitikimus su studentais ir pan. Be to, jie galės siųsti pranešimus elektroniniu paštu ir/arba trumpąja SMS žinute dėstytojams bei studentams. Taip pat bus realizuota galimybė kurti studentų ar dėstytojų grupes, kuriems bus siunčiami pranešimai.

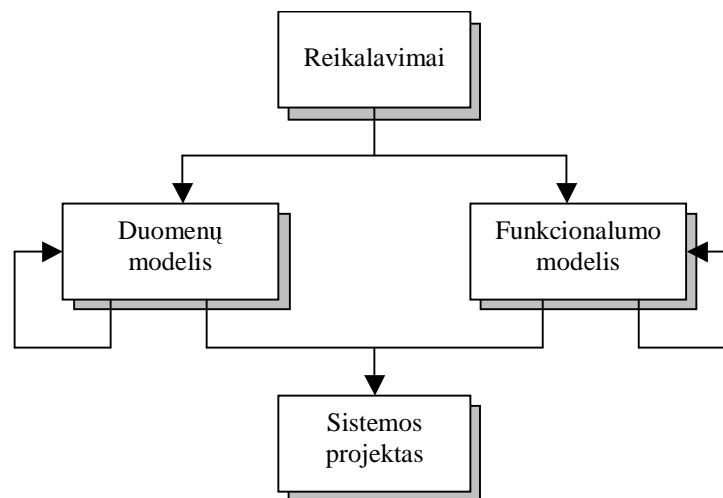
*Darbo struktūra.* Šis darbas sudarytas iš 2 pagrindinių dalių: teorinės dalies ir praktinės realizacijos. Teorinėje dalyje apžvelgiamos šiuo metu egzistuojančios panašios sistemos, analizuojamas sistemos funkcionalumas, vartotojų vykdomi procesai, projektuojama duomenų bazė. Praktinėje dalyje pateikiamas realiai veikiančios sistemos pavyzdys su visais reikalavimais bei funkcionalumu.

## 1. Analizės dalis

Šioje darbo dalyje suprojektuota „Mobili informavimo sistema“, skirta Kauno technologijos universiteto dėstytojams bei studentams. Darbo teorinėje dalyje išanalizuotos panašios sistemos, sudaryti sistemos analizės modeliai, duomenų bazės modelis, ištirtas funkcionalumas.

### 1.1 Informacinių sistemų analizės metodai ir priemonės

Daugumos projektavimo metodologijų branduolys [5] yra parodytas 1 paveiksle



1 pav. Projektavimo metodologijų branduolys

Projektavimo procesas prasideda kruopščiu reikalavimų dokumentavimu, po to pereinama prie dviejų klasikinių modelių: duomenų ir funkcijų. Rodyklės prie kiekvieno modelio parodo, kad projektavimo procesas yra iteratyvus. Antrajame projektavimo etape naudojamos formalios metodikos ir CASE priemonės, kuriomis galima kurti diagramas, jas spausdinti ir nagrinėti, bet svarbiausia – greitai ir lengvai jas keisti. Po iteratyvaus projektavimo etapo gaunamas taikomosios sistemos projektas.

Čia suformuluotos pagrindinės projektavimo proceso problemos:

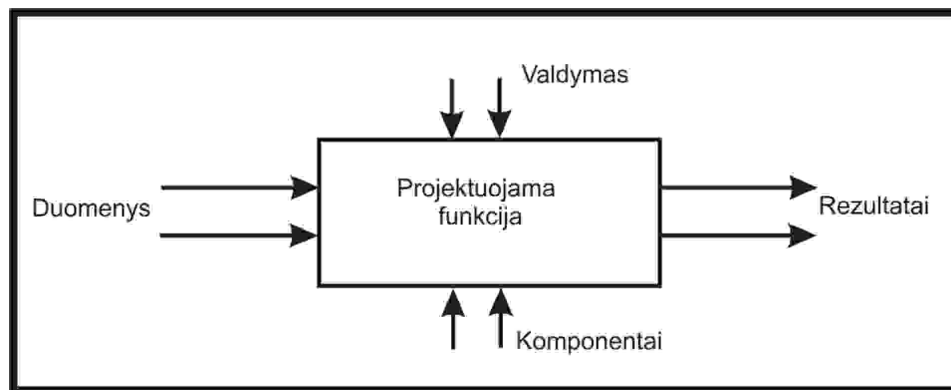
1. Kada, kur ir kaip spręsti duomenų apdorojimo paskirstymo klausimus?
2. Kur šiame procese yra vartotojo sąsajos projektavimo vieta?
3. Kaip valdyti projektavimo procesą ir gauti lankstų projekto variantą, kurį galima būtų lengvai keisti?

Taigi, vartotojo sąsajos projektavimo klausimą autorius laiko viena iš trijų pagrindinių informacijos sistemų projektavimo problema.

### 1.1.1 Integration Definition metodų šeima

Šis metodas yra vienas iš daugelio, leidžiančių projektuoti informacinę sistemą, atlikti jos analizę ir grafiškai atvaizduoti visus sistemoje esančius ryšius, galimus veiksmus [8]. Integration Definition (IDEF) yra ne vienas metodas, o metodų šeima. Ją sudaro 6 metodai (IDEF0, IDEF1, IDEF1x, IDEF3, IDEF4, IDEF5). Trumpai apžvelgsime kiekvieną iš jų.

Pirmasis iš jų - **IDEF0**. Tai metodas, sukurtas sistemų ar organizacijų sprendimų, veiksmų ir veiklų modeliavimui. Jis sukurtas Structured Analysis and Design Technique (SADT) grafinės kalbos pagrindu, todėl yra gerai suprantamas ir projektuotojui ir užsakovui. IDEF0 metodas orientuotas į funkcinę analizę. Šis metodas leidžia aprašyti visas reikalingas sistemos funkcijas ir tai, kas reikalinga šių funkcijų atlikimui. Pagrindinis modelio elementas pavaizduotas 2 pav.



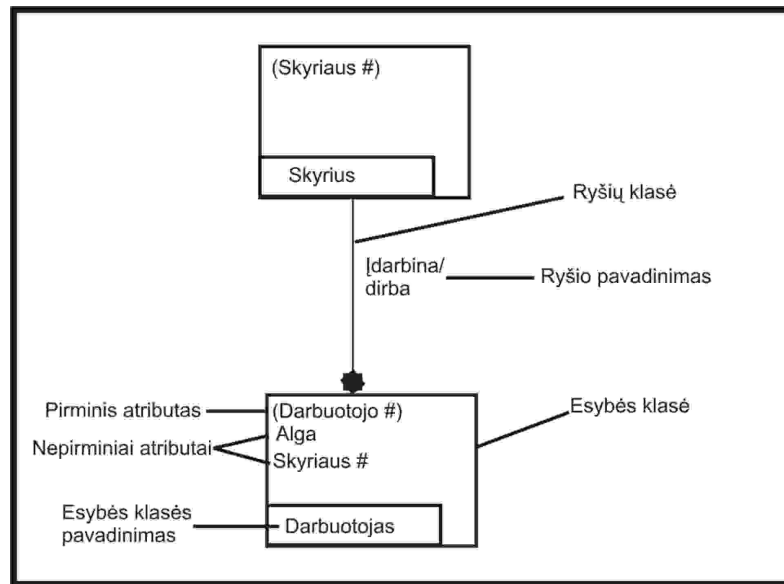
2pav. Pagrindinis IDEF0 modelio elementas

Šiame metode funkcijos yra atskirtos nuo organizacijos, tai leidžia sukurti detalesnį modelį, jis palengvina „iš viršaus žemyn“ interpretuojamų, tačiau analizuojamų „iš apačios į viršų“, modelių kūrimą. Šis metodas nepalaiko proceso specifikacijų. IDEF0 metodas turi ir trūkumų tokių kaip modelių trumpumas, dėl šios priežasties jų analizės ir projektavimo procesai apsunkinami. Be to, modeliai gali būti nekorektiškai interpretuojami kaip veiklų sekos, kurių IDEF0 nepalaiko.

Kitas informacijos modeliavimo metodas yra **IDEF1**, skirtas sistemos ar organizacijos reikalavimų analizei bei jų perdavimui. Metodas naudojamas nustatyti:

- nustatyti, kokia informacija tuo metu yra valdoma organizacijoje;
- išsiaiškinti, kokios iškyla problemos esant informacijos trūkumui;
- specifiuoti, kokia informacija bus valdoma ateityje.

Vienas iš šio metodo tikslų yra sudaryti struktūrizuotą ir paremtą griežtomis taisyklėmis procesą, turimai informacijai analizuoti. Žemiau sekanti diagrama iliustruoja, koku būdu braižomi IDEF1 modeliai (3pav.).



3 pav. IDEF1 diagrama

Naudojant šį metodą, galima sukurti modelius, kurie suformuoja informacijos valdymo taisykles. Diagramų objektai jungiami loginiais ryšiais. Metodo grafiniuose modeliuose naudojamos taisyklės atvaizduoti ir išskirti realaus pasaulio objektus, fizinius bei abstrakčius ryšius tarp šių objektų, turimą informaciją apie juos.

Reliacines duomenų bazes su sintakse, tinkama kurti koncepcines schemas, galima modeliuoti **IDEF1x** metodu. IDEF1x sistemos perspektyva fokusuojama į konkrečius duomenų elementams reliacinėje duomenų bazėje. Šis metodas netinka modeliuoti objektiškai orientuotas duomenų bazes, kadangi jame būtina reikia sukurti raktinę klasę tam, kad atskirti vieną esybę nuo kitos. Kuomet daugiau kaip vienas atributas gali individualizuoti IDEF1x esybę, yra būtina vienai iš jų priskirti pirminį raktą, o kitus išvardinti kaip alternatyvius raktus. Pagrindinė IDEF1x savybė yra ta, kad šis modeliavimo procesas yra griežtai standartizuotas, todėl nėra galimybės modeliuoti skirtingais būdais ar padaryti klaidų.

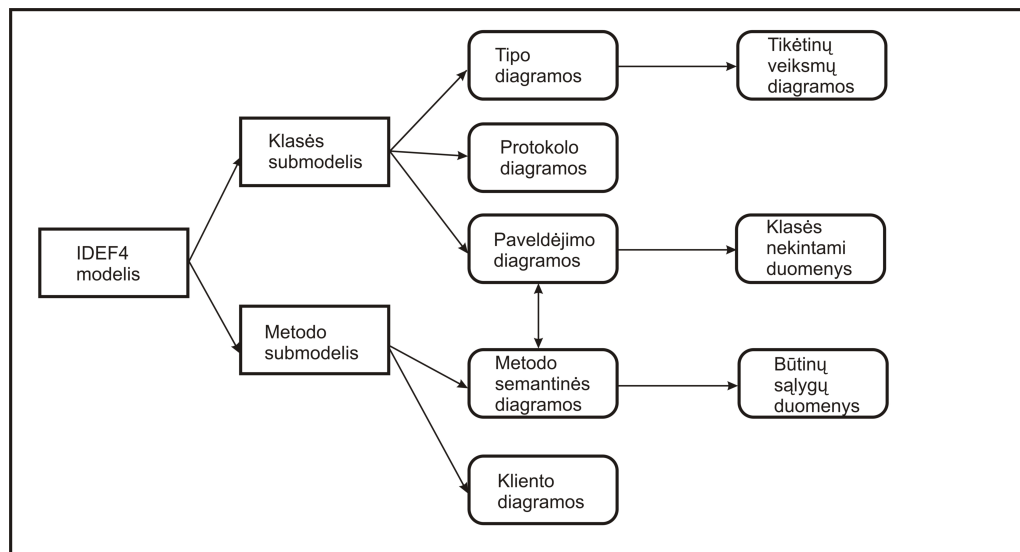
Jei reikia modeliuoti procesus bei juos aprašyti, galima naudoti **IDEF3** modeliavimo metodą. Šiuo metodu galima atvaizduoti pirminius reiškinius, priežastinius ryšius tarp situacijų bei įvykių struktūrizuoti. Naudojantis juo, galima:

- nustatyti organizacijos informacijos šaltinių įtaką įmonės operacijų eigai,
- valdyti informaciją ir keisti jos kontrolės sistemą,
- sudaryti simuliacinius modelius,
- atvaizduoti sprendimų procedūras, kurios turi įtaką gyvybiškai svarbiai informacijai.

Šiuo metodu galima sukurti struktūrizuotą sistemos apibūdinimą, iš kurio galima spręsti, ką sistema iš tikrųjų daro ar darys ir kaip ji elgiasi įvairiose situacijose.



Jei sistema yra projektuojama objektiškai orientuota programavimo kalba, tai geriausias būdas modeliuoti **IDEF4** metodu. Naudojantis šiuo metodu, kiekviena veikla yra išskaidoma į atskirus komponentus. Nei viename IDEF4 metodo modelyje nėra pilnos informacijos apie visą sistemą, tad norint išvengti klaidų modeliuojant, turi būti padaryta išsami visos sistemos analizė. Naudojantis šia modeliavimo priemone, galima nustatyti ryšius tarp klasių, klasių paveldėjimumo, funkcijų skaidymosi ir polimorfizmų. Bendra šio metodo modelio struktūra pateikta žemiau esančioje diagramoje (4 pav.). Bendru atveju IDEF4 modelis susideda iš dviejų submodelių – klasių ir metodų, kas padeda detaliau išanalizuoti visą struktūrą.



4 pav. IDEF4 modelio struktūra

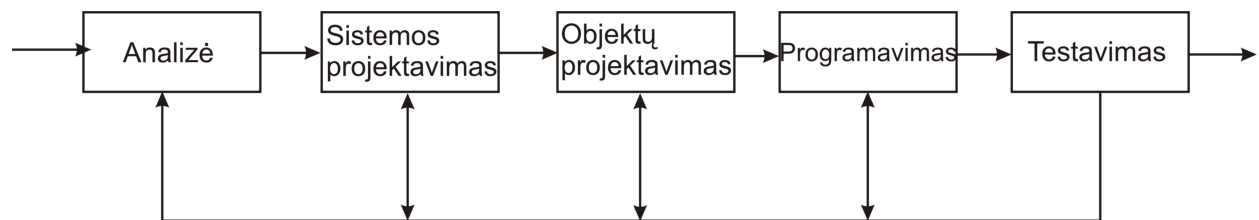
Sistemos ontologinei analizei yra sukurtas **IDEF5** metodas. Modeliavimo procesą sudaro 5 pakopos:

- Organizavimas;
- Duomenų rinkimas;
- Duomenų analizė;
- Pirminis ontologinis projektavimas;
- Ontologinis tobulinimas ir patvirtinimas.

