

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Dainius Jurčikonis

**Studijų modulių planavimo
ir valdymo sistema**

Magistro darbas

Darbo vadovas

prof. E. Kazanavičius

Kaunas
2005

TURINYS

Santrumpų ir terminų žodynas	4
Įvadas	5
1. Informacinės sistemos analizės metodai ir priemonės	6
2. Informacinės sistemos analizės rezultatai.....	7
2.1. Esamos situacijos analizė.....	7
2.1.1. Analizės tikslas.....	8
2.1.2. Veiklos sąveikų modelis	8
2.1.3. Veiklos tikslų modelis	8
2.1.4. Veiklos panaudojimo atvejų modelis.....	9
2.1.4.1. Katedros vedėjo veiklos panaudojimo atvejo modelis	10
2.1.4.2. Dėstytojo veiklos panaudojimo modelis	10
2.1.4.3. Studento veiklos panaudojimo modelis.....	11
2.1.5. Veiklos objektų modelis.....	11
2.1.6. Organizacinės struktūros modelis.....	12
2.1.7. Veiklos procesų modelis	13
2.2. Literatūros šaltiniuose pateiktų sprendimų problemai spręsti lyginamoji analizė.....	13
3. Darbo tikslas ir jo pagrindimas, kokybės kriterijų apibrėžimas	29
4. Projektavimo metodų, priemonių parinkimas	30
5. Kompiuterizuojamos sistemos varianto parinkimas.....	32
6. Darbo rezultatai	33
6.1. Kompiuterizuojamų veiklos procesų modelis	33
6.2. Reikalavimų modelis	34
6.2.1. Vartotojų veiklos procesų modelis	34
6.2.2. Specifikacijos kiekvienam panaudojimo atvejui.....	34
6.2.3. Dalykinės srities statinis modelis.....	38
6.2.4. Vartotojo sąsajos modelis	41
6.2.5. Papildomi reikalavimai sistemai.....	42
7. Sistemos struktūra ir realizacija	44
7.1. Duomenų integracijos panaudojimų atvejų ir sekų (arba bendradarbiavimo) diagramos.....	44
7.1.1. XML srauto sudarymo iš KTU akademinės DB veiklos procesų sekų modelis	44
7.1.2. XML srauto sudarymo iš KTU akademinės DB bendradarbiavimo modelis.	45
7.1.3. XML srauto sudarymo iš KTU kompiuterių katedros DB veiklos procesų sekų modelis.....	46
7.1.4. Duomenų atnaujinimo KTU akademinės DB veiklos procesų sekų diagrama	47
7.2. Sistemos elgsenos modelis	48
7.2.1. Sistemos elgsenos modelis , kai atnaujinama viena iš duomenų bazių	48
7.2.2. Sistemos elgsenos modelis , kai atliekama pilna sinchronizacija	49
7.3. Duomenų bazių modelis.....	50
8. Realizacijos modelis	58
8.1. Komponentų diagrama.....	58
8.2. Paskirstymo diagrama.....	58
8.3. Testavimo modelis.....	58
8.4. Reikalavimai sistemos funkcionavimo palaikymui.....	59
8.5. Sistemos panaudojimo atvejis	59
Darbo išvados	60
Naudota literatūra	61
Priedai	63

Planning and management system of studies modules

Summary

In this work analyzed some standards and technologies according to data transfer processes between different data sources. New database development and usage requires legacy database data integration into new databases. The common integration structure must be created for this purpose. During integration we should pay attention to name conflicts. XML is being used because of its usage flexibility and simplicity. Data exchange using XML enables to ensure data storage for the future regenerative possibility. System realization was used in KTU Computer Cathedral activity.

Santrumpų ir terminų žodynas

XML – išplėstinė žymių kalba

RDB – reliacinė duomenų bazė

DB – duomenų bazė

DBVS - duomenų bazių valdymo sistema

IS – informacinė sistema

DTD – dokumento tipų aprašymas

SQL – struktūrizuota užklausų kalba

HTML – hiperteksto žymių kalba

TP – tarpininkavimo programa

Įvadas

Dauguma kuriamų sistemų, kurios operuoja dideliais duomenų kiekiais, naudoja reliacinės 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.

Darbo tikslas - išnagrinėti pasirinkto objekto ypatumus, panaudojant CASE metodologijas, aprašyti pasirinktą taikomąją sritį, kurioje yra integruojamos duomenų bazės. Šiame darbe taip pat numatyta išnagrinėti tarpininkavimo programų panaudojimas duomenų integracijai ir XML programavimo kalbos specifiška. Sukūrus transformacijos metodus, reikia sugeneruoti XML srautą ir jį panaudoti heterogeninių duomenų bazių duomenų integracijai.

Heterogeninių duomenų bazių integracija panaudojant XML srautus buvo pritaikyta Kompiuterių katedros veikloje. Tokio tipo integracija yra naudinga, nes ją pritaikant katedros veikoje pavyksta išvengti iškilusių problemų, kurios atsiranda katedros dėstytojams dirbant su Kompiuterių katedros studijų modulių sistema, kuomet reikia ta pačią informaciją suvedinėt ir į KTU akademinę duomenų bazę. XML srautais paremtai integracijai nereikia žinoti partnerių duomenų bazės struktūros. Tereikia aprašyti bendrą duomenų, kurie yra naudojami keitimosi procese, struktūrą.

Darbo pabaigoje pateikiamos šio darbo išvados, taip pat nurodoma naudota literatūra, santrumpų ir terminų žodynas. Priedų skyriuje yra pateikiamas busimai konferencijai "Informacinės technologijos'2005" pristatomas straipsnis „Duomenų integravimas“ ir realizuoto eksperimento XML dokumento aprašas.

Tyrimo sritis apima duomenų integraciją tarp KTU studijų modulių IS duomenų bazės ir KTU kompiuterių katedros modulių IS duomenų bazės.

Srities objektas – KTU kompiuterių katedros studijų modulių sistemos duomenų bazė, kuri realizuota MySQL aplinkoje ir KTU studijų modulių sistemos duomenų bazė, realizuota ORACLE aplinkoje. KTU studijų modulių sistemos duomenų bazės struktūra ir

jos duomenys bus imituojami, pagal realiai veikiančią, kad būtų galima atlikti duomenų mainus. Duomenų srautų formavimas ir transportavimas bus atliekamas panaudojus XML (EXtensible Markup Language) technologiją.

Problema, darbo aktualumas. Darbo tikslas yra išspręsti problemas, kurios atsiranda, bet kuriuo metu kuriant naujus modulius ar atnaujinant senus, atliekant veiksmus su studentų sąrašais mokymo proceso eigoje. Tam, kad nereiktų papildyti ta pačia informacija abiejų duomenų bazių, dėl šių priežasčių ir planuojama sukurti studijų modulių planavimo ir valdymo sistemą, kuri integruotų minėtas abi sistemas į vieningą visumą vartotojo atžvilgiu. Šios sistemos pagrindinis tikslas taisyklingai suformuoti XML duomenų srautus ir atlikti apskaitimą tarp dviejų duomenų bazių, kurie nepriklausytų nuo abiejų IS funkcijų ir architektūros.

1. Informacinės sistemos analizės metodai ir priemonės

Tiriamąjį darbo analizę galima atlikti taikant UML priemones arba panaudojus struktūrinį projektavimą. Darbe bus naudojamas analizės metodas taikant UML priemones, nes ši metodologija, skirtingai nuo struktūrinio projektavimo leidžia pilnai aprašyti kuriamos sistemos problemine sritį, reikalavimus bei realizacijos procesą. Taikant UML analizės metodiką, rinktasi iš dviejų UML notacijų ir semantiką palaikančių CASE priemonių (1 lentelė).

1 lentelė. Rational Rose ir MagicDraw paketų palyginimas

Pasirinkimo kriterijai	Rational Rose 2000	MagicDraw
Pilnas UML (1.4 versijos) palaikymas	+	+
Diagramų pasirinkimo sąrašai	+	+
Diagramų suderinamumo kontrolė	+	+
Modelio navigacija	+	+
Spausdinimas	+	+
Dokumentavimas HTML	+	+
Diagramų eksportavimas	+	+
Versijų palaikymas	+	+
Kodo generavimas	+	+
Platformų palaikymas	C++, VisualC,	Java, C++, .NET

	Java, VBasic, ...	
Duomenų modeliavimas	+	+
Tiesioginis ir atvirkštinis projektavimas	+	+
Operacinių sistemų palaikymas	Windows	Windows, Linux/Unix
Modulių neprikaištingumo tikrinimas	+	+

Atlikus MagicDraw ir Rational Rose paketų palyginimą, bei atsižvelgus į kiekvieno paketo galimybes, nuspręsta pasirinkti objektinio modeliavimo CASE priemonę "Rational Rose 2000 Enterprise Edition". Su šia modeliavimo priemone buvo susipažinta vieno iš studijų modulio dėstytojų metu, tai palengvino jos įsisavinimą ir galimybių pažinimą bei įtakoją šios CASE priemonės pasirinkimą. Šio paketo pagalba sukurtos diagramos, leidžiančios pilnai aprašyti kuriamos sistemos probleminę sritį ir reikalavimus, pateiktos šiame darbe.