Šiame metode naudojamos dvi modeliavimo „kalbos“: schematinė ir detalizavimo. Schematinė kalba yra grafinė, naudojama pateikti dažniausiai pasikartojančias ontologinės informacijos formas. Kita kalba – struktūrizuota, tekstinė, leidžianti detalizuoti konkrečius elementus.

### 1.1.2 Objektinio modeliavimo metodas

Objektinio modeliavimo metodas (Object Modeling Technique OMT) naudojamas sistemų ir programinės įrangos objektiniam modeliavimui [12]. Jis apima trijų tipų modelius: objektų, dinaminį ir funkcinį, kurie tarpusavyje susiję. Objektų modelis aprašo objektus, klase bei jų ryšius, dinaminis modelis – sąveikas tarp objektų klasių, funkcinis modelis – sistemoje vykstančias duomenų transformacijas. Kiekvienas iš šių trijų modelių susideda iš kelių diagramų, tai leidžia smulkiai išskaidyti problemą ir ją korektiškai atvaizduoti diagramoje. Šiame metode visi naudojami objektai yra identifikuojami – turi vardą, juos galima atskirti vieną nuo kito. Objektai jungiami į klases, turinčias savo atributus, savybes, operacijas. Modeliavimo procesas, naudojant OMT metodą, pavaizduotas 5 paveiksle.



5 pav. OMT modeliavimo procesas

Šis metodas dar ypatingas tuo, kad galima objektus rūšiuoti, įgyvendinti klasių savybių paveldėjimą.

### 1.1.3 Objektiškai orientuoto proceso, aplinkos ir notacijos metodas

Objektiškai orientuoto proceso, aplinkos ir notacijos metodas (angl. Object-oriented Process, Environment and Notation), toliau OPEN metodas [13], apima visus būtinus metodui atributus, ne tik modeliavimo kalbą ir metamodelį, kaip UML. OPEN metodo esmė yra ryšiai tarp užduočių (Tasks) ir gyvavimo ciklo veiklų (Activities). Šiame metode apibrėžtos veiklos (activities), suskirstytos į etapus:

- 1) Biznio organizavimo etapas (business build stage);
- 2) Realizavimo etapas (build phase);
- 3) Įdiegimo etapas (delivery phase).

### 1.1.4 Unifikuota modeliavimo kalba

Unifikuota modeliavimo kalba (ang. Unified Modeling Language), toliau UML - labiausiai šiuo metu paplitusi projektavimo ir modeliavimo priemonė [11]. Ši modeliavimo kalba susideda iš 7 skirtingų tipų diagramų, skirtų palapsniam perėjimui nuo uždavinio koncepcijos

prie jo formalios specifikacijos, aprašančios tiek duomenų bazės sandarą, tiek uždavinio veiksmų sekas ir objektų sandarą.

- 1) Klasių (class) diagrama – tai grafas, susidedantis iš sistemos elementų, sujungtų įvairių tipų statiniais ryšiais. Klasių diagrama parodo modelio statinę sandarą, t.y. elementus, jų vidinę sandarą ir ryšius su kitais elementais. Klasių diagrama nevaizduoja laikinos informacijos, tačiau gali turėti elementų, kurie aprašo laikiną elgseną. Klasių diagrama gali būti naudojama vartotojo sąsajos langų elementinei sudėčiai specifikuoti.
- 2) Panaudojimo atvejų (use case) diagrama – tai sistemos aktorių ir panaudojimo atvejų ryšių sandara. Panaudojimo atvejai parodo sistemos, posistemės arba klasės funkcionalumą, kuris pasireiškia per išorės santykius su sistema, posisteme arba klase. Panaudojimo atvejų diagrama gali parodyti atliekamas funkcijas per projektuojamos sistemos vartotojo sąsają.
- 3) Sekų (sequence) diagrama parodo sistemos objektų sąveiką laiko atkarpomis, o ypač jų dalyvavimo tvarką ir seką visame gyvavimo cikle ir veikiančius stimulus, išdėstytus laiko tėkmėje. Ši diagrama nerodo ryšių tarp objektų.
- 4) Bendradarbiavimo (collaboration) diagrama parodo sąveiką tarp konkrečių aktorių ir tarpusavio ryšius. Skirtingai nei sekos diagramoje, ši diagrama parodo ryšius tarp objektų vykdančių skirtingus vaidmenis, tačiau nerodo laiko kaip atskiros dimensijos, todėl sąveikos seka ir lygiagretūs procesai turi būti nustatyti vartojant sekos numerius.
- 5) Būsenų (statechart) diagrama aprašo modelio elementų, t.y. objektų ir sąveikų, elgseną. Konkrečiai ji aprašo galimas būsenų sekas ir veiksmus, kuriuos elementas gali pereiti per savo gyvavimo ciklą, reaguodamas į diskrečius įvykius.
- 6) Veiklos (activity) diagrama – tai veiklos grafas, parodantis būsenas, prieš vykdamas veiksmus, ir jų pokyčius, sukeltus pasibaigusiu veiksmu. Ši diagrama taip pat parodo procedūros būsenų kombinacijas. Ši diagrama gali būti panaudota aprašyti sistemos uždavinių veiksmų sekai, kuri realizuojamai per vartotojo sąsają.
- 7) Įgyvendinimo (implementation) diagrama parodo realizavimo aspektus, įtraukiant išėties tekstų sandarą ir vykdymo realizacijos sudėtį. Ji gali būti dviejų formų:
  - a) sudėtinių dalių (component) diagrama parodo programinio kodo sandarą.
  - b) išdėstymo (deployment) diagrama parodo veikiančios sistemos sandarą.

UML metodikos pagrindas – anksčiau naudotos OOAD, OMT ir OOSE koncepcijos. Jos susideda iš diagramų, skirtų analizei, modeliavimui, projektavimui ir testavimui, tačiau iš vartotojo sąsajos modeliavimo ir projektavimo klausimų yra nagrinėjami tik elementinė ir funkcinė sudėtis bei veiksmų seka. UML metodika ir yra naudojama šiame darbe.

## **1.2 Literatūros šaltiniuose pateiktų sprendimų problemai spręsti lyginamoji analizė**

### **1.2.1 Dienotvarkės ir planavimo įrankis „Appointment Scheduler“**

„Appointment Scheduler“, viena iš panašių į mobilią informavimo sistemą. Ši sistema skirta planuoti ir tvarkyti dienotvarkę [14]. Tai yra vadinamoji Web – based aplikacija, todėl norint ja naudotis privaloma turėti interneto ryšį. Interneto svetainėje galima susikurti savo dienotvarkės tvarkymo padėjėją (assistant). Įvedus vartotojo vardą, elektroninį paštą ir slaptažodį sukuriamas internetinis nuoroda į dienotvarkę. Įvedus vartotojo vardą (elektroninis paštas) ir slaptažodį patenkama į dienotvarkės redagavimo aplinką. Dienotvarkė pateikiama metų dydžio (mėnesiai, dienos, laikai). Įvedus pažymėjimą apie įvyki leidžiama pasirinkti laiką, datą, įterpti trumpą aprašymą apie įvykį. Jeigu reikia, galima užduoti priminimo funkciją, kuri primins apie reikiamą įvykį elektroniniu paštu. Atskirai yra pateikiamas savaitgalių, švenčių, ne darbo dienų, atostogų planavimas.

Šioje sistemoje vartotojas (kuris sukūrė savo dienotvarkę) gali sukurti autorizuotą priėjimą kitiems asmenims, tai yra, jis gali sukurti vartotojo vardą ir slaptažodį kitam vartotojui, kad šis galėtų peržiūrėti jo dienotvarkę. Galimas sukurtų vartotojų šalinimas. Interneto svetainėje taip pat yra trumpas pagalbinis aprašymas nežinančiam kaip elgtis.

Ši sistema yra tik asmeninės dienotvarkės tvarkymo įrankis, vartotojai kuriems suteikiama teisė, gali peržiūrėti dienotvarkę, bet jie nebus informuoti apie tai. Tai pagrindinis skirtumas lyginant su „Mobilia informavimo sistema“, kurios pagalba sistemos vartotojai yra informuojami apie dienotvarkės pasikeitimus.

### **1.2.2 Kalendorius „Web Calendar“**

„Web Calendar“ yra „ProVerbs“ kompanijos pateikiamas sprendimas įgyvendintas PHP ir MySQL priemonėmis [15]. Tai paprastas kalendorius, kurį gali žiūrėti bet kas prisijungęs prie sveitainės kurioje jis patalpintas. Kalendorius yra pateikiamas mėnesio pavidalo, administratorius įveda visus įvykius, kuriuos vartotojai mato naršyklėje. Sistema yra labai paprasta ir nefunkcionala. Ji pateikiama PHP failuose, kuriuos reikia įdiegti į serverį. Tada yra sukuriamas duomenų bazė, kurioje saugojami įrašai apie įvykius. Norint, kad sistema veiktų reikalingas PHP ir MySQL palaikymas. Dienotvarkę gali tvarkyti tik vienas asmuo, negalima išskirstyti vartotojų, kad jie patys suvestų reikalingus įvykius. Sistema patogi tik tuo, jei reikia

įgyvendinti smulkų darbų planavimo kalendorių, tai tinka mažoms firmoms. Reikėtų paminėti, kad tai atviro kodo projektas, kurį galima redaguoti ir plėsti, tačiau reikia visur palikti kūrėjų logotipus.

### **1.2.3 Dienotvarkės tvarkyklė „Flex 7“**

Internetu pagrįsta (Web - based) aplikacija „Flex 7“, siūloma kompanijos „Time Management“, skirta tvarkyti dienotvarkę [6]. Kaip ir kitos anksčiau paminėtos sistemos, taip ir ši, yra skirta tvarkyti tik asmeninę dienotvarkę. Bet reikia pastebėti, kad ji skiriasi nuo kitų. Visų pirma ši dienotvarkė turi lietuviybę, tai yra visos savaitės dienos mėnesiai, įrašai gali būti pateikiami lietuvių kalba, reikia tik išsirinkti atitinkamą meniu punktą, kuomet yra renkami duomenys. Kalendorius yra pateikiamas mėnesine forma. Galimas posėdžių, darbo susitikimų, švenčių, priminimų (reminder) planavimas ir koregavimas.

Visas valdymas pagrįstas dešiniuoju pelės mygtuku iškviečiamo meniu valdymu. Skirtingai nuo kitų sistemų šioje yra įdiegta SMS priminimo paslauga. Šioje svetainėje registracija yra nemokama, reikalinga mokėti už SMS, jeigu viršijamas leidžiamas limitas 2 SMS per mėnesį. Ši sistema turbūt panašiausia i Mobiliją Informavimo Sistemą, nes joje yra įdiegta SMS priminimo – informavimo funkcija. Šioje sistemoje negalimas kitų vartotojų prisijungimas, tai reiškia, kad tik vienas asmuo gali matyti dienotvarkę.

### **1.2.4 Dienotvarkės planavimo sistema „Appointment Scheduler“.**

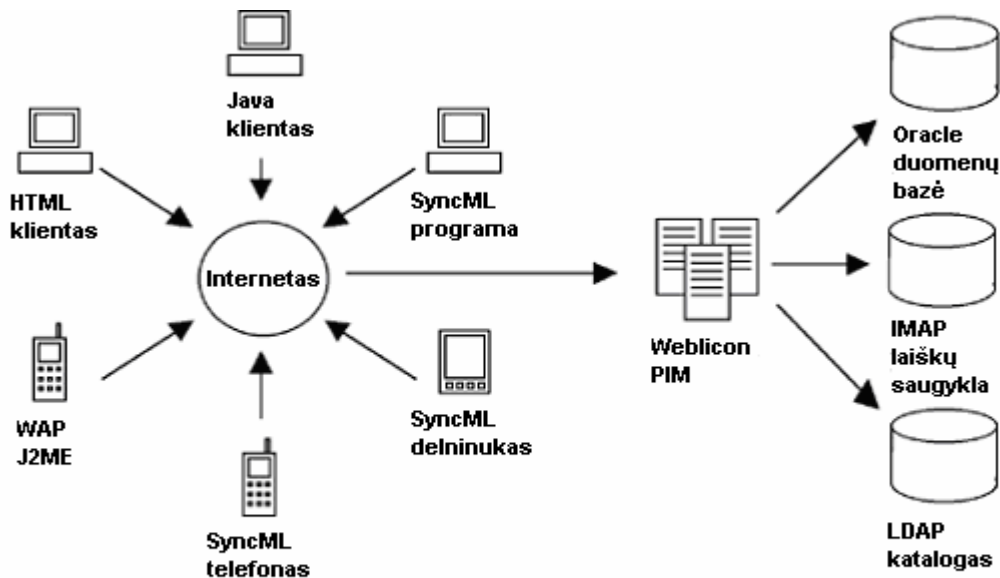
Šis produktas - dienotvarkės planavimo sistema. Kūrėjai ją platina su nemokamu 60 dienų naudojimo laikotarpiu, po to programą reikia įsigyti arba išinstaliuoti. Ši aplikacija patogiausia gydytojams, advokatams, žmonėms, kurie turi daug darbo susitikimų. Galima suvesti informaciją apie asmenį su kuriuo ketinama susitikti, grupuoti suinteresuotus asmenis į grupes pagal tam tikrus požymius. Galima organizuoti paiešką, pagal daugelį pasirinktų kriterijų. Paieška vykdoma greitai nepaisant to kad įrašų gali būti labai daug, nes naudojama greita vietinė duomenų bazė[2]. Priminimas yra realizuotas elektroninio pašto pagalba, tai yra kai įvyksta įvykis, elektroniniu paštu išsiunčiamas pranešimas suinteresuotam asmeniui. Programa gali naudotis keli vartotojai, kurie gali peržiūrėti vienas kito dienotvarkes, tam reikia įsidiegti programos serverinę versiją.

### 1.2.5 Internetu pagrįsta programa „Super Calendar“

Dar viena internetu pagrįsta aplikacija - „Super Calendar“, skirta tvarkyti dienotvarkes. skirtingai nei kitos sistemos, kurios skirtos tvarkyti tik asmenines dienotvarkes, šioje sistemoje vartotojus galima jungti į grupes, tai yra, kiekvienas vartotojas gali įtraukti į sąrašą kitą sistemos vartotoją, kuris galės matyti pastarojo dienotvarkę arba tiesiog pasirinkti opcijose, kad jo dienotvarkę matytu visi. Sistemos valdymas nedaug kuo skiriasi nuo anksčiau aprašytų Web aplikacijų. Kalendorius yra mėnesinės formos, galima įtraukti įvairius įvykius ir priminimus. Priminimai gali būti siunčiami tik elektroniniu paštu. Apie dienotvarkės pasikeitimus kiti asmenys nėra informuojami. Apie pasikeitimus kiti suinteresuoti asmenys gali sužinoti tik apsilankę tinklapyje.