2. Informacinės sistemos analizės rezultatai

2.1. Esamos situacijos analizė

Kompiuterių katedra yra Kauno Technologijos Universiteto Informatikos fakulteto padalinys. Tiek kompiuterių katedroje, tiek ir kitose pastaruosiu metu sparčiai vystantis informacijos apdorojimo technologijoms išskyla pačios informacijos pateikimo problema, t.y. kaip paruošti ir pateikti informaciją, kad vėliau tiek pačiam informacijos tiekėjui, tiek informacijos gavėjui būtų lengva ja naudotis, būtų galima lengvai paskleisti turimą informaciją. Tai yra svarbus yra informacijos organizavimo uždavinys. Toks informacijos tvarkymas reikalauja tam tikro supratimo ir mokėjimo naudotis naujausiomis technologijomis. Tai galima nesunkiai išspręsti, bet išskyla rimtesnė problema - laiko sąnaudų klausimas. Didelę laiko dalį užima informacijos surinkimas ir jos paruošimas tam tikroms reikmėms. Todėl jos pateikimas turi būti kiek galima paprastesnis.

Dėstytojo informacijos pateikimas yra problematiškas. Norėdamas padėti informaciją studentų naudojimui jis turi pats spręsti su tuo susijusius uždavinius. Tai ir vietos suradimas, ir priėjimo kontroliavimas, ir kita. Studentams taip pat kartais sunku surasti reikiama informaciją, nes keičiasi jos vieta, pati informacija kartais būna pateikta netvarkinga ir nelogiška tvarka.

Šiai problemai išspręsti buvo sukurta Kompiuterių katedros portalo studijų modulių sistema, kuri palengvina informacijos organizavimą tarp dėstytojo ir studento. Tačiau

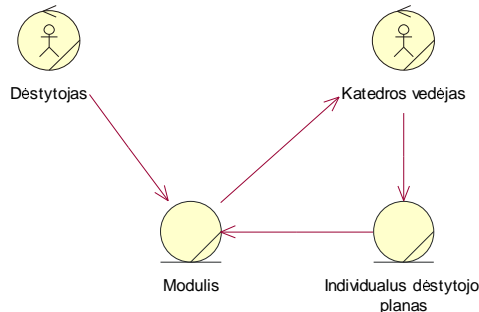
nebuvo atsižvelgta į tai, kad dėstytojam tenka įvedinėti į akademinę duomenų bazę tokius pat duomenis, kaip ir į kompiuterių katedrai sukurtos sistemos duomenų bazę.

2.1.1. Analizės tikslas

Analizės tikslas yra atlikti katedroje vykstančių procesų analizę, išsiaiškinti ir išanalizuoti šių procesų realizavimo būdus ir priemones, kurių kompiuterizavimas leistų katedros vedėjui ir dėstytojams efektyviau organizuoti savo veiklą (atlikti naujų modulių kūrimo ar senų atnaujinimo, studentų sąrašų sudarymo procesus).

2.1.2. Veiklos sąveikų modelis

Pagrindiniai organizacijos veiklos dalyviai yra katedros vedėjas ir dėstytojas. Katedros dėstytojai kuria savo modulius ir gali atlikti jų redagavimą ar pašalinimą, tuo tarpu katedros vedėjas (turi sistemos administratoriaus teises) gali matyti visus katedros dėstytojų sukurtus modulius ir atlikti analogiškus veiksmus su duomenų bazėje esančių moduliais. Taigi, katedros vedėjas turėdamas priėjimą prie visų dėstytojų individualių modulių planų atlieka patvirtino ar atmetimo funkciją, taip kontroliuodamas katedroje dėstomų modulių sąrašą. Veiklos sąveikų modelis pavaizduotas 1 pav.



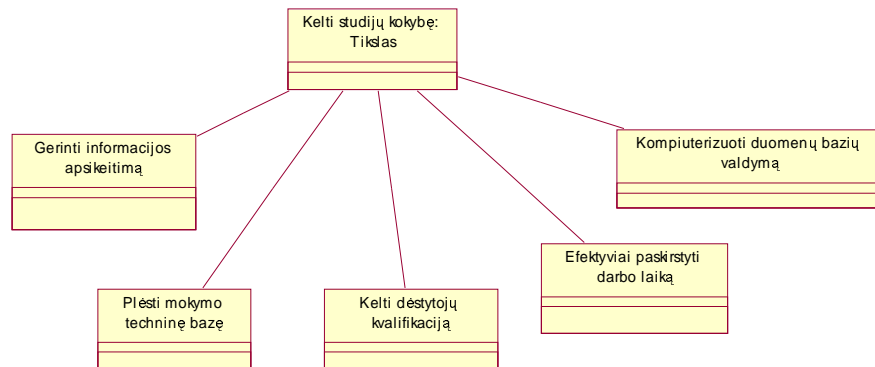
1 pav. Veiklos sąveikų modelis

2.1.3. Veiklos tikslų modelis

Kompiuterių katedros darbo organizavimo kokybės kėlimą sudaro tokie tikslai (uždaviniai) (2pav.):

- a) Apsikeitimo informacijos gerinimas – efektyvus informacijos dalinimas leidžia efektyviai išdėstyti mokymo medžiagą.
- b) Techninės mokymo bazės plėtimas, leidžia mokymui naudoti naujausią techninę įrangą

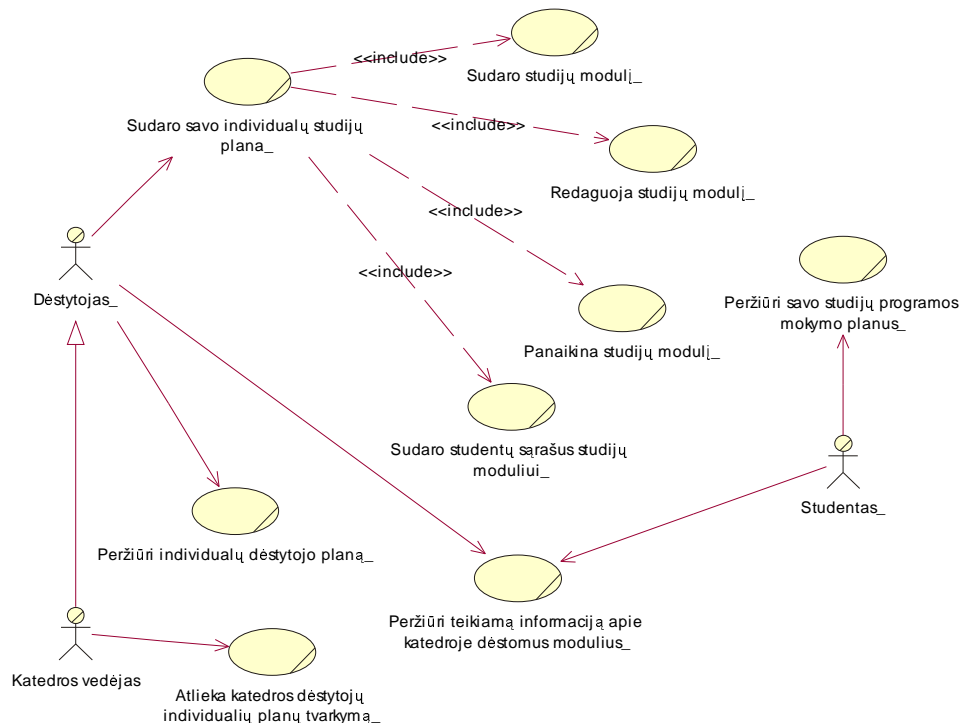
- c) Dėstytojų kvalifikacijos kėlimas
- d) Efektyvus laiko paskirstymas leidžia gerai suplanuoti laiką skirtą konsultacijoms ir pan.
- e) Naujas tikslas - kompiuterizuoti duomenų bazės valdymą