### 1.2.6 Profesionali dienotvarkės valdymo sistema „Weblicon“ PIM

„Weblicon“ PIM, tai profesionaliausia iš prieš tai aprašytų dienotvarkės valdymo sistemų [18]. Į ją įeina adresų knygtė, kalendorius, „reikiamų atlikti“ darbų sąrašai integruotos EMAIL, GROUP SMS, WAP, MMS funkcijos. Multi architektūra leidžia prisijungti prie sistemos naudojant HTML, WAP, JAVA klientus. Yra numatyta integracija su Microsoft Outlook ir PDA (delninukais). Naudojantis aplikacija galima tikrinti elektroninį pašta, tam naudojamas tekstas-į-kalbą metodas ir balso atpažinimas.



6 Pav. Weblicon PIM veikimo schema

Weblicon HTML dalis yra pasiekama iš bet kokio nutolusio kompiuterio naudojantis interneto naršykle. Vartotojai gali jungtis prie savo asmeninių duomenų iš bet kokio nutolusio kompiuterio.

Adresų knygutę galima rūšiuoti pagal įvairius atributus.

Naudojant kontaktų grupės vaizdą galima sudaryti sąrašus, kuriais išsiunčiamos trumposios SMS žinutės ar kitaip informuojami suinteresuoti asmenys.

Pranešimų modulis suteikia pilną internetinio pašto (webmail) klientą, taip pat hierarchinę katalogų struktūrą saugoti pranešimus centriniame IMAP elektroninio pašto serveryje.

Naudojantis WAP klientu, vartotojai turi galimybę prieiti prie duomenų iš bet kokio WAP protokolą palaikančio telefono. Weblicon WAP klientas teikia pilną galimybę prieiti prie vartotojo duomenų įskaitant elektroninį paštą ir SMS žinutes, adresų knygelę ir kalendorių. Vartotojai gali ne tik keisti savo kalendoriuje įrašus, taipogi galimas elektroninio pašto ir SMS siuntimas.

Integruotas JAVA klientas leidžia naudotis aplikacija paprastame PC kompiuteryje. Ji užkraunama kaip paprasta programa, kuri jungiasi prie pagrindinio serverio naudodama TCP/IP protokolą.

Java klientas veikia visai kaip standartinė aplikacija su visomis „tempk ir mesk“ funkcijomis, taigi jis yra puikiai suprantamas paprastiems kompiuterių vartotojams.

J2ME klientas garantuoja, kad prie sistemos galima prisijungti naudojant delninį kompiuterį. Kaip ir per WAP galima siųsti elektroninį paštą, SMS, tvarkyti kalendorių su dienotvarkę.

Taip pat yra numatytas prisijungimas iš paprasto telefono, tam yra naudojamas balso atpažinimas, yra palaikomas vyro arba moters balso atpažinimas taip pat palaikomos anglų, prancūzų, vokiečių kalbos.

Alternatyvių sistemų palyginimų lentelė

1 lentelė

Sistema	Reikalingas interneto ryšys	Kiti vartotojai gali peržiūrėti dienotvarkę	Apsaugota slaptažodžiu	Yra priminimo funkcija	Yra galimybė informuoti
Appointment Scheduler“	Taip	Taip	Taip	Taip	Ne
Web Calendar	Taip	Taip	Ne	Ne	Ne
„Flex 7“	Taip	Ne	Taip	Taip	Taip
CyberMatrix Pro Sched™: Appointment Scheduler	Taip	Taip	Taip	Taip	Taip
„Super Calendar“	Taip	Taip	Taip	Ne	Ne
„Weblicon“	Taip	Taip	Taip	Taip	Taip

### 1.2.7 Išvados.

Apžvelgus šias sistemas padaryta išvada, kad tokių pačių informavimo sistemų nėra sukurta. Todėl Mobili informavimo sistema yra unikalus projektas, apjungiantis apžvelgtų aplikacijų įvairias savybes. Sistema sukurta Internetu pagrįsta technologija, kurioje yra integruotas darbo kalendorius, galimybė redaguoti kalendorių, skubus suinteresuotų asmenų informavimas mobiliomis ir/ar internetinėmis technologijomis apie dienotvarkės pasikeitimus, išpildytas saugus prisijungimas prie sistemos, greita informacijos paieška duomenų bazėje paprasta vartotojo sąsaja.

## 1.3 Organizacijos veiklos analizė

### 1.3.1 Sistemos kokybės kriterijai

Mobilios informavimo sistemos kokybės kriterijai [3, 7, 9]:

1. Tinkamumas – ar sistema pakankamai patogi ir tinkama naudotis jai keliamiems reikalavimams;
2. Sąveika su kitomis jau egzistuojančiomis savybėmis – ar sistema nekliudo jau veikiančiai universiteto informacinei sistemai;
3. Saugumas - saugus prisijungimas ir atsijungimas nuo sistemos;
4. Patikimumas – ar sistema užbaigta, ar yra tolerancija klaidoms, kaip lengvai atstatoma;
5. Patogumas –vartotojo sąsajos intuityvumas, patrauklumas;
6. Efektyvumas – sistemos veikimo greitis, kompiuterio resursų naudojimas;



7. Priežiūros savybės – kaip lengvai sistema prižiūrima, testuojama, stabili;
8. Perkeliamumas – kaip lengvai sistemą galima perkelti iš vienos vietos į kitą, ar sunkiai įdiegiama

### 1.3.2 Projektavimo metodai ir priemonės

Magistrinio darbo projektavimo priemonė pasirinkta „Rational Software“ kompanijos kūrinys „Rational Rose Enterprise Edition“ [16]. Šis produktas pasirinktas dėl to, kad UML pagrindus suformulavo Rational Software Corporation ir trys žymiausi informacinių sistemų ir technologijų industrijos metodologai: Grady Booch, James Rumbaugh ir Ivar Jacobson. Modeliavimo programa visiškai palaiko UML modeliavimo kalbą ir turi visas 9 diagramas: klasių, objektų, panaudojimo atvejų, sekų, bendradarbiavimo, būsenų diagrama, veiklos, komponentų bei įrangų. Šių diagramų visiškai pakanka sumodeliuoti bet kokia organizacijos veiklai, verslo procesus ir kita.

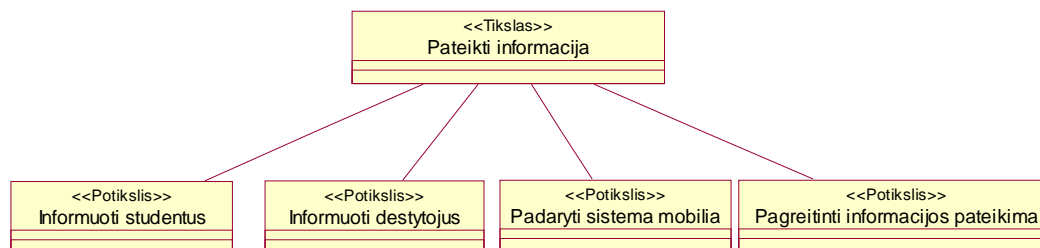
## 1.4 Organizacijos veiklos analizė

### 1.4.1 Analizės tikslas

Organizacijos analizė daroma tam, kad iš anksto išsiaiškinti visus sistemos vartotojus, jų veiksmus, su jais susijusias klases, apibrėžti operacijas ir sąryšius tarp klasių. Taip pat reikia išsiaiškinti visus sistemos veiklos atvejus tam, kad realizuojant sistemą būtų galima išvengti klaidų.

### 1.4.2 Veiklos tikslų modelis

Visi organizacijos tikslai yra atvaizduojami tikslų modelyje (7 pav.). Organizacijos tikslai - visos veiklos, kuriais įgyvendinami organizacijos pagrindiniai uždaviniai.



7 pav. Veiklos tikslų modelis

Šiais organizacijos tikslais paremta ir „Mobilios informavimo sistemos“ struktūra bei jos funkcionalumas.

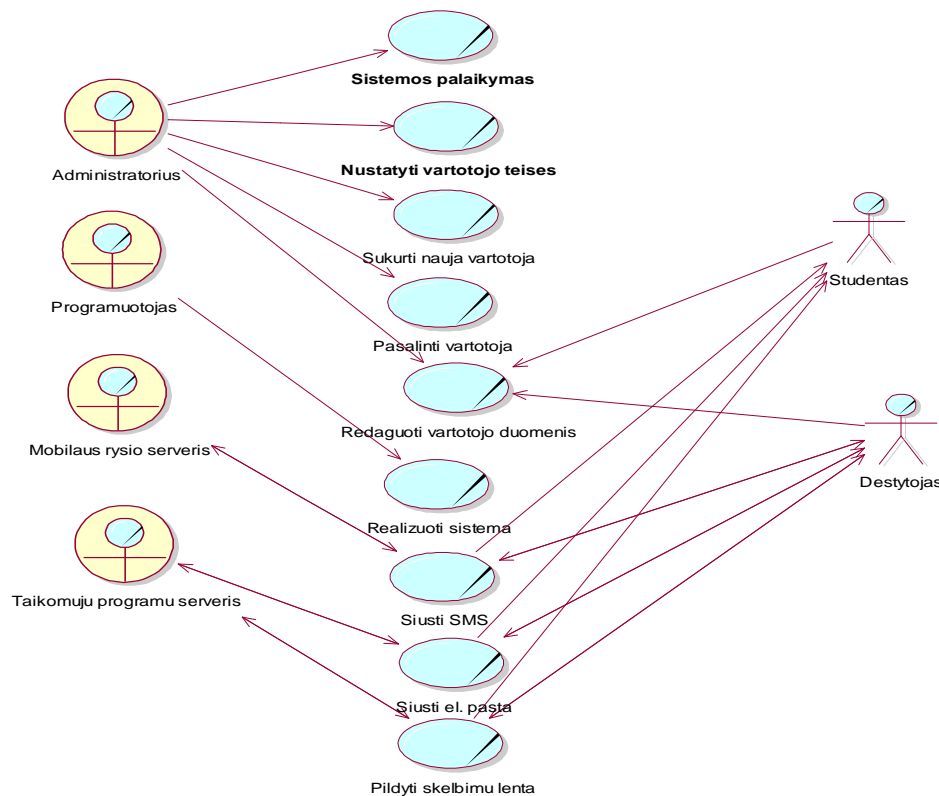
### 1.4.3 Veiklos procesų modelis

Iš veiklos panaudojimo atvejų modelio (8 pav.) matosi, kad „Mobili informavimo sistema“ skirta dėstytojams ir studentams [1]. Vartotojai skirstomi į dvi dalis, studentus ir dėstytojus. Studentai gali atlikti šiuos veiklos procesus:

- Gauti pranešimus iš dėstytojų apie planų pakeitimą.
- Peržiūrėti dėstytojo dienos tvarkę.
- Siųsti elektroninius paštus dėstytojams ir grupių, kurioms jis priklauso, studentams.
- Skaityti skelbimus
- Redaguoti savo asmeninę informaciją

Dėstytojai gali atlikti šiuos veiksmus:

- Sudaryti ir tvarkyti savo dienos tvarkę;
- Peržiūrėti kolegų dienos tvarkes;
- Siųsti pranešimus dėstytojams ir studentams apie planų pasikeitimą (pvz.: susitikimą, susitikimo perkėlimą į kitą laiką ir t.t.);
- Gauti pranešimus
- Įdėti naują skelbimą
- Redaguoti savo asmeninę informaciją

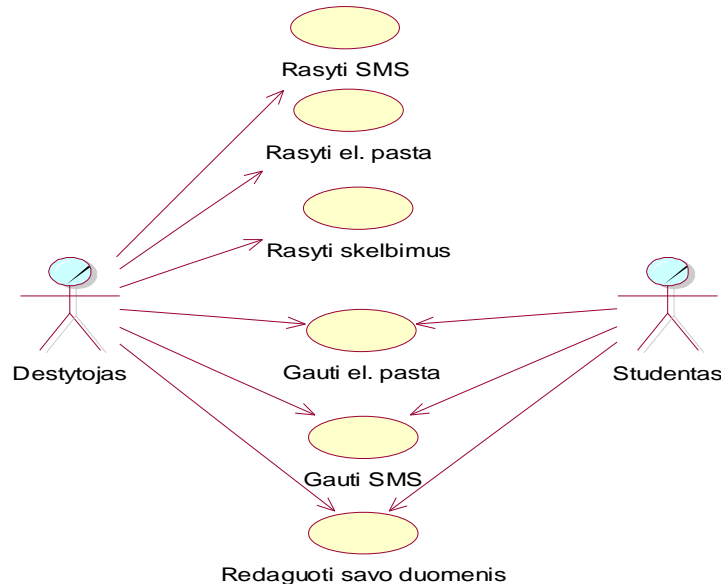


8 pav. Sistemos vartotojų panaudojimo atvejų modelis

Iš šios diagramos galima sudaryti dėstytojo bei studento panaudojimo atvejų modelį.

### 1.4.4 Studento ir dėstytojo veiklos procesas

Tam, kad sužinoti, kokias veiklas gali atlikti sistemos vartotojai, yra sudaroma studento ir dėstytojo panaudojimo atvejų modelis [ 4] . Šiame modelyje ( 9 pav.) aiškiai matyti, kad dėstytojas turi žymiai daugiau teisių ir gali atlikti daugiau veiksmų sistemoje, nei studentas.

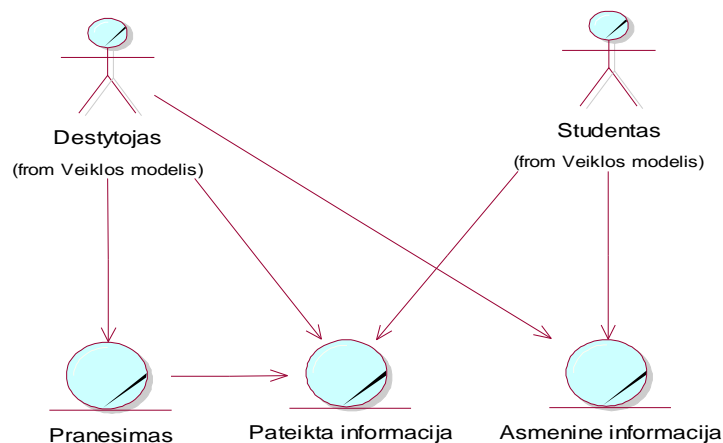


9 pav. Studento ir dėstytojo panaudojimo atvejų modelis

Tokiu principu yra atskiriami du vartotojų tipai bei jų teisės. Ši struktūra yra įgyvendinta realioje sistemoje.