2 pav. Veiklos tikslų modelis

2.1.4. Veiklos panaudojimo atvejų modelis

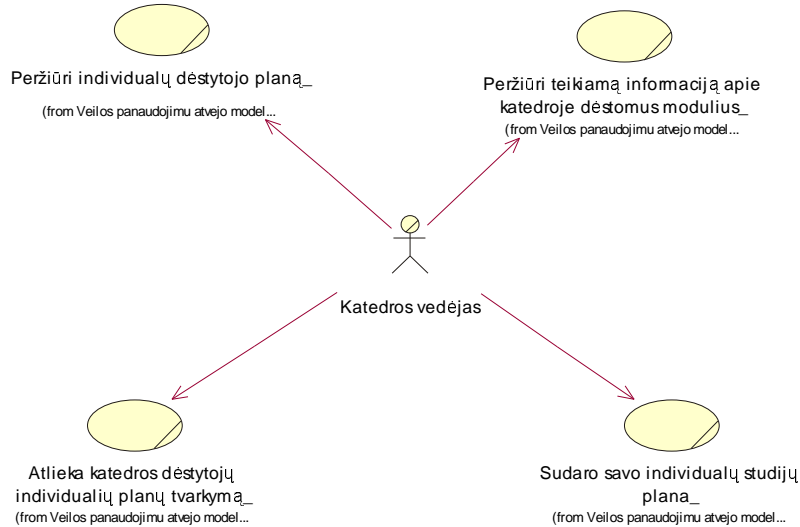
Kaip ir bet kuri kita organizacija, kompiuterių katedra siekia didinti darbo kokybę ir našumą. Tai padaryti yra daug būdų. Vienas iš jų yra veiklos kokybės gerinimas. Pagrindiniai kompiuterių katedros veiklos dalyviai yra dėstytojai ir studentai. Pagrindinė katedros veikla yra žinių perdavimas studentams, bei dėstytojų profesinis tobulėjimas.



3 pav. Veiklos panaudojimo atvejų modelis

2.1.4.1. Katedros vedėjo veiklos panaudojimo atvejo modelis

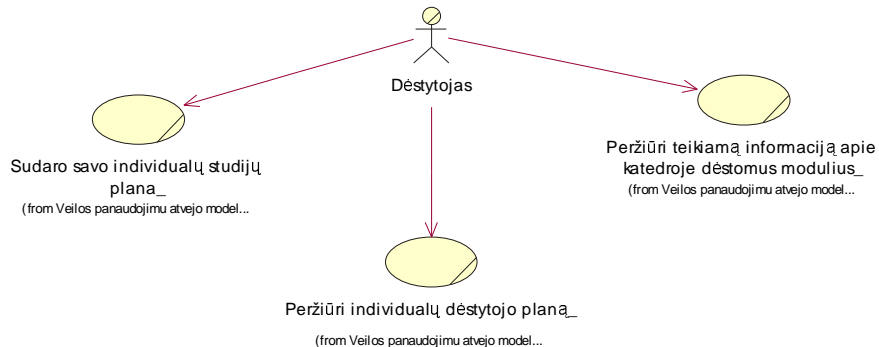
Veiklos panaudojimo atvejo diagramoje pavaizduota organizacijos darbo procesai, kurios atlieka Kompiuterių katedros vedėjas. Jo pareigas galima būtų apibūdinti šiais procesais: Sudaro savo individualų studijų planą, peržiūri individualų dėstytojo planą, atlieka katedros dėstytojų individualių planų tvarkymą, peržiūri visų katedroje dėstomų modulių informaciją, nes tai aukščiausias pareigas einantis katedros asmuo.



4 pav. Katedros vedėjo veiklos panaudojimo atvejo modelis

2.1.4.2. Dėstytojo veiklos panaudojimo modelis

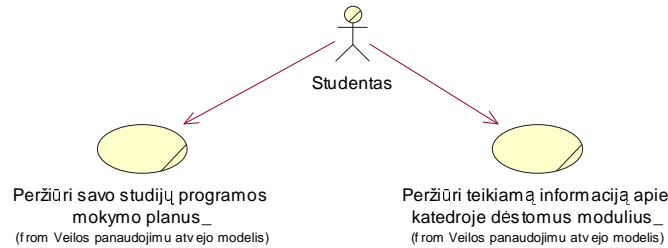
Iš veiklos panaudojimo atvejų diagramos matome, jog dėstytojas atlieka tokius organizacijos darbo procesus, kaip individualaus studijų plano sudarymą ir peržiūrą individualaus plano, bei gali peržiūrėti katedroje tik savo dėstomų modulių informaciją.



5 pav. Dėstytojo veiklos panaudojimo atvejo modelis

2.1.4.3. Studento veiklos panaudojimo modelis

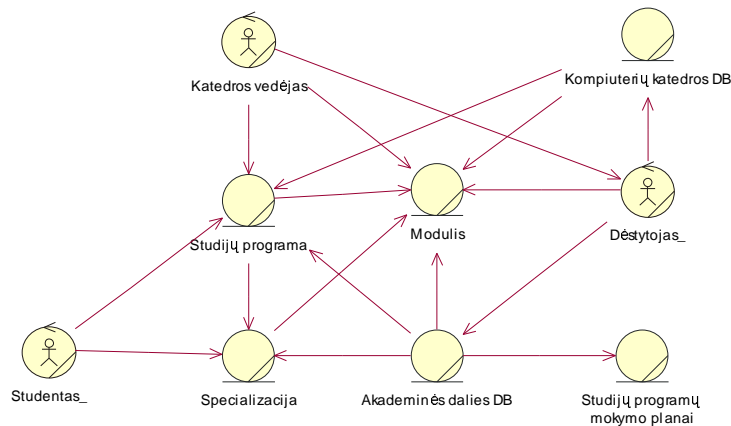
Iš veiklos panaudojimo atvejų diagramos matome, jog studentas atlieka tokius organizacijos darbo procesus: peržiūri savo studijų programos mokymo planus ir gali peržiūrėti katedroje teikiamą informaciją, tik apie jam dėstomus modulius.



6 pav. Studento veiklos panaudojimo atvejo modelis

2.1.5. Veiklos objektų modelis

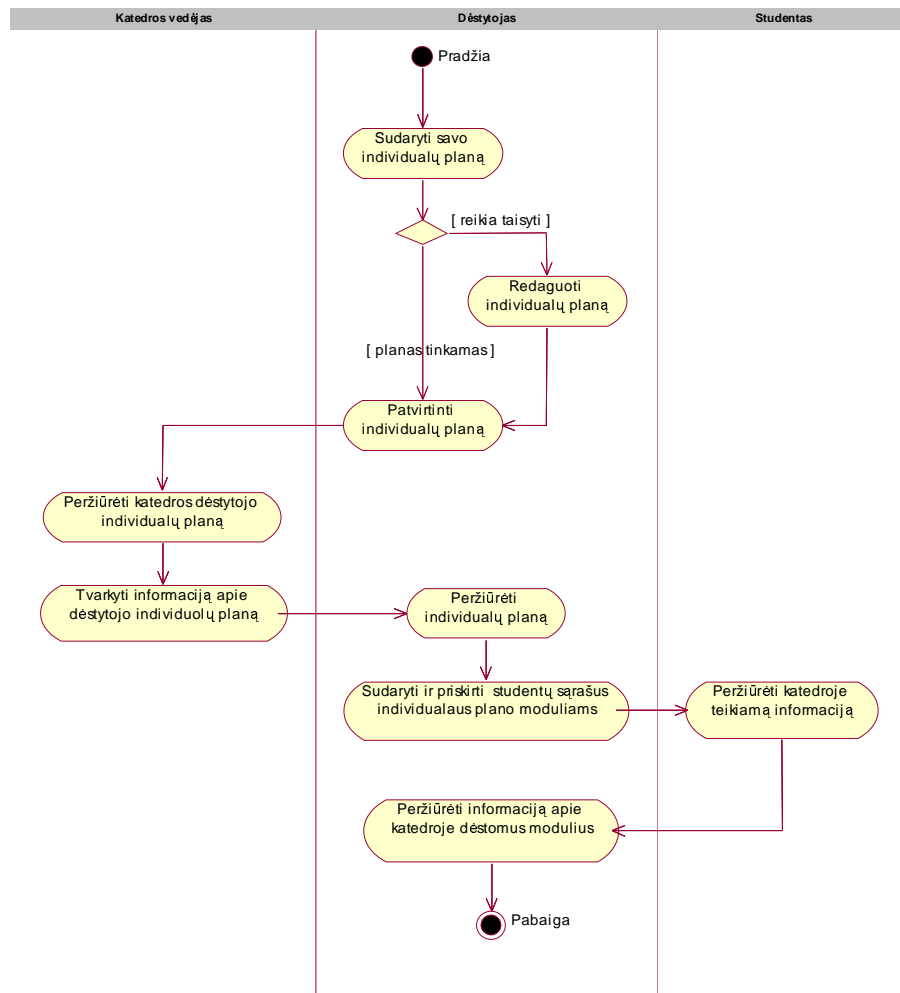
Pagrindiniai veiklos objektai yra katedros vedėjas, dėstytojas, studentas, akademinės dalies ir kompiuterių katedros duomenų bazės (7pav.). Studentas gali peržiūrėti savo studijų programos, visų studijų programų, specializacijų mokymo planus ir specializacijų specialybinius modulius. Katedros vedėjas gali peržiūrėti visų katedros studijų programų, specializacijų mokymo planus, atlikti informacijos apie studijų programas, specializacijas, modulius ir universiteto dėstytojus tvarkymą. Priklausomai nuo to, kur informacija įvedama, ją apdoroja akademinės dalies duomenų bazė, kurioje sukuriami studijų programos mokymo planai arba kompiuterių katedros duomenų bazėje, kur laikoma informacija tik su katedroje dėstomų studijų moduliais.