### 1.4.5 Veiklos objektų modelis

Pagrindiniai veiklos objektai yra: dėstytojas ir studentas. Jie tarpusavyje dalinasi informacija(10 pav.). Asmeninė informacija yra pasiekama tik patiems vartotojams. Tokiu būdu yra užtikrinamas konfidencialumas.

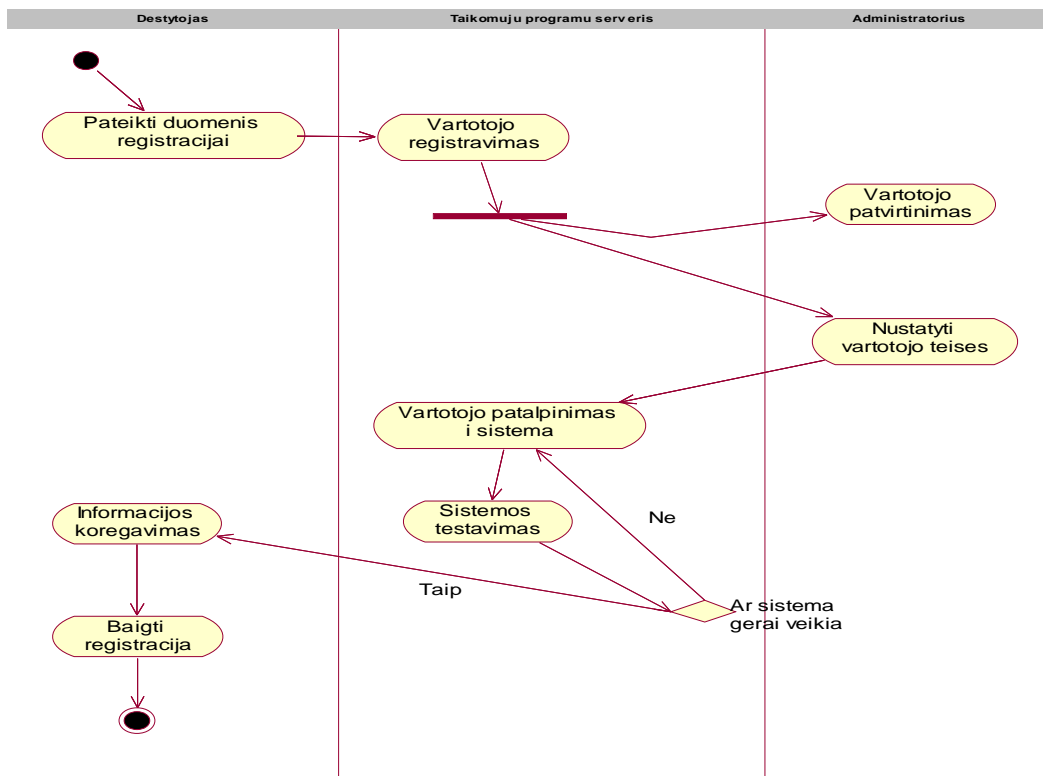


10 pav. Veiklos objektų modelis

Kadangi „Mobili informavimo sistema“ skirta tik informuoti, veiklos objektams nėra numatyta papildomų veiklų.

### 1.4.6 Veiklos procesų modelis

Šiame modelyje (11 pav.) matosi, kokie procesai ir kaip vyksta sistemoje. Šioje diagramoje pavaizduotas naujo vartotojo prisijungimas prie sistemos, duomenų apie naują vartotoją talpinimas į sistemą. Šiuo atveju dalyvauja ir sistemos administratorius, kuris tvirtina naują vartotoją. Vartotojo patvirtinimas reikalingas tik registruojantis dėstytojui, o studentui – ne, nes studentas šioje sistemoje turi ribotas teises.



11 pav. Veiklos procesų modelis

Veiklos procesų modelyje pavaizduoti visi sistemoje vykstantys procesai, jų eiliškumas bei sąlygos naujo vartotojo registravimo atveju. Tokiu principu yra užtikrinamas saugumas, nes ne bet kuris prie sistemos prisijungęs asmuo gali gauti dėstytojo teises.

### 1.5 Analizės išvados

Šiuo metu rinkoje egzistuoja panašių į „Mobilią informavimo sistemą“ sistemų, tačiau dauguma jų yra komercinės ir mokamos. Nemokamos sistemos nepasižymi dideliu funkcionalumu bei patogumu naudoti.

Šios sistemos analizės modeliai nubraižyti UML pagrindu. Iš visų modelių aiškiai matyti, kokius veiksmus gali atlikti sistemos vartotojai, sistemos objektų ryšiai, veiklos procesai. „Mobilios informavimo sistemos“ vartotojai yra 3 rūšių: sistemos administratorius, dėstytojai ir

studentai. Administratorius gali daryti tai, ko negali kiti – ištrinti vartotoją, redaguoti bet kurio iš jų duomenis. Dėstytojas turi žymiai daugiau teisių nei studentas, jis gali ne tik peržiūrėti pateiktą informaciją, bet ją pateikti ir pats, siųsti žinutes elektroniniu paštu bei trumpąja SMS žinute. Visi duomenys yra saugomi pagrindiniame serveryje, todėl jie yra apsaugoti. Kadangi sistema yra valdoma vien interneto naršykle, todėl papildomos programinės įrangos pas vartotoją nereikia įdiegti. Vartotojai gali matyti ir keisti tik savo asmeninę informaciją, tokiu būdu yra užtikrinamas konfidencialumas.

## 2. Sistemos struktūra ir realizacija

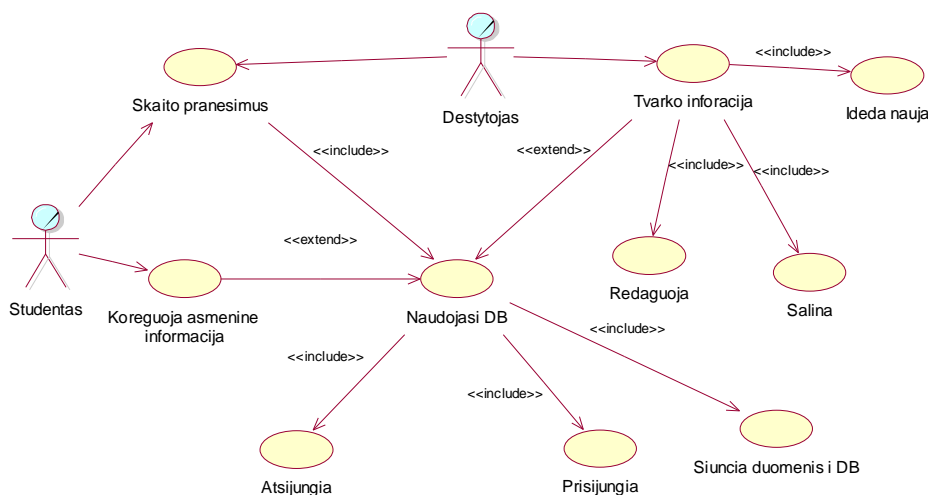
### 2.1 Kompiuterizuojamos sistemos varianto parinkimas

Šiame darbe bus kompiuterizuojamas informacijos pateikimas vartotojui, jos apdorojimas. Taip pat bus kompiuterizuojamas ir pats informavimas (siunčiami pranešimai elektroniniu paštu bei trumpąja SMS žinute), ir informacijos gavimas (sistemos vartotojai iš bet kurios vietos galės gauti reikalingą informaciją).

Sistemos kompiuterizavimas vyksta naudojant internetines ir mobiliąsias technologijas. Tam tikslui reikia duomenų bazės, taikomųjų programų bei mobilaus ryšio serverių. Klientinei daliai reikia kompiuterio, turinčio prieigą prie interneto, naršyklės bei mobilaus telefono.

### 2.2 Vartotojų panaudojimo atvejų modelis

12 paveiksle pateiktas vartotojų panaudojimo atvejų modelis. Iš šio modelio matyti, kad tik dėstytojas gali kurti pranešimus, tačiau ir studentas ir dėstytojas gali peržiūrėti kitų pedagogų tvarkaraščius, bei skelbimus.



12 pav. Vartotojų panaudojimo atvejų modelis

Šiame modelyje matosi visi vartotojų veiksmai, jų galimybės bei veiksmai sistemoje. Visa tai realizuota realioje sistemoje.

## 2.3 Specifikacijos sistemos panaudojimo atvejams

### 2.3.1 Naujo vartotojo registracijos panaudojimo atvejis

2 lentelė

<b>Panaudojimo atvejis</b>	<b>Naujo vartotojo registracija</b>
<b>Aktorius</b>	<b>Dėstytojas/ studentas</b>
<b>Sistema</b>	<b>Mobili informavimo sistema</b>
<b>Prieš sąlyga</b>	<b>Sistemos užsikrovimas</b>
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
<b>1. Užkraunama sistema interneto naršyklėje</b>	<b>1.1. Spaudžiama atitinkama nuoroda</b>
<b>2. Informacijos įvedimas/koregavimas</b>	<b>2.1. Suvedami reikalingi duomenys</b> <b>2.2. Atliekamas informacijos apdorojimas</b> <b>2.3. Gauta informacija patalpinama į duomenų bazę</b>
<b>3. Patvirtinimas</b>	<b>3.1. Vartotojas gauna patvirtinimą, kad jis sėkmingai priregistruotas sistemoje</b>
<b>Po sąlyga</b>	<b>Sistemoje užregistruojamas naujas vartotojas</b>
<b>Alternatyvos (nesėkmės atvejai)</b>	<b>Nepavyksta prisiregistruoti</b>
<b>Vykdyto variantai</b>	
<b>Veiklos taisyklės</b>	<b>Vartotojas privalo nurodyti teisingus duomenis</b>
<b>Specialūs reikalavimai (nefunkciniai)</b>	<b>Turi būti užtikrintas duomenų saugumas ir teisingumas</b>

### 2.3.2 Studentų ir dėstytojų informavimo panaudojimo atvejis

3 lentelė

<b>Panaudojimo atvejis</b>	<b>Studentų ir dėstytojų informavimas</b>
<b>Aktorius</b>	<b>Dėstytojas</b>
<b>Sistema</b>	<b>Mobili informavimo sistema</b>
<b>Prieš sąlyga</b>	<b>Suformuota užklausa</b>
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
<b>1. Prisijungiama prie sistemos</b>	<b>1.1. Tikrinamas vartotojo identiškumas</b> <b>1.2. Vartotojo teisių patvirtinimas</b>
<b>2. Informacijos įvedimas/koregavimas</b>	<b>2.1. Nustatomas užklauso pobūdis</b> <b>2.2. Atliekamas informacijos apdorojimas</b> <b>2.3. Gauta informacija patalpinama į duomenų bazę</b>
<b>Po sąlyga</b>	<b>Informacija išsiunčiama studentams pasirinktu būdu</b>
<b>Alternatyvos (nesėkmės atvejai)</b>	<b>Studentai negauna dėstytojo siųstos informacijos</b>
<b>Vykdyto variantai</b>	<b>Pranešimą galima siųsti elektroniniu paštu bei trumpąją SMS žinute, įrašyti į skelbimų lentą.</b>
<b>Veiklos taisyklės</b>	<b>Vartotojas turi nustatyti teisingus adresatus</b>
<b>Specialūs reikalavimai (nefunkciniai)</b>	<b>Turi būti užtikrintas duomenų saugumas ir teisingumas</b>

### 2.3.3 Asmeninės informacijos koregavimo panaudojimo atvejais

4 lentelė

<b>Panaudojimo atvejis</b>	<b>Asmeninės informacijos koregavimas</b>
<b>Aktorius</b>	<b>Dėstytojas/ studentas</b>
<b>Sistema</b>	<b>Mobili informavimo sistema</b>
<b>Prieš sąlyga</b>	<b>Asmeninių duomenų pasikeitimas</b>
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
<b>1. Prisijungiama prie sistemos</b>	<b>1.1. Tikrinamas vartotojo identiškumas 1.2. Vartotojo teisių patvirtinimas</b>
<b>2. Duomenų koregavimas</b>	<b>2.1. Koreguojami duomenys apie vartotoją 2.2. Atliekamas informacijos apdorojimas 2.3. Koreguota informacija patalpinama į duomenų bazę</b>
<b>Po sąlyga</b>	<b>Duomenų bazėje pakeičiami asmeniniai vartotojo duomenys</b>
<b>Alternatyvos (nesėkmės atvejai)</b>	<b>Nepavyksta pakeisti duomenų</b>
<b>Vykdomo variantai</b>	
<b>Veiklos taisyklės</b>	<b>Vartotojas gali keisti tik savo asmeninius duomenis</b>
<b>Specialūs reikalavimai (nefunkciniai)</b>	<b>Turi būti užtikrintas duomenų saugumas ir teisingumas</b>