7 pav. Veiklos objektų modelis

2.1.7. Veiklos procesų modelis

Iš procesų veiklos diagramos matyti kaip ir kokie procesai vyksta kompiuterių katedroje (9 pav.).



9 pav. Veiklos procesų modelis

2.2. Literatūros šaltiniuose pateiktų sprendimų problemai spręsti lyginamoji analizė

Kadangi pasaulyje naudojamos įvairios platformos (aparatinė, programine ir informacine prasme), organizacijų bendravimas per kompiuterines sistemas įmanomas tik tuo atveju, kai nustatytas ryšys tarp jų leidžia informacijos mainus. [2]

EAI (Organizacijų kompiuterinių sistemų integracija) – tai neapribotas naudojimas bendrais susijungusių kompiuterinių sistemų verslo procesais ir duomenų šaltinių duomenimis.

Organizacijų kompiuterinės sistemos (OKS) gali būti integruojamos keliais būdais. Išskiriami tokie integravimo tipai:

1. Integravimas duomenų lygmenyje
2. Integravimas programinių sąsajų lygmenyje
3. Integravimas metodų lygmenyje
4. Integravimas vartotojo sąsajos lygmenyje

Integravimas duomenų lygmenyje – tai procesai, metodai ir technologijos, skirtos duomenų perkėlimui iš vieno duomenų saugyklų į kitas. Tai gali būti informacijos išgavimas iš vienos duomenų bazės, informacijos apdorojimas (jei reikalingas) ir atitinkamas kitos bazės pakeitimas. Duomenų transformavimas gali būti daromas atsižvelgiant į OKS verslo logiką.

Šio EAI tipo privalumas yra pigumas, nes nėra daromi taikomų programų pakeitimai, testavimai ir įdiegimai. Taip pat yra sukurta daugybė priemonių ir technologijų, kurias galima panaudoti taikant duomenų lygio integravimą. Šios priemonės ir technologijos irgi palyginus yra nebrangios.

Šio integravimo tipo trūkumas yra tas, kad jis daugiau skirtas integruoti vidines įmonės dalis. Bandant integruotis su verslo klientais, iškyla patikimumo, kompleksškumo, valdomumo problemos.

Integravimas programinių sąsajų lygmenyje – tai metodai ir technologijos, apibrėžiančios taikomųjų programų bendravimą tarpusavyje, naudojant programines sąsajas (Application Programming Interface, API). Programinės sąsajos leidžia prieiti tiek prie verslo procesų, tiek prie duomenų. Tokio tipo EAI dažniausiai taikomas integruojant programų paketus, pvz. SAP, PeopleSoft ir Baan, kurie jau turi programines sąsajas, skirtas priėjimui prie procesų ir duomenų.

Šio tipo privalumai: OKS bendrauja tarpusavyje automatiškai ir užtikrina labiau kontroliuojamą informacijos apsikeitimą, nei duomenų lygio integracija; plačiai paplitusios tokios programinės sąsajos: Java RMI (Remote Method Invocation), CORBA (Common Object Request Broker Architecture) IIOP (Internet Inter-ORB Protocol), Microsoft DCOM (Distributed Component Object Model).

Trūkumai: šis integracijos tipas brangesnis už duomenų lygio integracijos tipą, nes reikia modifikuoti dalį taikomųjų programų; programinės įrangos gamintojai paprastai kuria skirtingų tipų programines sąsajas, todėl priėjimas prie procesų ar duomenų skirtingose taikomose programose skiriasi.

Integravimas metodų lygmenyje – tai naudojimas bendrai prieinama verslo logika. Tai gali būti vienas iš taikomosios programos metodų, prieinamas iš kitų programų. Tokiu atveju, nereikia modifikuoti visų programų, papildomai kiekvienai įdedant tokį metodą. Paprastai tokie bendrai prieinami metodai yra sudedami į taikomųjų programų serverius arba atskiriems programų metodams yra suteikiamas prieinamumas, taikant įvairias technologijas (pvz. *paskirstyti objektai*).

Privalumas – užtikrina aukščiausią integracijos lygį, kuris leidžia formalizuoti ir automatizuoti organizacijos darbą.

Trūkumas – palyginus, su kitais tipais, kur dažniausiai nereikia keisti vienos iš bendraujančių taikomųjų programų, metodų lygio EAI tipas reikalauja daugumos programų pakeitimų.

Integravimas vartotojo sąsajos lygmenyje – tai taikomųjų programų integravimas, naudojantis jų vartotojo sąsają. Priėjimas prie programos procesų ir duomenų galimas tik per jos vartotojo sąsają. Dažniausiai priėjimui prie vartotojų sąsajų yra naudojami terminalai. Jei vartotojo sąsaja sukurta naudojant objektines technologijas, tai programos gali prieiti tiesiogiai prie vartotojo sąsajoje matomų objektų.

Privalumai: taikomos technologijos yra seniai naudojamos, todėl daugelis problemų, tokių kaip greitis, patikimumas, išplečiamumas, yra išspręstos. Taikant šį integracijos tipą, nereikia modifikuoti taikomųjų programų.

Trūkumai: tai pats primityviausias EAI tipas, dažnai laikomas nestabiliu ir atgyvenusiu.

J2EE Integration (JInt) Demo projektas

“J2EE Integration Demo” (JInt) [5] pateikia integravimo scenarijų tarp dviejų kompanijų „Pluto Motors“ ir „Marvel Accessories“. Tai parodo, kaip galima atlikti integraciją tarp dviejų kompanijų vidinių sistemų. Abi šios sistemos yra J2EE (Java 2 Platform, Enterprise Edition) pagrindu, tačiau integracijos funkcija gali būti išplėsta heterogeninių sistemų integravimui, kadangi sprendimas yra atliktas remiantis atvirais standartais, tokiais kaip SOAP, XML ir J2EE.

Taikymo scenarijus

Kompanija „Pluto Motors“, kuri turi savų pirkėjų duomenų bazę, įsigijo kitą kompaniją - „Marvel Car Accessories“. Kaip dalį įsigijimo proceso „Pluto Motors“ kompanija nori pasiūlyti savo produktus „Marvel Car Accessories“ kompanijos esamiems klientams. „Pluto Motors“ kompanija sugalvojo organizuoti infrastruktūrą egzistuojantiems „Marvel Car Accessories“ kompanijos klientų poreikiams aprūpinti, greitu ir efektyviu būdu be migracijos ar keitimo organizavimo. Šių dviejų kompanijų taikomieji uždaviniai yra realizuoti nevienodose platformose.

„Pluto Motors“ išlaikė vartotojo sąsają, kur klientai gali užsisakyti „Pluto Motors“ ir „Marvel Car Accessories“ kompanijų produktus. Šie užsakymai bus atskirti ir nusiusti atitinkamoms taikomosioms programoms. „Pluto Motors“ taip pat suteikia galimybę klientams savo užsakymų būsenų pasižiūrėjimą per internetą. Kaip dalis administravimo sąsajos „Pluto Motors“ leidžia administratoriams su tinkama autentifikacija peržiūrėti biznio statistiką. Visa užsakymų ir atsakymų komunikacija tarp „Pluto Motors“ ir „Marvel Car Accessories“ kompanijų taikomųjų uždavinių yra atliekama per HTTP protokolą.