### 2.3.4 Vartotojo ištrynimo panaudojimo atvejais

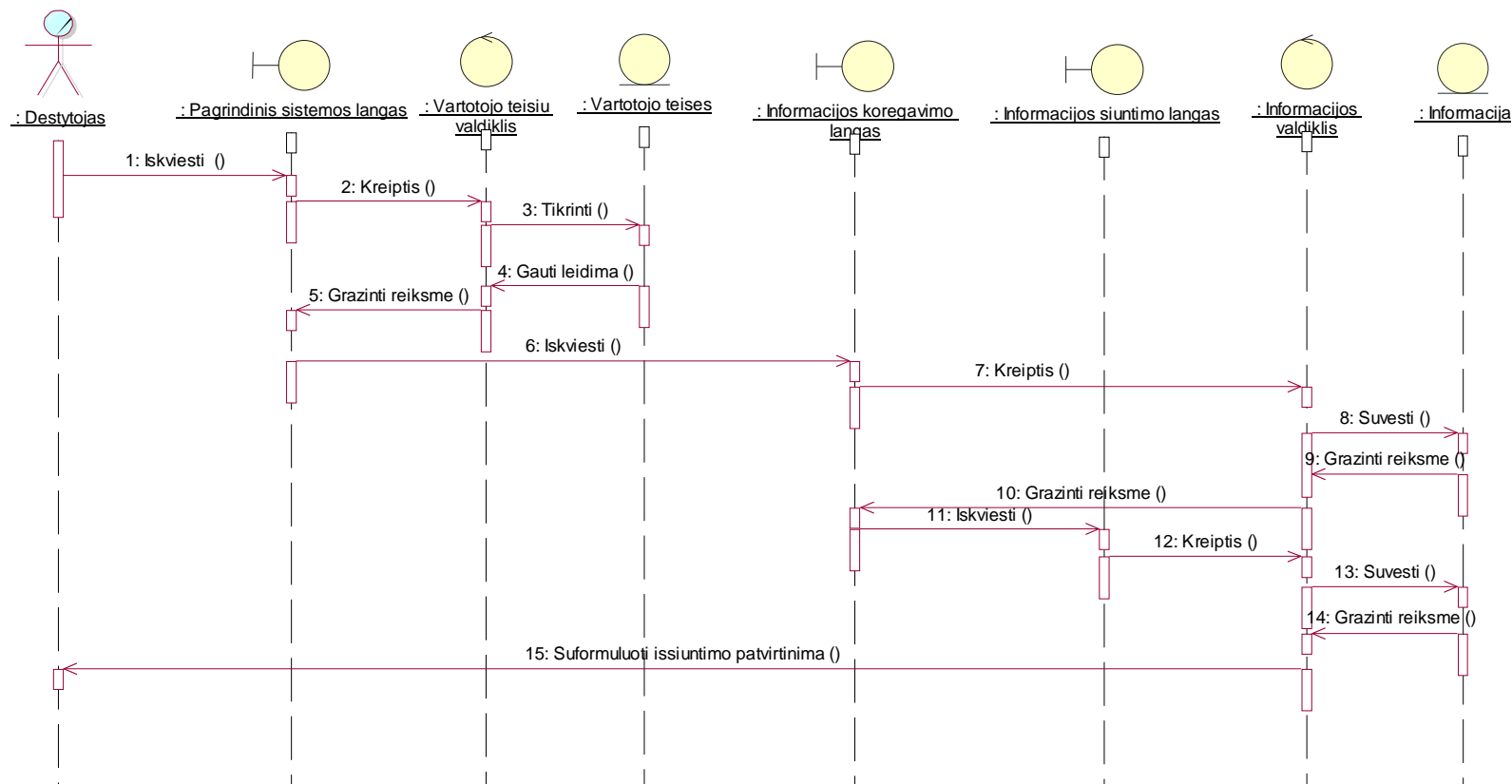
5 lentelė

<b>Panaudojimo atvejis</b>	<b>Sistemos vartotojo ištrynimas</b>
<b>Aktorius</b>	<b>Sistemos administratorius</b>
<b>Sistema</b>	<b>Mobili informavimo sistema</b>
<b>Prieš sąlyga</b>	<b>Vartotojas jau nebegali naudotis sistema</b>
<b>Pagrindinis įvykių srautas</b>	<b>Sistemos reakcija ir sprendimai</b>
<b>1. Prisijungimas prie sistemos</b>	<b>1.1. Tikrinamas vartotojo Identiškumas 1.2. artotojo teisių patvirtinimas</b>
<b>2. Vartotojo suradimas</b>	<b>2.1. Surandamas reikalingas vartotojas</b>
<b>3. Vartotojo ištrynimas</b>	<b>3.1. Vartotojo duomenys ištrinami iš duomenų bazės</b>
<b>Po sąlyga</b>	<b>Vartotojo duoemnys ištrinami iš duomenų bazės</b>
<b>Alternatyvos (nesėkmės atvejai)</b>	<b>Nepavyksta ištrinti vartotojo</b>
<b>Vykdomo variantai</b>	
<b>Veiklos taisyklės</b>	<b>Ištrinti vartotoją gali tik administratorius</b>
<b>Specialūs reikalavimai (nefunkciniai)</b>	<b>Turi būti užtikrinama, kad trinti galėtų tik administratorius</b>



## **2.4 vartotojo ir sistemos sekų modelis**

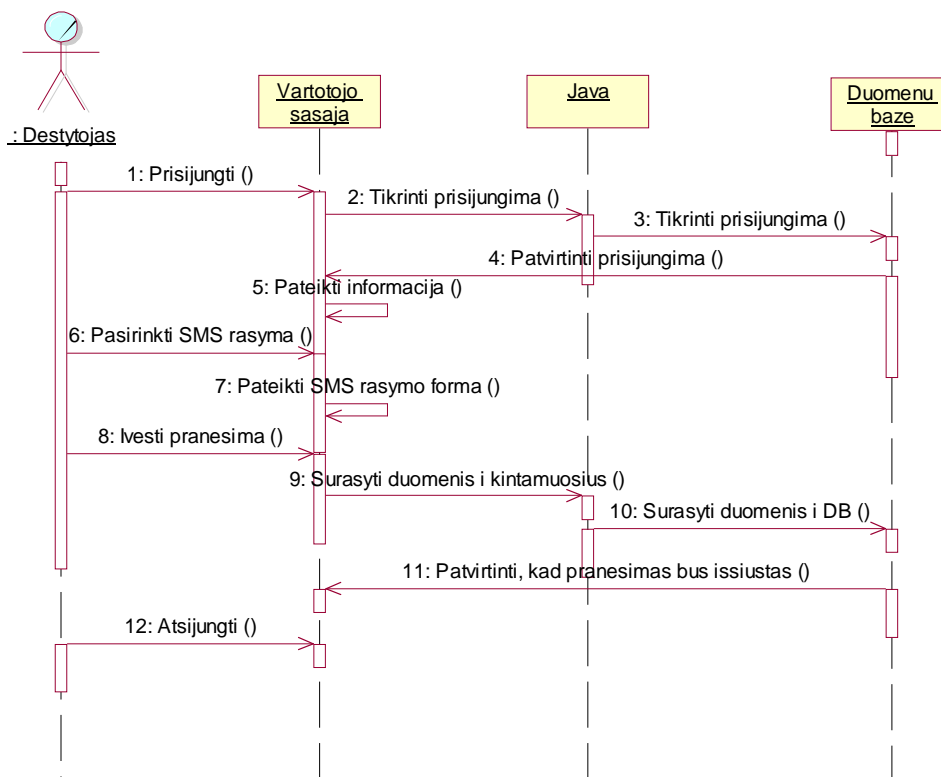
Sekų ir bendradarbiavimo diagramos yra sąveikų diagramų realizavimo variantai. Galima pastebėti ir panašius ryšius bei analogiškus pranešimus naudojamus jose. Tokios sąveikos diagramos apibūdina dinaminį realizuojamos sistemos vaizdą. Jei bendradarbiavimo diagramos skirtos kuo tiksliau struktūrizuoti objektus, kuriems sąveikaujant siunčiami pranešimai, tai sekos diagramos skirtos kuo tiksliau pavaizduoti operacijų ir jų tarpusavio sąveikos rezultatų – pranešimų, priklausomybę nuo laiko. Žinant, jog sekų ir bendradarbiavimo diagramos yra izomorfinės pagal išanalizuotos sistemos kriterijus sudaroma vartotojo ir sistemos sekų modelis (13 pav.).



13 pav. Dėstytojo ir sistemos sekų modelis

## 2.5 Dėstytojo SMS pranešimo sukūrimo sekų modelis

Kadangi UML sekų diagrama dažniausiai apibūdina vieną konkretų veiksmų planą tai joje galima pavaizduoti eilę bazinių objektų ir konkretizuotų pranešimų, kurie bus perduodami tarp objektų esant tam tikram panaudojimo atvejui. Pranešimo siuntimo modelyje ( 14 pav.) matosi visi dėstytojo ir sistemos veiksmai. Rezultatas – išsiųstas pranešimas.

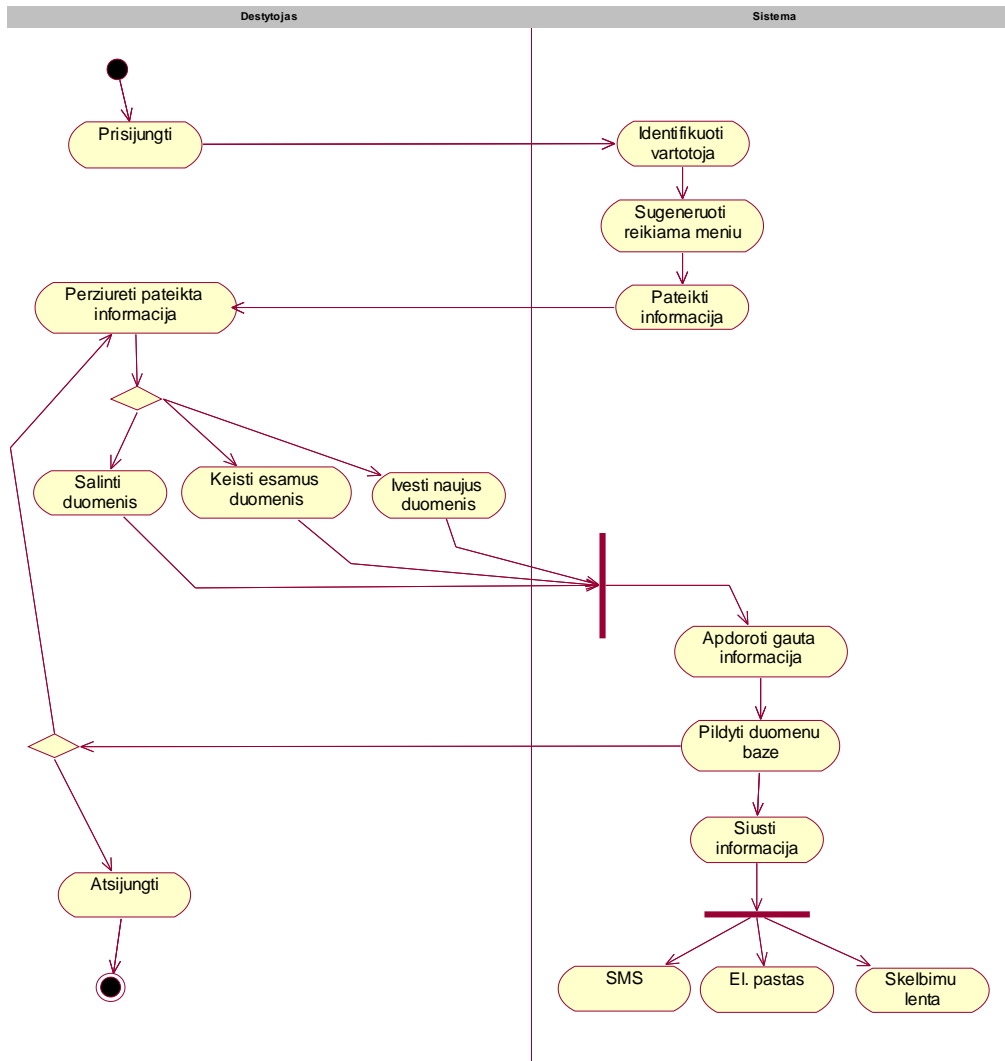


14 pav. Dėstytojo SMS sukūrimo sekų modelis

Šiame modelyje detaliam išdėstyti vis vartotojo bei sistemos veiksmai. Tai labai svarbu programuojant. Jei kuri nors iš pavaizduotų sekų yra negalima ar neveikia, tuomet sutrinka normalus sistemos darbas.

## 2.6 Sistemos elgsenos modelis, kai vartotojas dėstytojas

Sistemos elgsenos, kai vartotojas dėstytojas, (15 pav.) pavaizduota visos sistemos elgsena, kai prisijungęs dėstytojas ir jis vykdo įvairius jam leistinus veiksmus.

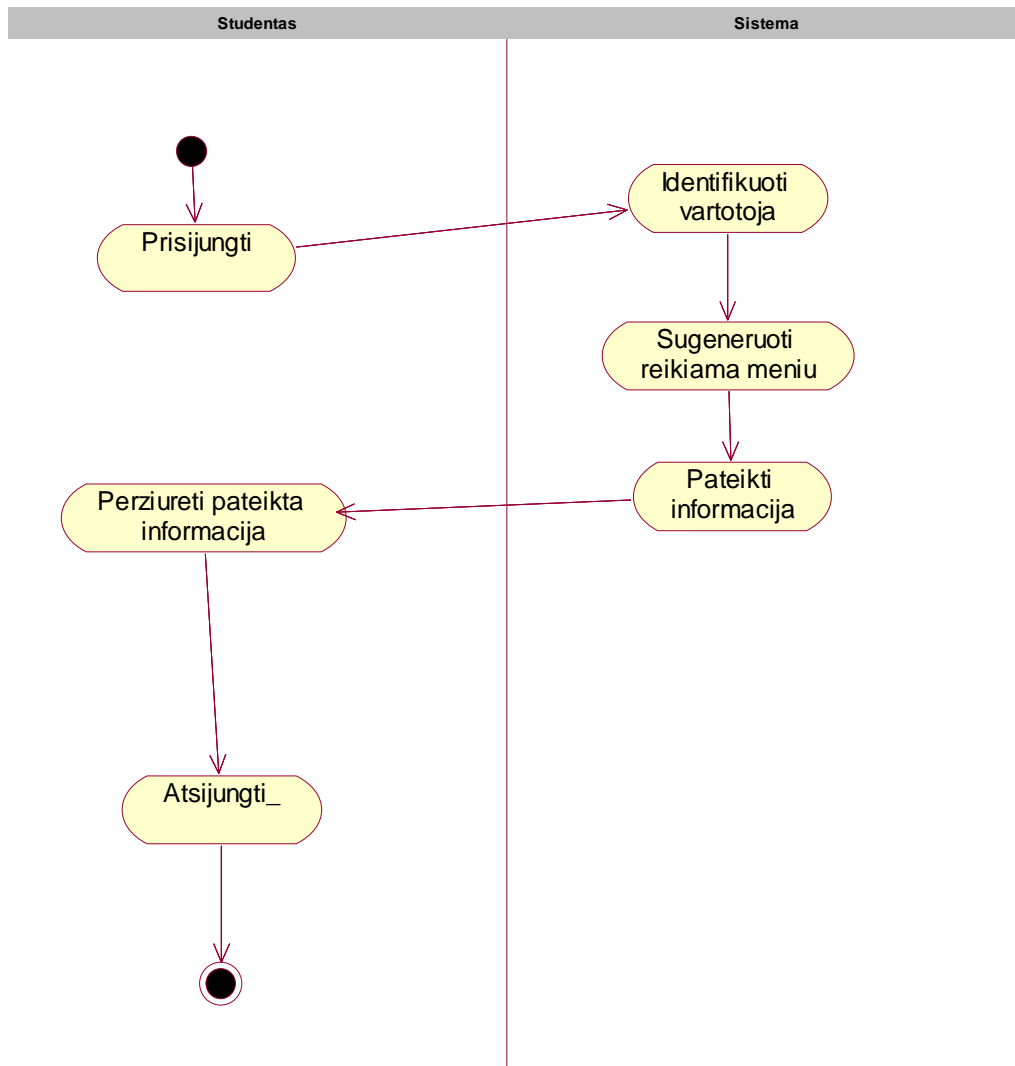


15 pav. Sistemos elgsena, prisijungus dėstytojui

Analizuojant pateiktą diagramą pastebime, kad atsiranda papildomos šakos tik tose vietose, kur keli veiksmi gali būti vykdomi vienu metu. Lygiagrečiai veiksmi sujungiami, todėl, kad sekantis veiksmas gali priimti tik vieno iš jų rezultatus

## 2.7 Sistemos elgsenos modelis, kai vartotojas studentas

Šiame modelyje pavaizduota (16 pav.) visos sistemos elgsena, kai prisijungęs studentas ir jis vykdo įvairius jam leistinus veiksmus.