Sistema po įsigijimo ir prieš integraciją

- Klientai talpina automobilių užsakymus į „Pluto Motors“ ir detalių užsakymus į „Marvel Car Accessories“ internetinius puslapius.
- Kasdieninės transakcijos siunčiamos per e-paštą ar faksą į „Pluto Motors“, kas sukelia didelių nepatogumų.
- „Marvel Car Accessories“ praneša biznio sprendimus, tokius kaip produkto kainą, naudojant elektroninį paštą ar faksą.
- Apsunkintas biznio veiklos stebėjimas tiek „Pluto Motors“, tiek „Marvel Car Accessories“ kompanijų.

Sprendimai - sistema po integracijos

- „Pluto Motors“ klientai gali talpinti detalių užsakymus toje pačioje vietoje, kur ir užsisako automobilį.
- „Pluto Motors“ klientai gali matyti savo užsakymų statusą šios kompanijos internetiniame puslapyje.
- Transakcijų informacija iš „Marvel Car Accessories“ yra automatiškai siunčiama „Pluto Motors“ kompanijai.

- „Marvel Car Accessories“ informuojama automatiškai apie biznio sferos sprendimus priimtus „Pluto Motors“ kompanijoje.

Kaip dalis integracijos yra pristatomi du komponentai pavadinant „Integracijos Brokeriu“ (Integration Broker) ir „Integracijos Agentu“ (Integration Agent)

Klientų atlikti užsakymai bus patalpinami į duomenų bazes naudojantis „Užsakymų išlaikymu“ (Order Persistence) ir „Tyrimo moduliu“ (Enquiry module). Užsakymai bus asinchroniškai pasiusti atitinkamoms taikomosios programoms („Pluto“ / „Marvel“) naudojant „Dispečerio modulį“ (Dispatcher module)

Ryšys tarp Pluto pagrindinės sistemos ir „Integracijos Brokerio“ sudaromas naudojant tiesioginį kreipimosi metodą (RPC Užklauso - Atsakymo), kur komunikacija tarp „Integracijos Brokerio“ ir „Integracijos Agento“ įgyvendinta naudojant Web serviso paslaugų technologijas.

Užsakymų gavimui realizuotas „Integracijos Agento“ komponentas - „Užsakymų gavimo paslauga“ (Order Receiver Service (internetinė paslauga)), paverčia užsakymus į atitinkamą formatą suprantamą „Marvel“ pagrindinės sistemos komponento. Toliau taip konvertuotą užsakymo informaciją siunčia į „Marvel“ pagrindinės sistemos „Užsakymų Valdymo Modulį“ (Order Management Module), kuri atnaujina duomenų bazę.

Sistemoje egzistuoja dviejų tipų administratoriaus vaidmenys :

- Prižiūrėtojas (Supervisor);
- Administratorius (Administrator).

Prižiūrėtojas naudojantis administravimo sąsają, priskiria užsakymus darbuotojams prie „Pluto“ ir „Marvel“ kompanijų sandelio (warehouse). Tai pasiekta naudojant „Užsakymo vykdymo modulį“ (Order Execution Module), kuris atnaujina duomenų bazę su užsakymo priskyrimo statusu. Užsakymo įvykdymas yra bendras tiek „Pluto“, tiek ir „Marvel“ sistemoms.

Administratorius naudojantis administravimo sąsają „Pluto“ sistemoje gali keisti informaciją apie produktą, kuris yra „Pluto“ ir „Marvel“ sistemų. Pakeista produkto informacija bus nusiusta į „Integracijos Agento“ komponentą „Produkto kainos gavėjo“ servisą. Prieš išsiusdamas atnaujintą produkto kainą, „Produkto kainos gavėjas“ (Product Price Receiver) konvertuoja informaciją į atitinkamą formatą suprantamą „Marvel Core“ komponentui. Toliau tai siunčiama „Produkto valdymo moduliui“ (Product Management Module) esančiam „Marvel Core“ komponente, kuris atnaujina duomenų bazę.

„Integracijos Brokeris“: (Aktorius sukurtas integracijos tikslui)

Šis komponentas suteikia tokias funkcijas:

- Saugo užsakymus patalpintus klientų visuotiniam stebėjimui.
- Atskiria ir paskirstyto užsakymus į „Pluto“ / „Marvel“.
- Priima ir išlaiko produkto informacijos atnaujinimą.
- Priima ir išlaiko transakcijų elementus iš „Pluto“ ir „Marvel“ sistemų.

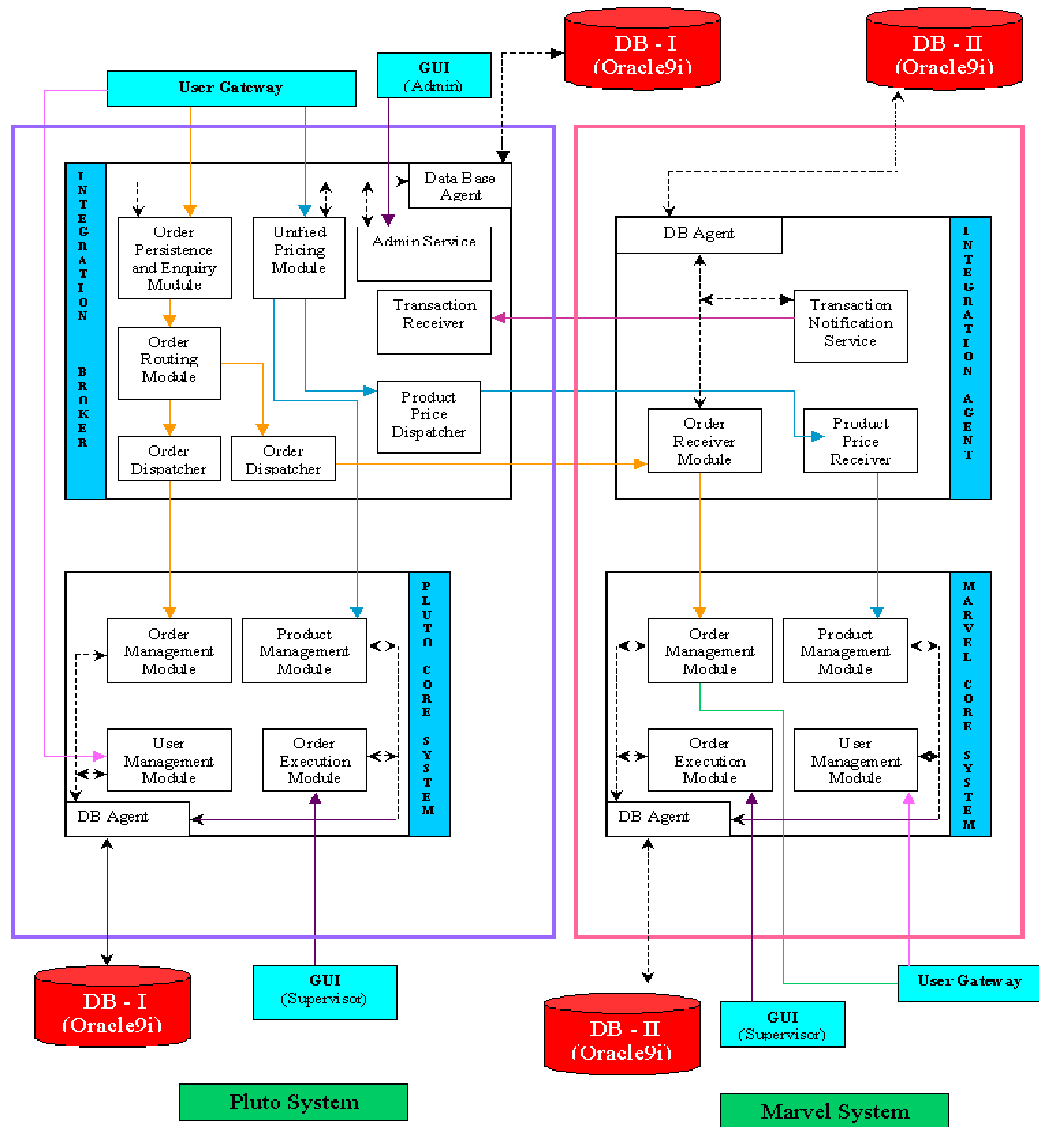
„Integracijos Agentas“: (Aktorius sukurtas integracijos tikslui)

Šis komponentas suteikia tokias funkcijas:

- Priima ir paskirsto užsakymo informaciją iš „Integracijos Brokerio“ į „Marvel Core“ sistemą.
- Priima ir paskirsto atnaujintą produkto informaciją iš „Integracijos Brokerio“ į „Marvel Core“ sistemą.
- Surenka ir išsiunčia transakcijų elementus iš „Marvel“ sistemos į „Integracijos Brokerį“, jam suprantamu formatu.

„Pluto“ ir „Marvel“ kompanijų egzistuojančių pagrindinių sistemų modulių integracija pavaizduota 10 paveiksle.

Architektūros diagrama :



10 pav. „Pluto Motors“ ir „Marvel Accessories“ integracijos modelis.

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 būtų naudoti XML [7, 11].

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 funkcinius reikalavimus. Reikia sukurti bendrą šabloną, kuriuo remiantis būtų įmanoma atlikti keitimąsi duomenimis tarp įvairių sistemų. Naudojant XML šiuo šablonu patogiausia naudoti XML schemas.

XML schemų paskirtis

XML schema nusako XML dokumento struktūrą. XML schemas skirtos apibrėžti [4]:

- 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

Kaip matome iš XML schemas paskirties aprašymo jos struktūra ir sandara yra labai artima reliacinių duomenų bazių struktūrai. Todėl norint keistis duomenimis reikia sukurti XML schemą, kuri leistų perduoti duomenis į bet kurią sistemą. Tačiau dėl nevisiško XML schemų ir reliacinių bazių struktūros atitikimo iškyla konvertavimo problemos.

Kadangi XML yra kalba, leidžianti struktūrizuoti duomenis, DTD (Document Type Definition) bei XML schemas yra naudojamos apibrėžti teisingus/leistinus elementus, kurie gali būti naudojami XML dokumente. Jei XML dokumentas atitinka apibrėžtą DTD schemą yra laikoma, kad XML dokumentas yra teisingas. [1]

```
Paprastas XML failo fragmentas  
<student num="54874">  
<name>Saulius Burba</name>  
<age>21</age>  
</student>
```

DTD - tai rinkinys taisyklių, kurios apibrėžia leidžiamą dokumento klasių struktūrą, turi skirtingą sintaksę nei XML failai. DTD apibrėžia XML elementų eiliškumą bei tinkamumą.

```
DTD fragmentas  
<!ELEMENT student (name, age)>  
<!ATTLIST student num CDATA>  
<!ELEMENT name (#PCDATA)>  
<!ELEMENT age (#PCDATA)>
```

Daugiausiai kritikos DTD sulaukė dėl to, kad ji naudoja skirtingą sintaksę nei XML kalba. Kaip pakaitalas DTD buvo sukurta XDR (XML Data Reduced), kuri turėjo tokią pačią sintaksę, kaip ir XML.

```
XDR fragmentas  
<Schema name="myschema" xmlns="urn:schemas-microsoft-com:xml-data"  
xmlns:dt="urn:schemas-microsoft-com:datatypes">  
<ElementType name="age" dt:type="uil" />  
<ElementType name="name" dt:type="string" />  
<AttributeType name="num" dt:type="string" />  
<ElementType name="student" order="seq">  
<element type="name" minOccurs="1" maxOccurs="1"/>  
<element type="age" minOccurs="1" maxOccurs="1"/>  
<attribute type="num" />  
</ElementType>  
</Schema>
```

XML produktai ir jų kategorijos

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

- **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.

Tarpininkavimo programos (TP)

TP yra programinė įranga, skirta duomenų perdavimui tarp XML dokumentų bei duomenų bazės. Dažniausiai naudojamos reliacinės DBVS naudojančios ODBC, JDBC arba OLE DB. Tačiau yra ir kitokių produktų, skirtų duomenų mainams tarp kitokio tipo DB. Tačiau tokie produktai nėra labai paplitę. Žemiau esančioje lentelėje pateiktas TP produktų palyginimas[8].

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	IBM	Analizei	Reliacinis	x	--

Extractor (XLE)					
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.

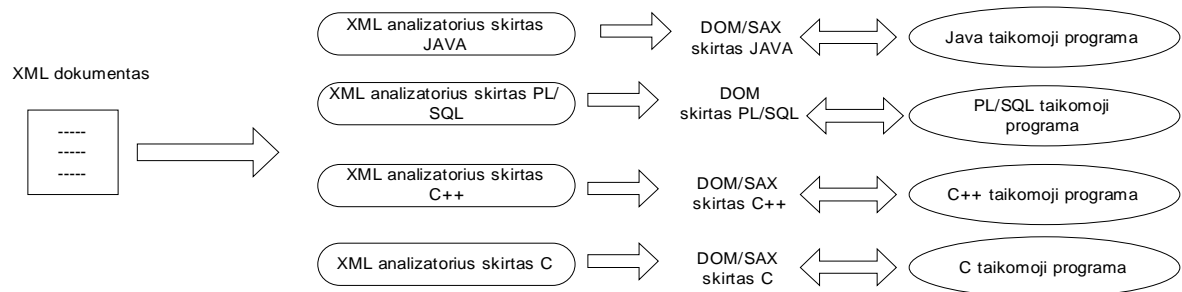
Mysql, mysqldump

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

XML panaudojimas Oracle duomenų bazėse

XML analizatoriai (XML Parsers)

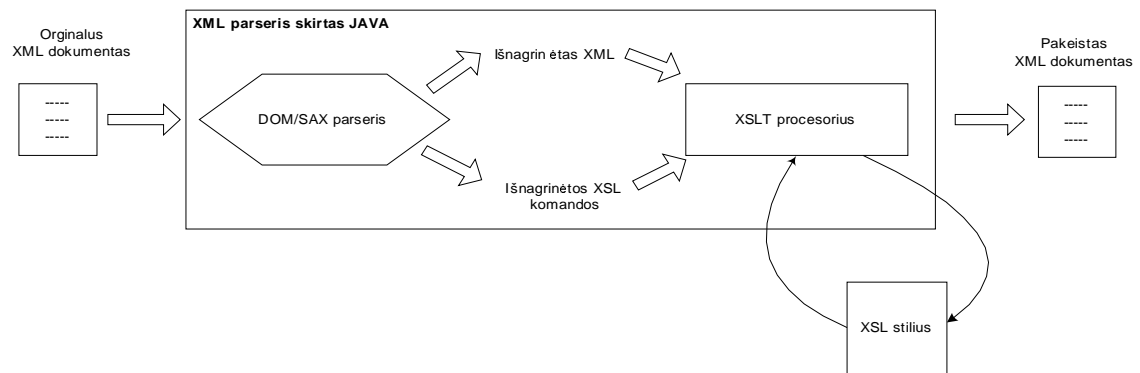
Oracle turi XML analizatorius Java, C, C++ bei PL/SQL kalboms [9]. Kiekvienai kalbai yra naudojamas atskiras komponentas, kuris nagrinėja ir sutvarko XML dokumentus taip, kad juos būtų galima apdoroti su kitomis aplikacijomis perkeliant į DB. Analizatoriai palaiko DOM (Document Object Model) objektų, bei SAX (Simple API for XML) sąsają. XML „Namespaces“, XSL transformacijos taip pat yra palaikomos Oracle platformose (11 pav.).



11 pav. XML analizatoriai

XSL transformacijų palaikymas

Pradedant antrąją versiją, XML analizatoriaus turi integruotą XSL transformacijų (XSLT) procesorių. Jo uždavinys – transformuoti esamų XML dokumentų panaudojant XSL stilių failą. Su XSLT galima transformuoti XML dokumentą į XML, HTML arba bet kokią kitą tekstinį failą (12 pav.).



12 pav. XSL transformacijų palaikymas

„Namespace“ palaikymas

Java, C ir C++ analizatoriai taip pat palaiko ir XML „Namespaces“. Tai yra toks mechanizmas, kuris apsprendžia arba panaikina vardų kolizijas XML dokumentuose tarp elementų tipų (tags), bei atributų. Šis mechanizmas nurodo pagrindinius elementų tipus, bei atributus kurie galioja visam XML dokumentui. Tokie elementai yra nusakomi URIs (uniform resource identifiers), pvz.:

```
<oracle:EMP xmlns:oracle="http://www.oracle.com/xml"/>.
```

Šis pavyzdys gali būti naudojamas nusakant Oracle <EMP> duomenų elementą, kuris yra skirtingas nei kitos kompanijos tas pats <EMP> elementas.

DOM ir SAX aplikacijų programavimo sąsajos (Application Programming Interface.)