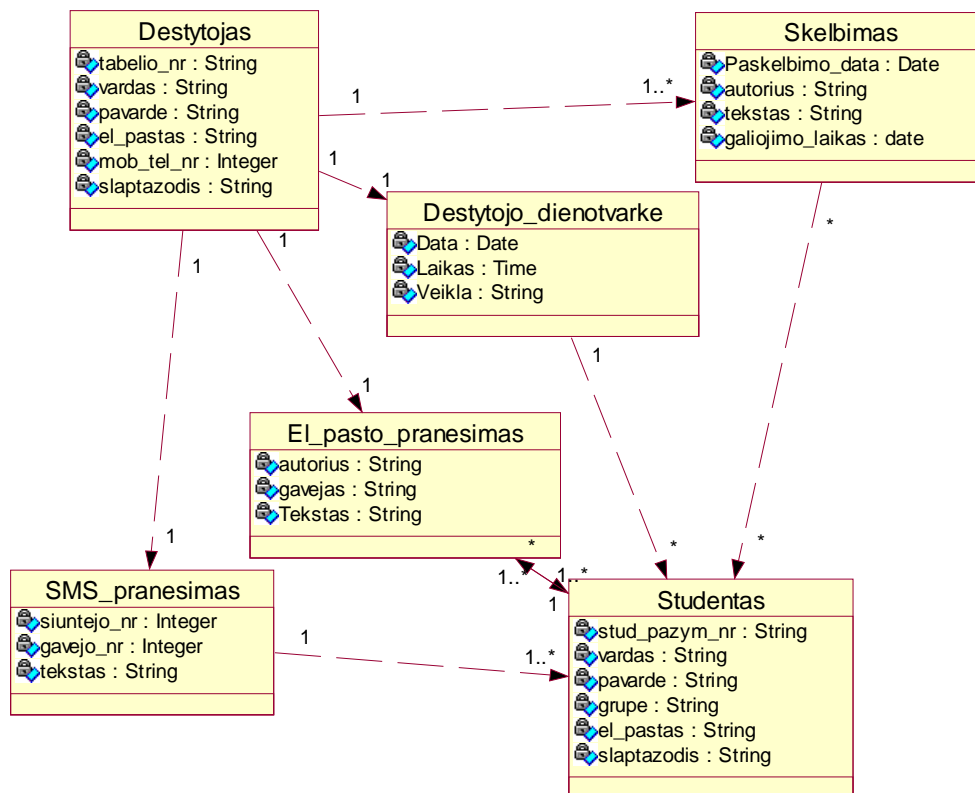


16 pav. Sistemos elgsena, prisijungus studentui

Studentas sistemoje gali atlikti mažiau veiksmų, nei dėstytojas, todėl modelis paprastesnis, nei dėstytojo atveju.

## 2.8 Dalykinės srities statinis modelis

Dalykinės srities klasių modelyje (17 pav.) sumodeliuotos sistemos ir vartotojų klasės, bei ryšiai tarp jų. Tai pravartu kadangi ši diagrama suteikia bendrąjį sistemos vaizdą. Tačiau klasių diagrama yra statiška, t.y. parodoma kas sąveikauja, bet neparodoma kokie yra tos sąveikos rezultatai.

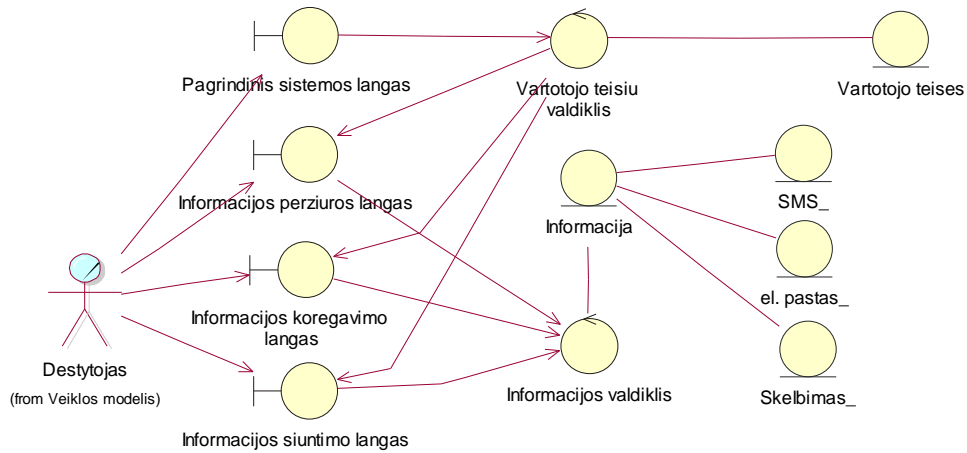


17 pav. Dalykinės srities klasių modelis

Daugelis ryšių yra „1 arba daug“, kadangi informaciją skelbia tik vienas žmogus, o skaito ją daugiau nei vienas.

## 2.8.1 Dėstytojo klasių modelis

Dėstytojo klasių modelyje (18 pav.) matyti, su kokiais objektais ir klasėmis susijęs sistemos vartotojas.

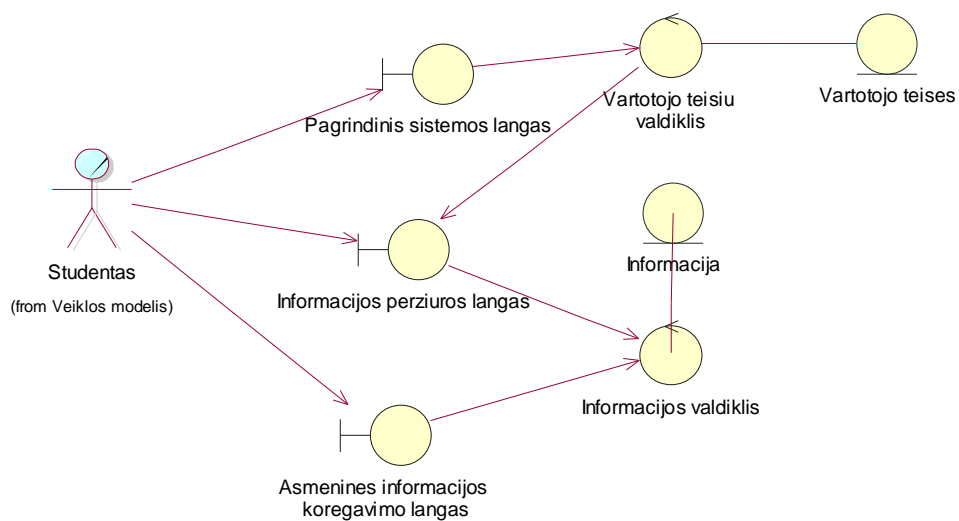


18 pav. Dėstytojo klasių modelis

Iš šio modelio matyti, koku būdu dėstytojas valdo informaciją ir kokio tipo.

## 2.8.2 Studento klasių modelis

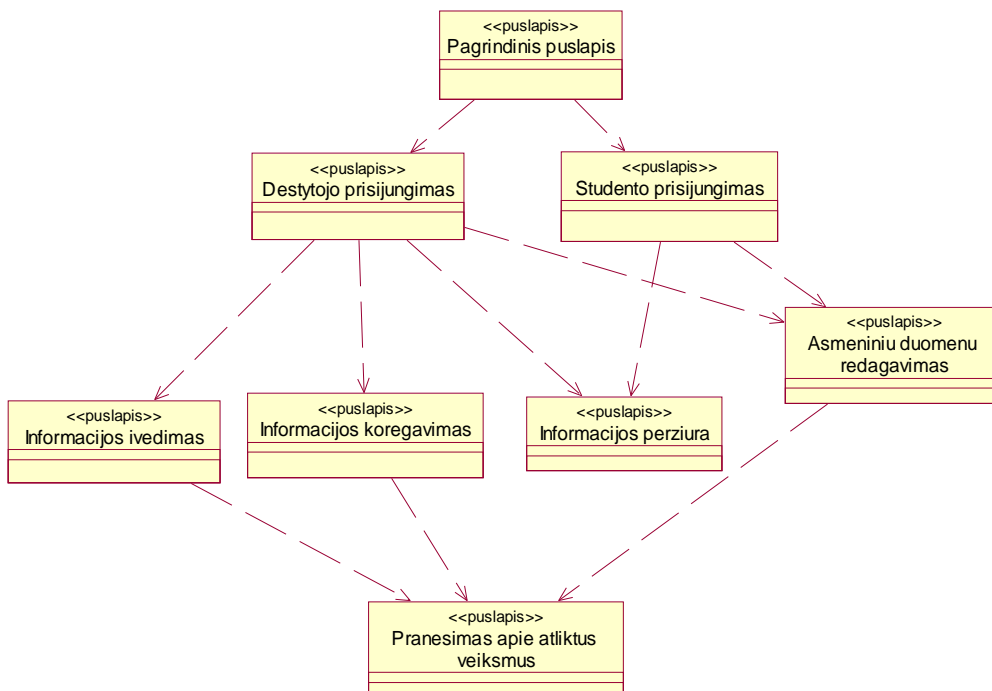
Analogiškai klasių modelis yra sudaromas ir studentui (19 pav.). Taip yra daroma todėl, kad studentai ir dėstytojai sistemoje turi skirtingas teises ir jie atlieka skirtingus veiksmus.



19 pav. Studento klasių modelis

## 2.9 Vartotojų interfeiso modelis

Šiame modelyje (20 pav.) pavaizduota vartotojo sąsaja su sistema. Visi vartotojo veiksmai atliekami per interneto naršyklę. Dėl šios priežasties visa sąsaja realizuota internetiniais puslapiais.



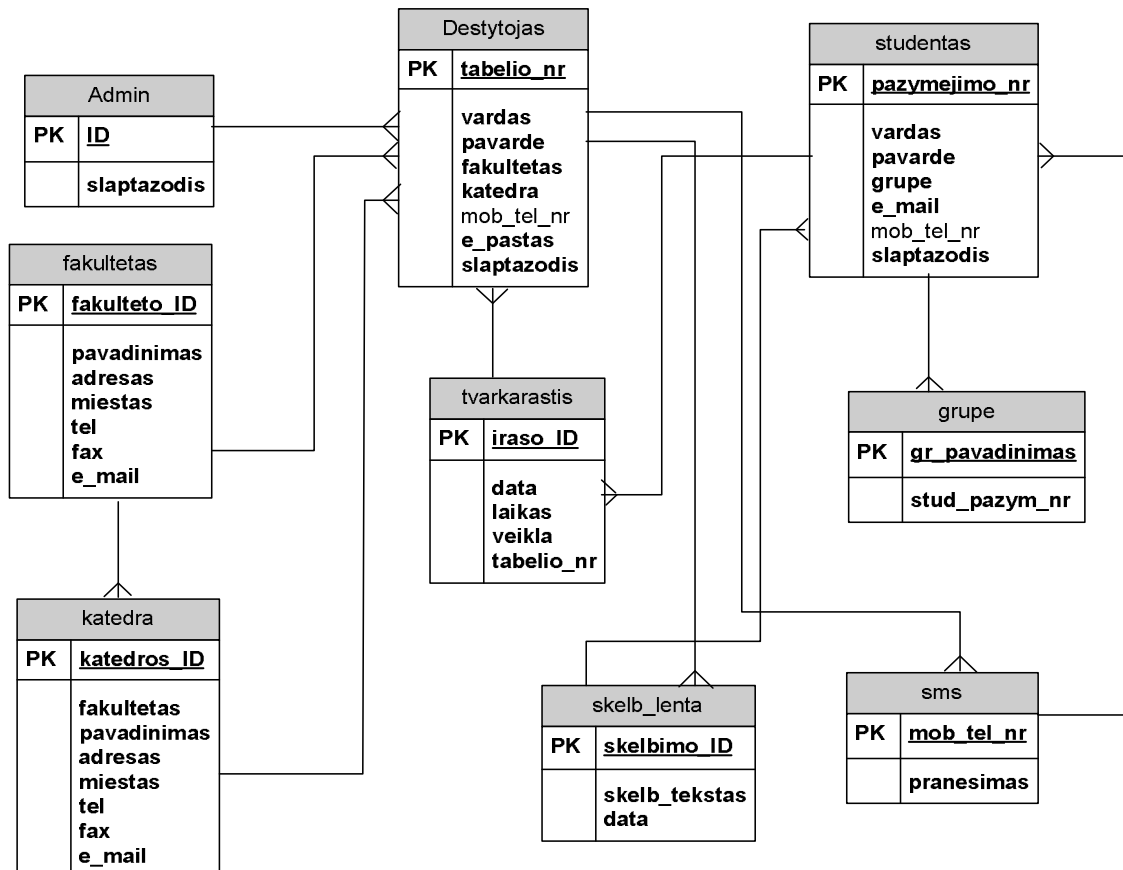
20 pav. Vartotojo interfeiso modelis

Visi vartotojo veiksmai valdomi keliais puslapiais. Taip yra išvengta sudėtingo sistemos valdymo bei kūrimo.



## 2.10 Duomenų bazės modelis

Visi „Mobilios informavimo sistemos“ duomenys saugomi MySQL duomenų bazėje. Tokiu būdu užtikrinamas duomenų klasifikavimas, išrinkimas, kai jų reikia. Visos duomenų bazės lentelės turi savo pirminius raktus, pagal kuriuos ieškoma konkreči informacija visoje lentelėje. Pilnas duomenų bazės modelis pateiktas 21 pav.



21 pav. Duomenų bazės modelis

Lentelėse visi laukai privalo turėti reikšmes, kitu atveju gaunami nepilni duomenys, dėl ko gali sutrikti normalus sistemos darbas.

## 2.11 Duomenų bazės lentelių aprašymas

Sekančioje lentelėje pateiktos visų duomenų bazės lentelių laukai bei jų reikšmių tipai.

6 lentelė

Lentelės pavadinimas	Lauko pavadinimas	Tipas
Admin	<b>ID</b>	Integer
	slaptažodis	Varchar (32)
Destytojas	<b>tabelio_nr</b>	Integer
	vardas	Varchar (32)
	pavarde	Varchar (32)
	fakultetas	Varchar (64)
	katedra	Varchar (64)
	mob_tel_nr	longinteger
	e-pastas	Varchar (32)
	slaptazodis	Varchar (32)
Fakultetas	<b>fakulteto_ID</b>	Integer
	pavadinimas	Varchar (64)
	adresas	text
	miestas	Varchar (32)
	tel	longinteger
	fax	Varchar (32)
	e_mail	Varchar (32)
Katedra	<b>katedros_ID</b>	Integer
	fakultetas	Varchar (64)
	pavadinimas	Varchar (64)
	adresas	text
	miestas	Varchar (32)
	tel	longinteger
	fax	Varchar (32)
	e_mail	Varchar (32)
Tvarkarastis	<b>iraso_ID</b>	Integer
	data	Date
	laikas	Time
	vieta	text
	tabelio_nr	integer
Studentas	<b>pazymejimo_nr</b>	Integer
	vardas	Varchar (32)
	pavarde	Varchar (32)
	grupe	Varchar (32)
	E_mail	Varchar (32)
	mob_tel_nr	Longinteger

	slaptazodis	Varchar (32)
<b>Grupe</b>	<b>gr_pavadinimas</b>	Varchar (32)
	stud_pazym_nr	Integer
<b>Skelb_lenta</b>	<b>skelbimo_ID</b>	Integer
	skelb_tekstas	Text
	data	Date
<b>SMS</b>	<b>mob_tel_nr</b>	Integer
	pranesimas	text

## 2.12 Sistemos panaudojimo atvejis

Prisijungus prie pagrindinio sistemos puslapio vartotojas matys pradinį sistemos puslapį (22 pav.).

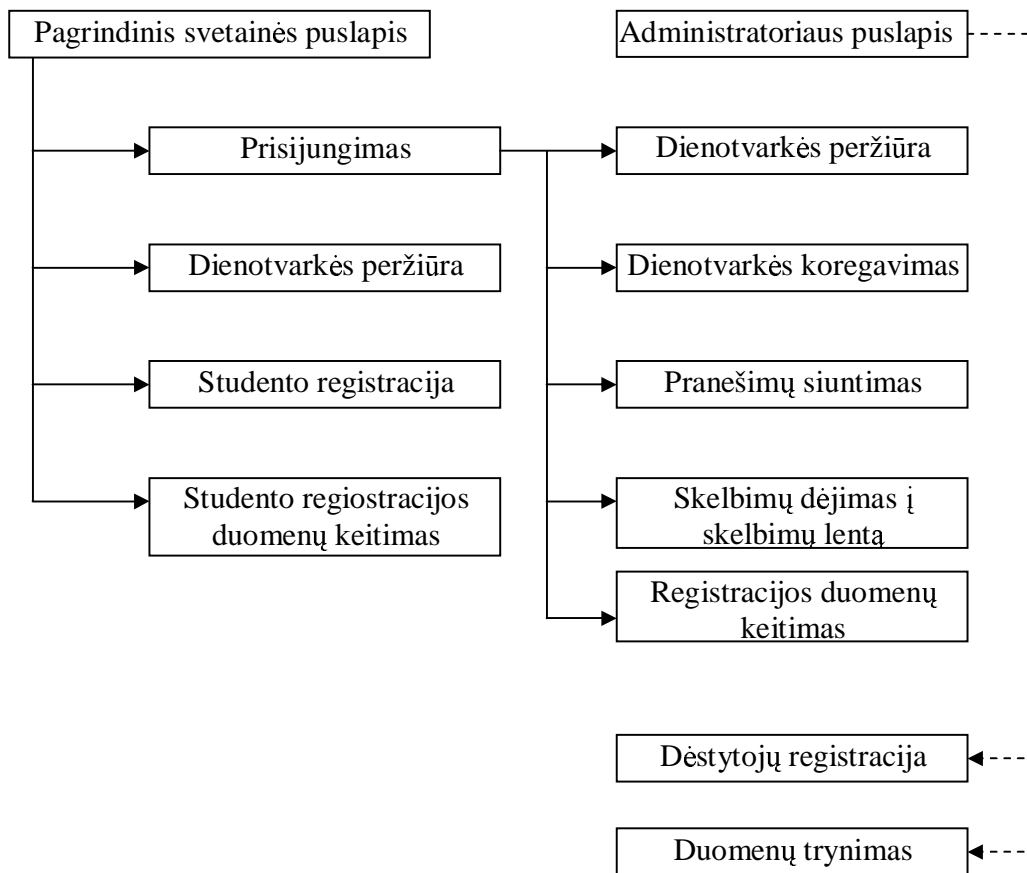


22 pav. Pagrindinis sistemos puslapis

Kairiojo meniu pagalba vartotojai atlieka savo veiksmus.

Vartotojas patekęs į pagrindinį sistemos puslapį gali patekti į studento arba dėstytojo registracijos puslapius, taip pat peržiūrėti galimas dienvarkes [13, 17].. Dėstytojas gali jungtis prie sistemos ir koreguoti, bei peržiūrėti savo ir kolegų dienvarkes, išsiųsti pranešimus suinteresuotiems asmenims.

Vartotojo sąsajos langų struktūra pateikta 23 paveiksle.



23 pav. Vartotojo sąsajos langų struktūra.

### 3. Išvados

- Šiame darbe apibrėžta „Mobili informavimo“ sistema, jai keliami reikalavimai.
- Detaliai išanalizuoti analizės metodai, jau egzistuojančios sistemos, sudaryti veiklos, klasių, būsenų ir kt. modeliai, leidžiantys suprasti sistemos vartotojų veiksmus, jų galimybes, sistemos reakciją į vartotojo veiksmus.
- Sudarytas sistemos duomenų bazės modelis bei detalus jo aprašas.
- Sukurtas realiai veikiantis sistemos pavyzdys, kuriame realizuotos visos aprašytos funkcijos bei galimybės.
- Realus „Mobilios informavimo sistemos“ pavyzdys leidžia kelis kartus paspartinti informacijos tvarkymą bei perdavimą suinteresuotiems asmenims.
- Sistema įgalina dėstytojus informaciją kitiems asmenims perduoti 2 būdais vienu metu.
- Reali sistema suprojektuota taip, ją būtų galima plėsti.

## 4. Literatūra

- [1] Barker R. CASE\*METHOD: Entity Relationship Modelling. New York: Addison – Wesley Publ. Co., 1990.
- [2] База PHP скриптов [interaktyvus]. 2003 m. [žiūrėta 2003-12-21]. Prieiga per Internetą: <http://phpclub.net/>
- [3] Butkienė R., Butleris R. The Approach for User Requirements Specification // 5<sup>th</sup> East-European conference ADBIS'2001, Research Communications, Ed. by A Čaplinskas, J.Eder, Vilnius, 2001, p. 225-240.
- [4] Chen P. The entity-relationship model: toward a unified view of data // ACM Transactions on Database systems, I(1), 1976, p. 9-36.
- [5] Client/Server Strategies. A survival guide for corporate reengineers. IDG Books Worldwide. (Vertimas į rusų kalbą, Kijevas, 1996, 297-321 p.).
- [6] Flex 7 Time Management [interaktyvus]. 2003 m. [žiūrėta 2003-12-04]. Prieiga per Internetą: <http://www.flex7.com/>
- [7] Gustas R. Semantic and Pragmatic Dependencies of Information Systems. Monograph. Kaunas: Technologija, 1997.
- [8] IDEF family of Methods [interaktyvus]. 2000m [žiūrėta 2004-05-13]. Prieiga per Internetą: <http://www.idef.com>
- [9] Loucopoulos P., Karakostas V. System requirements engineering. McGraw-Hill Book Company (UK) Limited, 1995.
- [10] MySql Database Server [interaktyvus]. 2003 m. [žiūrėta 2003-12-04]. Prieiga per Internetą: <http://www.mysql.com>
- [11] OMG Unified Modeling Language Specification [interaktyvus]. Object Management Group, Inc.1999. [žiūrėta 2003-05-14] Prieiga per Internetą: <http://www.omg.org/technology/documents/index.htm> .
- [12] OMT Object-Oriented Modeling and Design [interaktyvus] J.Rumbaugh, M.Blaho, W.Premarlani, F.Eddy, W.Lorensen Object-Oriented Modeling and Design, Englewood Cliffs. N.J., Prentice Hall, 1991.[žiūrėta 2004 -01-25] Prieiga per internetą: <http://control.ee.ethz.ch/edu/ciat2-SS98/OMT.pdf>

- [13] OPEN Object-oriented Process, Environment and Notation [interaktyvus] 2002m. [žiūrėta 2004-05-015]. Prieiga per internetą: <http://www.open.org.au/>
- [14] Project Management [interaktyvus]. 2003 m. kovas [žiūrėta 2003-05-04]. Prieiga per Internetą: <http://www.project-management-software.org/calendar/>
- [15] Proverbs Calendar [interaktyvus]. 2003 m. [žiūrėta 2003-12-04]. Prieiga per Internetą: <http://www.proverbs.biz/>
- [16] Rational Software Corporation, Rational Unified Process [interaktyvus], žinių bazė 2000m. [žiūrėta 2003-04-21] Prieiga per internetą: <http://www.rational.com/products/rup/index.jsp>
- [17] Snoeck M., Dedene G., Verhelst M., Depuydt A.M. Object-oriented Enterprise Modelling with MERODE. Belgium: Leuven University Press, 1999.
- [18] Weblicon PIM [interaktyvus]. 2003 m. [žiūrėta 2003-12-04]. Prieiga per Internetą: <http://www.weblicon.net/>

## 5. Terminų ir santrumpų žodynas

- API (*angl. Application Program Interface*) - įrankių, protokolų rinkinis aplikacijų kūrimui
- ASP (*angl. Active Server Pages*) - aktyvūs serverio puslapiai
- COM (*angl. Component Object Model*) - komponento objekto modelis
- CGI (*angl. Common Gateway Interface*) - bendra tinklų sąsaja
- GSM (*angl. Global System for Mobile Communications*) – globali mobilių komunikacijų sistema
- HTML (*angl. HyperText Markup Language*) - hipertekstinių žymių kalba
- HTTP (*angl. HyperText Transfer Protocol*) - hiperteksto siuntimo protokolas
- IIS (*angl. Internet Information Server*) - interneto informacijos serveris
- IMAP (*angl. Internet Message Access Protocol*) – protokolas elektroninių laiškų gavimui
- J2ME (*angl. Java 2 Platform Micro Edition*) – Java platforma bevieliams mobiliems įrenginiams
- JSP (*angl. Java Server Page*) – serverio puslapis realizuotas Java
- LDAP (*angl. Lightweight Directory Access Protocol*) - elementarus katalogų prieigų protokolas
- MMS (*angl. Multimedia Message Service*) – grafinių vaizdų, garsų, teksto siuntimo metodas bevieliais tinklais naudojant WAP protokolą
- PDA (*angl. Personal Digital Assistant*) – įrenginys, kuris turi kompiuterio, telefono/fakso, Interneto ir tinklo naudojimo bruožus
- PHP (*angl. Hypertext Preprocessor*) - hiperteksto pirminis redaktorius (programavimo kalba)
- PIM (*angl. Personal Information Manager*) – aplikacija padedanti vartotojui apdoroti įvairią informaciją
- RAM (*angl. Random Access Memory*) – operatyvioji kompiuterio atmintis
- RSA (*angl. Rivest, Shamir, Adelman*) – kūrėjų vardais pavadintas šifravimo algoritmas
- SMS (*angl. Short Message Service*) - trumpųjų žinučių paslauga
- SQL (*angl. Structured Query Language*) - struktūrizuotų užklausų kalba



TCP/IP (*angl. Transmission Control Protocol/Internet Protocol*) – protokolų rinkinys naudojamas susijungimams ineternete tarp kompiuterių

UML (*angl. Unified Modeling Language*) – projektavimo kalba

UTP (*angl. Unshielded Twisted Pair*) – kabelio tipas iš dviejų be šarvo laidų

WAP (*angl. Wireless Application [Protocol](#)*) – protokolas naudojamas keistis informacija tarp bevielių įrenginių

XML (*angl. Extensible Markup Language*) - išplečiama žymių kalba

## Priedai

### DUOMENŲ INTEGRAVIMAS

**Dainius Jurčikonis, Aurimas Paršonis, Egidijus Kazanavičius**

*Kauno Technologijos Universitetas, Kompiuterių katedra*

*Studentų g. 50-213, LT-3031 Kaunas*

Straipsnyje bus nagrinėjama tam tikri standartai ir technologijos, kuriais remiantis vyksta duomenų perdavimas tarp skirtingų duomenų šaltinių. Apžvelgiama esama situacija duomenų integracijos srityje, siūlomi sprendimai, esamos technologijos. Taip pat bus palyginamos tarpininkavimo programos ir jų tipai.

### Įvadas

Dauguma kuriamų sistemų, kurios operuoja dideliais duomenų kiekiais, naudoja reliacines duomenų bazes. Jos gali veikti skirtingose operacinėse sistemose, naudoti skirtingą programinę įrangą. Taip pat naudojamos skirtingos duomenų bazės (MySQL, Oracle, MS SQL ir kt.), kurių duomenų struktūra skiriasi viena nuo kitos. Atsiranda poreikis plėtojimuisi ir iškyla būtinybė keistis duomenimis. Dėl saugomų duomenų struktūrų nesuderinamumo susiformuoja informacijos apsikeitimo problema, apsikeitimas informacija tampa sudėtingas uždavinys.

Heterogeninėse duomenų bazėse duomenų integracija įmanoma tik naudojant bendrą integruojamų duomenų bazių struktūras apjungiantį standartą. Standartas turi būti pritaikomas skirtingoms duomenų bazėms neatsižvelgiant į jų struktūros specifiką ir pasižymėti universalumu.

### Duomenų perdavimo standartai

Norint keistis duomenimis tarp heterogeninių duomenų bazių reikia sukurti apibendrintą struktūrą, kuri apimtų visų sistemų duomenis, kurie gali būti tarpusavyje visiškai nesuderinami. Tokiu atveju kiekvienai sistemai reiktų kurti programinius įrankius arba paketus, kurie palaikytų prisijungimą prie kitų sistemos duomenų bazių. Tai yra tikrai nepatogu ir sudėtinga. Dėl to duomenų tarp nehomogeninių duomenų bazių perdavimui patogiausia būtų naudoti tokį duomenų tipą, kuris labiausiai atitiktų sistemos duomenų bazių duomenis ir tiesiogiai nepriklausytų nuo atitinkamos sistemos ar duomenų bazės. Šiuo atveju patogiausia naudoti XML (EXtensible Markup Language) [7].