XML aplikacijų programavimo sąsajos (APS) skirstoma į dvi kategorijas: įvykių tipo (event-based) ir medžio tipo (tree-based). Pirmojo tipo API (tokios kaip SAX) naudojamos pranešimams apie analizatoriaus įvykius generuoti. Aplikacijos apdoroja įvykių informaciją bei valdo įvykių elgseną. Įvykiai saugo savyje informaciją apie elementų pradžią ir pabaigą, bei patį elemento turinį. Priešingai nei medžio tipo APS, įvykių tipo APS nesaugo atmintyje informacijos apie medžio struktūrą, bei XML dokumento atvaizdavimo detalių.

Medžio tipo APS (tokios kaip DOM) saugo atmintyje XML dokumento atvaizdavimo medį. Šios APS turi integruotas klases bei metodus, kurie leidžia valdyti XML struktūrinį medį (navigacija per visus elementus ir pan.). DOM daugiausiai yra naudojamas kai reikia atlikti veiksmus su XML dokumento struktūra (perrūšiuojant elementus, pervardinant elementus arba atributus ir pan.). Oracle 9i XDK palaiko tiek DOM 2.0, tiek SAX 2.0 specifikacijas

Standartiniai duomenų bazių įrankiai duomenų importavimui ir eksportavimui XML formatu

Efektyvus ir automatizuoto XML duomenų konvertavimo į ir iš reliacinių duomenų bazių problemai spręsti egzistuoja keletas būdų [3]. Duomenų bazių kūrėjai, kaip IBM, Microsoft, Oracle ir Sybase yra išvystę programinius įrankius padedančius konvertuoti XML dokumentus į reliacines lenteles. Keletas siūlomų sprendimų:

- Oracle XML SQL įrankis modeliuoja XML dokumentų elementus kaip lizdinę lentelių kolekciją. Pridėtas elementas yra modeliuojamas kaip Oracle objekto duomenų tipas. „SQL į XML“ konvertavimas sudaro XML dokumentą naudojant vienas su vienu asociaciją tarp lentelių, nurodant objekto duomenų tipą ir lizdinį elementą. „XML į SQL“ reikalauja arba duomenų modelio redagavimo (konvertuojant jį iš reliacinio į objektiškai orientuotą) arba pertvarkymo originalaus XML dokumento.
- IBM DB2 XML plėtinys leidžia talpinti XML dokumentus kaip BLOB objektus arba kaip suskaidytus į lenteles. Vėliausia transformacija, žinoma kaip XML kolekcija, apibrėžta XML 1.0 sintaksės.
- Microsoft sprendžia problemą praplėsdamas SQL-92 ir pateikdamas OPENXML eilučių seką.
- Sybase Adaptive Server įveda „ResultSetXml“ Java klasę kaip abiejų krypčių XML konvertavimo bazę.

XML dokumentų konvertavimas yra simetrinis, o talpinimas neleidžia modifikuoti daugiau kaip vieną lentelę. XML išgavimas transformuoja SQL užklauso rezultatus į plokščios struktūros dokumentą.

Kūrėjų palyginimas pateiktas 3 lentelėje.

3 lentelė. Kūrėjų palyginimai

Kūrėjas	Konvertavimo taisyklės	Viena lentelė / daug lentelių	Konvertavimo priemonės	Simetrinis konvertavimas
Oracle	Besąlygiškai; sudarant objektų – reliacinį duomenų modelį	Daug	Nustatytos Java klasės	Simetrinis, jei XML dokumentas ir objektų – reliacinis modelis derinasi
IBM	Duomenų priėjimo apibrėžimo failas	Daug	Suprojektuotos talpinimo procedūros	Simetrinis
Microsoft	SQL išplėtimas; row set funkcijos	Daug išgaunant XML Viena talpinant XML	Naudojant SQL construct FOR XML ir row set OPENXML	Asimetrinis
Sybase	DTD	Viena; užklausa gali apimti kelias lenteles	Naudojant Java klases	Simetrinis

Bendros savybės:

1. Visos priemonės importuoja ar eksportuoja duomenis į ar iš duomenų bazės, tačiau neturi galimybių tai atlikti pagal tam tikrą logiką, esant poreikiui transformavimo metu atlikti tam tikrus veiksmus ir pagal tai spręsti kokius duomenis naudoti;
2. Šios priemonės netinka esant poreikiui sujungti kelių skirtingų technologijų duomenų bazių duomenis;
3. Nėra bendrų taisyklių konvertuojant duomenų bazes į XML; specialus žymėjimas reikalingas, jei XML dokumentas naudoja naują sintaksę;

Išvados.

Apžvelgus esamus produktus, duomenų perdavimui ir integravimui XML formatu, galima daryti sekančias išvadas:

- XML kalba tinkama duomenų keitimuisi tarp heterogeninių duomenų bazių. Tačiau šis procesas nėra toks paprastas, nes reikia atsižvelgti į įvairių sistemų funkcinius reikalavimus.
- Pateiktas sąrašas tarpininkavimo programų (Middleware), skirtų duomenų perdavimui tarp XML dokumentų bei duomenų bazių. Tačiau, kaip ir su kiekviena programine įranga reikia susipažinti ir išsivinti jos veikimo specifiką, o tai užima laiko.
- Duomenų bazių įrankiai duomenų apskeitimui XML formatu dažniausiai neturi galimybių tai atlikti pagal tam tikrą logiką, nėra numatyta transformavimo metu atlikti tam tikrus veiksmus ir pagal tai spręsti kokius duomenis naudoti;
- Standartiniai duomenų bazių įrankiai netinka kelių skirtingų technologijų duomenų bazių duomenų sujungimui, eksportuojant duomenis į XML formatą;

3. Darbo tikslas ir jo pagrindimas, kokybės kriterijų apibrėžimas

Darbo tikslas – išspręsti problemas, atsirandančias, bet kuriuo metu kuriant naujus modulius ar atnaujinant senus, atliekant veiksmus su studentų sąrašais mokymo proceso eigoje. Tam, kad nereiktų suvedinėti tos pačios informacijos KTU studijų modulių IS ir KTU kompiuterių katedros modulių IS, planuojama sukurti studijų modulių planavimo ir valdymo sistemą, kuri integruotų minėtas abi sistemas į vieningą visumą vartotojo atžvilgiu. Šios sistemos pagrindinis tikslas taisyklingai suformuoti XML duomenų srautus ir atlikti apsikeitimą tarp dviejų duomenų bazių, kurie nepriklausytų nuo abiejų IS funkcijų ir architektūros.

Kad įgyvendinti projekto tikslą reikia:

- atlikti tyrimų srities analizę;
- išsiaiškinti keliamus kuriamai sistemai reikalavimus;
- suprojektuoti kuriamą sistemą;
- atlikti sukurtos programinės įrangos testavimą ir įdiegimą.

Pagrindinis darbo kokybės kriterijus – taisyklingai veikianti programinė įranga. Pagrindinė sukurtos programinės įrangos funkcija bus taisyklingai suformuoti XML duomenų srautus ir atlikti apsikeitimą tarp dviejų duomenų bazių, nepriklausomai nuo IS funkcijų ir architektūros.

Sukurta programinė įranga turi būti patikrinta ar atitinka jai keliamus tokius reikalavimus:

- ar atitinka jai keliamus funkcinius reikalavimus;
- ar yra atliekama klaidų kontrolė;

4. Projektavimo metodų, priemonių parinkimas

Projektavimo metodų pasirinkimas atliktas pateikus F³, MERODE ir RUP metodų palyginimo kriterijus. Trumpai apie kiekvieną metodą aptarsiu šiame skyriuje.

F³ – tai metodika, kuri skirta padėti informacinės sistemos vartotojams, reikalavimų savininkams ir reikalavimų inžinieriams gauti suderintą ir patvirtintą kuriamos sistemos specifikaciją sudarant integruotą reikalavimų inžinerijos specifikaciją. Tačiau ši metodika nėra objektinė.

R.Gusto metodikoje yra svarbios būsenos – pagal jas specializuojami objektai ir tokiu būdu specifikuojamos veiksmų vykdymo prieš ir po sąlygos. Viena iš pagrindinių R.Gusto metodikos idėjų – pragmatinių, statinių ir dinaminių priklausomybių integravimas vienoje diagramoje.

MERODE – tai modeliu pagrįsta sistemos kūrimo metodika. Pasižymi abstrakčia objektų sąveikos sąvoka ir formaliu visų priemonių aprašymu. Čia išspręstos tokios problemos, kaip savybių, garantuojančių specifikacijos kokybę, trūkumas; objektiškai orientuotos analizės priemonių polinkis į realizaciją; aukšto lygio agregavimo priemonių trūkumas taikant objektiškai orientuotą metodiką.