### XML technologija

- XML skirta keitimuisi duomenimis;
- XML skirta keistis informacija;
- XML gali būti naudojama duomenų paskirstymui;

- XML gali būti skirta duomenų saugojimui;
- XML gali padaryti duomenis labiau prieinamus.

Remiantis aukščiau išvardintais XML požymiais galima daryti išvadą, jog XML kalba tinka duomenų keitimuisi tarp heterogeninių duomenų bazių. Tačiau šis procesas nėra toks paprastas, nes reikia atsižvelgti į įvairių sistemų funkcinis reikalavimus. Sukūrus bendrą šabloną, būtų įmanoma atlikti keitimąsi duomenimis tarp įvairių sistemų. Naudojant XML šabloną duomenų keitimui patogiau naudoti XML schemas.

### **XML schemų paskirtis**

XML schema nusako XML dokumento struktūrą. XML schemas skirtos apibrėžti [8]:

- dokumente naudojamus elementus
- dokumente naudojamus atributus
- kurie elementai yra elementai – vaikai
- elementų – vaikų tvarką
- elementų – vaikų kiekį
- ar elementas yra tuščias ar gali saugoti tekstinę informaciją
- elementų duomenų tipus ir atributus
- standartines ir fiksuotas elementų ir atributų reikšmes
- ryšius tarp elementų

XML schemas paskirties aprašymo struktūra ir sandara yra artima reliacinių duomenų bazių struktūrai. Todėl norint keistis duomenimis reikia sukurti XML schema, kuri leistų perduoti duomenis į bet kurią sistemą. Tačiau dėl XML schemų ir reliacinių bazių struktūros neatitikimo iškyla konvertavimo problemos.

Pagal taisykles suformuotas XML dokumentas tai – dokumentas, atitinkantis XML sintaksę ir dokumento tipo aprašymą DTD (Document Type Definition). XML dokumento struktūra yra aprašoma DTD arba XML schema. DTD - tai rinkinys taisyklių, kurios apibrėžia leidžiamą dokumento klasių struktūrą [1].

XML dokumentas yra lengvai pernešamas tarp sistemų ir platformų, nes jo turinys yra paprastas tekstas (plain text).

W3C pasiūlytas standartas yra XSL (eXtensible Stylesheet Language), apibrėžiantis, kaip XML dokumentas bus vaizduojamas naršyklėje [4, 5]. XSL susideda iš trijų dalių: XSLT, XPath ir XSL Formatting Objects. XSLT (eXtensible Stylesheet Language Transformations) tai – kalba, skirta transformuoti XML dokumentą į kitokio formato dokumentą. XSLT gali būti naudojama XML dokumentą transformuoti į kitą XML dokumentą. XPath tai – kalba skirta identifikuoti XML dokumento dalis. XSL Formatting Objects tai – kalba, kuri formuoja XML dokumento vaizdą po XSLT transformacijos. XSLT gali transformuoti XML dokumentą į dokumentą, kuris gali būti atvaizduotas naršyklėje. XSLT taip pat

gali koreguoti informaciją transformuotame dokumente. XSLT elementų, atitinkančių tam tikras sąlygas ir po to atliekančių elementų transformaciją, suradimui XML pradiniam dokumente naudoja XPath.

XSL naudoja vieną ar kelis šablonus, kurie apibrėžia, kaip atvaizduoti XML elementus. Standartas leidžia naudoti sąlygos ir pasirinkimo operacijas, taip pat gali perrūšiuoti pradinį XML dokumentą. Panaudojus XSL, galima pakeisti XML dokumento struktūrizacijos lygmenį, įtraukiant, ištrinant, pergrupuojant elementus. XSL privalumas yra tai, kad galima surasti konkretų elementą, pasinaudojus šablonais, kurie tikrina elementų turinį, ir po to su juo atlikti norimus veiksmus.

### Duomenų perdavimo technologijos

Tarpininkavimo programos (TP) – tai programinė įranga, tarpininkaujanti tarp skirtingų sistemų [6]. TP sukurta perdavimo sistemoms, komunikacijai ir duomenų sandėliui. Svarbiausia TP funkcija duomenų sandėliavime - sukurti pastovų kelią, pasiekiant duomenų bazę.

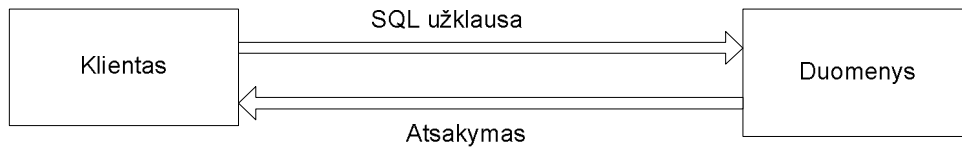
Tarpininkavimo programos - bet kokio tipo programinė įranga, kuri palengvina komunikaciją tarp dviejų ar daugiau programų sistemų. TP turi mechanizmą, kuris vienai esybei (taikomajai programai ar duomenų bazei) leidžia komunikuoti su kita esybe ar esybėmis, leidžia paslėpti pirminio šaltinio ir adresato sistemų sudėtingumus.

Taikomųjų programų tipai:

- Nutolusių procedūrų iškvietimas (RPC - Remote Procedure Calls);
- Į pranešimus orientuotos TP (MOM - Message Oriented MW);
- Paskirstytų objektų (Distributed Objects);
- Į DB orientuotos TP (DB oriented MW);
- Transakcinės TP įtraukiant transakcijų apdorojimo monitorius ir taikomųjų programų darbo stotis (Transactional MW);
- Pranešimų brokeriai (Message brokers).

Į DB orientuotos taikomosios programos

Šio tipo programos supaprastina komunikaciją su duomenų baze. Projektuotojai naudoja į duomenų bazes orientuotas taikomasias programas, kaip mechanizmą informacijos išgavimui iš lokalinės ar nutolusios duomenų bazės.



1 pav. Į DB orientuota taikomoji programa.

### XML produktai ir jų kategorijos

XML dokumentai skirstomi į dvi kategorijas: orientuoti į duomenis, bei orientuoti į dokumentus [5].

- **Tarpininkavimo programos (Middleware):** programinė įranga, skirta duomenų perdavimui tarp XML dokumentų bei duomenų bazės.
- **Duomenų bazės įgalinančios XML (XML-Enabled Databases):** duomenų bazės skirtos duomenų perdavimui tarp XML failų.
- **XML duomenų bazės (Native XML Databases):** XML duomenų bazės.
- **XML serveriai (XML Servers):** WEB serveriai, integravimo varikliai bei specifiniai serveriai palaikantys XML publikavimą, generavimą ir t.t.
- **Aplankalai (Wrappers):** programinė įranga, traktuojanti XML dokumentus kaip reliacinių duomenų šaltinį. Dažniausiai šie produktai generuoja XML dokumentus SQL sintaksės pagalba.
- **Turinio valdymo sistemos (Content Management Systems):** programinė įranga, skirta turinio valdymui (tikrinimas, verifikavimas, redagavimas, versijų kontrolė ir pan.).
- **XML užklausų varikliai (XML Query Engines):** specifinė programinė įranga XML duomenų užklausų vykdymui.
- **XML duomenų įrišimas (XML Data Binding):** įranga, susiejanti XML dokumentus su objektais.
- **Nutraukti produktai (Discontinued products):** produktai, kurių vystymas ir tolimesnis palaikymas jau nutraukti.

Išskyrus turinio valdymo sistemas, visais kitais atvejais, reikia patiems parašyti atitinkamą kodą, norint integruoti produktą į savo sistemą.

Dažniausiai naudojamos reliacinės DBVS naudojančios ODBC, JDBC arba OLE DB. Tačiau yra ir kitokių produktų, skirtų duomenų mainams tarp kitokio tipo DB. Žemiau pateikta produktų palyginimo lentelė.

2 lentelė. TP produktų palyginimas

Produktas	Gamintojas	Licenzija	DB tipas	DB >XML	XML >DB
eTools XML	GA eXpress	Komercinis	Daugiareikšmis (MultiValue)	x	x
Extreme Translator	Etasoft	Komercinis	Reliacinis	x	x
ADO	Microsoft	Komercinis	Reliacinis	x	x
e.Report	Actuate	Komercinis	Reliacinis	x	--

Easysoft XML-ODBC Server	Easysoft	Komercinis	Reliacinis	x	--
DBIx::XMLServer	Martin Bright	Atviro kodo	Reliacinis	x	--
DBIx::XML_RDB	Matt Sergeant	Atviro kodo	Reliacinis	x	x
DbToXml	SoftRUs	Komercinis	Reliacinis	x	x
DB2XML	Volker Turau	Atviro kodo	Reliacinis	x	--
Connect XML-2-DB	Skyhawk Systems	Komercinis	Reliacinis	--	x
Connect for SQL/XML	DataDirect Technologies	Komercinis	Reliacinis	x	x
Charteris Integration Toolkit	Charteris	Komercinis	Reliacinis	x	x
Castor	exolab.org	Atviro kodo	Reliacinis	x	x
XML Lightweight Extractor (XLE)	IBM	Analizei	Reliacinis	x	--
XML Spy	Altova	Komercinis	Reliacinis	x	x
XML Junction	Data Junction, Inc.	Komercinis	Reliacinis, ISAM ir kt.	x	x
XML::Generator::DBI	Matt Sergeant	Atviro kodo	Reliacinis	x	--
XML Gateway	SPI Ltd.	Komercinis	Reliacinis, Word, teksto failai	x	x
XML for Tables	IBM	Tik analizei (Evaluation only)	Reliacinis (DB2)	x	--
XML-DBMS	Ronald Bourret, et al	Atviro kodo	Reliacinis	x	x
XML DataDesk	NetBryx Technologies	Komercinis	Reliacinis	x	--
xlinkit	xlinkit.com	Nemokamas/ Komercinis	Reliacinis	x	--
X:Forge	Bibop Research International	Atviro kodo	Reliacinis, XML (Native XML)	x	--
SXQL	Goetz Hatop	Viešoji programa (Shareware)	Reliacinis	x	x
ASP2XML	Stonebroom	Komercinis	Reliacinis	x	x
mysql, mysqldump	MySQL	Atviro kodo	Reliacinis	x	--

### Connect for SQL/XML

Connect for SQL/XML yra JDBC tvarkyklė, kuri realizuoja SQL/XML specifikaciją lygiai taip pat kaip JDBC. Užklausų rezultatai gražinami JDBS „ResultSet“ pavidalu. XML reikšmės yra gražinamos XML tipo stulpeliuose. Programa pasiima tas reikšmes getObject metodo pagalba.

### Connect XML-2-DB

Connect XML-2-DBL yra JAVA programinė įranga skirta perduoti duomenis tarp XML dokumentų bei SQL serverio arba Oracle. Naudoja objektinį-reliacinį duomenų paėmimo būdą.

Connect XML-2-DB gali būti paleista tiek iš kitos JAVA aplikacijos, tiek ir iš komandinės eilutės.

### ASP2XML

COM objektas XML dokumentų perdavimui ODBC arba OLE DB duomenų šaltiniams ir atvirkščiai. Produktas supranta XML dokumentą kaip vientisą lentelę. Panaudojus SELECT sakinį, gražinamas dokumentas su specifiniais elementais bei reikiama informacija. Norint atgalinio ryšio, būtina, kad XML failas turėtų specifinius ASP2XML programos elementus (tags). Šis objektas gali veikti MS ASP puslapiuose, o taip pat kaip atskira aplikacija.

### **Mysql, mysqldump**

„Mysql“ yra komandinėje eilutėje valdomas įrankis, kurio pagalba vykdomos MySQL DBVS užklausos duomenų užkrovimui. „Mysqldump“ – atitinkama aplikacija, tačiau naudojama ne su MySQL, bet su MS SQL serveriu.

### **Išvados**

Duomenų integracijai dažniausiai naudojamas XML standartas. Jis naudojamas kuriant struktūrinius dokumentus arba aprašant duomenis. Dokumento struktūrai apibrėžti naudojama DTD arba XML schema. XML gali būti transformuotas į kitus formatus naudojant XSL standartą. Tam, kad perduoti duomenis tarp skirtingų tipų duomenų bazių bei XML dokumentų, naudojamos tarpininkavimo programos. Duomenų integracija - ilgas ir didelių darbo bei finansinių resursų reikalaujantis procesas.

### **Literatūros sąrašas**

[1] W3C XML working group, “Extensible Markup Language (XML) 1.0 (Second Edition)”, [žiūrėta 2004-01-03]. Prieiga per internetą: <http://www.w3.org/TR/2000/REC-xml-20001006>;

[2] David C. Fallside (W3C), “XML Schema Part 0: Prime”, [žiūrėta 2004-01-03]. Prieiga per internetą: <http://www.w3.org/TR/xmlschema-0/>;

[3] Sharon Adler, Anders Berglund, Jeff Caruso, Stephen Deach, Paul Grosso, Eduardo Gutentag, Alex Milowski, Scott Parnell, Jeremy Richman, Steve Zilles, “Extensible Stylesheet Language (XSL) Version 1.0”, [žiūrėta 2004-01-03]. Prieiga per internetą: <http://www.w3.org/TR/xsl/>;

[4] James Clark, “XSL Transformations (XSLT) Version 1.0”, [žiūrėta 2004-01-03]. Prieiga per internetą: <http://www.w3.org/TR/xslt.html>;

[5] Ronald Bourret, XML Database Products [žiūrėta 2004-01-03]. Prieiga per internetą: <http://www.rpbourret.com/xml/XMLDatabaseProds.htm>

[6] David S. Linthicum, Application Integration Exposed [žiūrėta 2005-01-03]. Prieiga per internetą: <http://www.softwagemag.com/L.cfm?Doc=archive/2000feb/EAI.html>

[7] T. Konovalovas, B. Paradauskas. Nehomogeninių duomenų bazių integracija naudojant XML formatus // Informacinės technologijos ir mokslų integracija – 2003 : aštuntoji magistrantų ir doktorantų konferencija : konferencijos pranešimų medžiaga. Kaunas : Technologija, 2003, p. 45 – 48.

[8] „Introduction to XML Schema“, interneto prieiga: [http://www.w3schools.com/schema/schema\\_intro.asp](http://www.w3schools.com/schema/schema_intro.asp)

## **Data integration**

### **Annotation**

In this article analyzed some standards and technologies according to data transfer processes between different data sources. Existing situation is reviewed in the data integration sphere, market offers, existing technologies. There will be compared middleware and their types.