RUP - yra panaudojimo atvejų valdomas, architektūra grindžiamas, iteracinis, vykdomas palaipsniui, riziką mažinantis projektavimo procesas. Tai pilnai objektinė metodika.

Informacinės sistemos kūrimo proceso atžvilgiu palyginimas pateiktas 4 lentelėje.

4 lentelė. Palyginimas IS kūrimo proceso atžvilgiu

Etapas	F3	R.Gusto	MERODE	RUP
Organizacijos veiklos analizė	Apima	Galima naudoti	Pritaikoma	Apima
Sistemos analizė	Apima	Galima naudoti	Pritaikoma	Apima
Sistemos projektavimas	Apima	Iš dalies apima	Pritaikoma	Apima
Sistemos realizavimas	Tik prototipų sukūrimas	Neapima	Pritaikoma	Apima
Sistemos testavimas	Galima atlikti trasavimą nuo reikalavimų iki realizuojamų komponentų	-	-	Apima

Atlikus metodų palyginimą galima daryti išvadą, kad RUP metodas plačiausiai apima informacinių sistemų inžineriją.

ADO.NET (Microsoft) ir JDO (Sun Sysytems) produktų lyginamoji analizė pateikta 5 lentelėje [6]

5 lentelė ADO.NET ir JDO produktų lyginamoji analizė

	ADO.NET	JDO (Java Data Objects)
Duomenų modelis	Reliacinis	Objektinis
Duomenų apibrėžimo kalba (<i>Data Definition Language</i>)	XML	Java ir XML
Užklausų kalba	SQL	JDOQL
Duomenų manipuliavimo kalba	SQL ir (C# arba C++ arba Visual Basic)	Java
Paveldėjimo ir polimorfizmo realizacija	Ne	Taip
Unikalus identifikatorius	Pirminis raktas	Pirminis raktas arba duomenų šaltinio identifikatorius
Transakcijos	Taip	Taip
Konkurentiškumas	Taip	Taip
Objektinio duomenų modelio palaikymas užklausose	Ne	Taip

Atlikus ADO.NET (Microsoft) ir JDO (Sun Sysytems) produktų lyginamąją analizę pasirinkta JDO palaikanti Java technologija. Nors ir JDO technologijos trūkumas yra du šaltiniai, kadangi kuriant Java klases sąsajai su duomenų šaltiniu naudojamas Java išeities tekstas ir XML byloje užrašyti metaduomenys. Tačiau JDO turi paveldėjimo ir polimorfizmo realizaciją ir objektinio duomenų modelio užklausų palaikymą.

5. Kompiuterizuojamos sistemos varianto parinkimas

KTU kompiuterių katedros modulių informacinei posistemei naudojama reliacinė duomenų bazių valdymo sistema MySQL. KTU studijų modulių IS naudoja ORACLE Server 8.1 duomenų bazę.

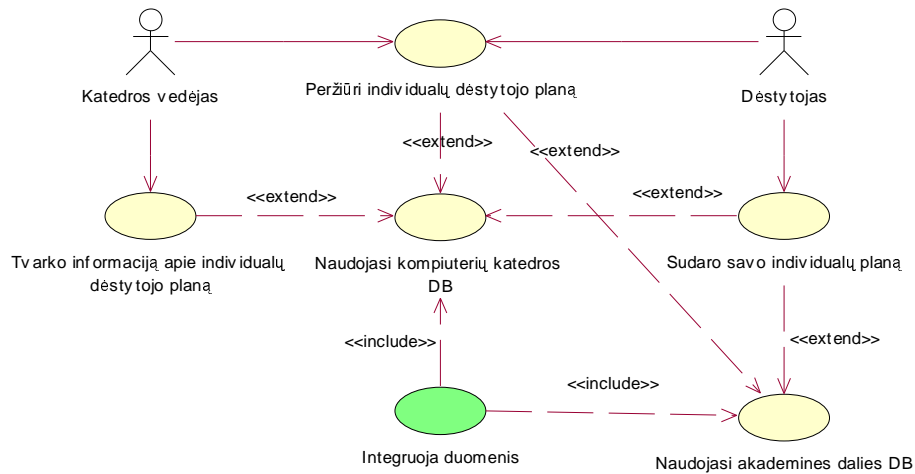
Projektavimui ir realizavimui naudojamų priemonių pasirinkimą lėmė tai, jog visa šiame projekte naudota PĮ yra nemokama ir laisvai prieinama. Tai mažina visos sistemos kūrimo kainą ir garantuoja, kad kuriama PĮ nepažeidžia jokių autorinių teisių. Visos priemonės yra realizuotos naudojant Java objektinę technologiją, todėl yra lanksčios, užtikrinančios spartų projekto realizavimą ir vystymą.

Java kodo rašymui bus naudojama „Eclipse Platform“ teksto redaktorius. Jis palaiko įvairių priemonių konstrukciją taikomųjų programų plėtojimui, nevaržomus nustatymus priemonių tiekėjams, įskaitant nemokamą programinę įrangą. Turi priemones dirbti su įvairių tipų failais (HTML, Java, C, JSP, EJB, XML ir GIF). Palengvina vientisą susijungimą priemonių viduje ir tarp skirtingų gamintojų turinių tipų ir priemonių. Palaiko grafines ir ne grafines programų kūrimo aplinkas. Paleidžiamas daugelyje operacinių sistemų. Eclipse platforma sukurta suradimui, integravimui ir paleidimui modulių vadinamų įskiepiais (angl. plug-in).

6. Darbo rezultatai

6.1. Kompiuterizuojamų veiklos procesų modelis

Kompiuterizuojamus organizacijos veiklos procesus vaizduoja panaudojimo atveju modelis, pateiktas žemiau. Iš jo matome, jog sistemos pagrindinės funkcija bus akademinės dalies ir kompiuterių katedros duomenų bazių tam tikrų duomenų integracija. Tai duomenų apjungimas susijęs su modulių saugoma informacija ir studentų sąrašais, kad nereiktų tos pačios informacijos kiekvieną kartą įvedinėti į skirtingas duomenų bazes (13pav.).

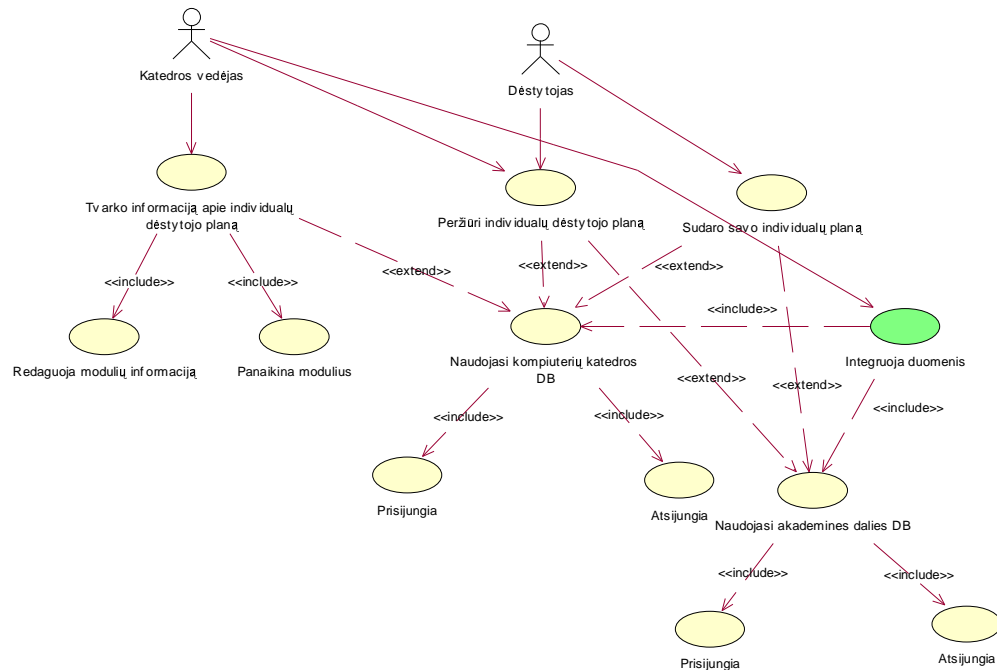


13 pav. Kompiuterizuojamų veiklos procesų modelis

6.2. Reikalavimų modelis

6.2.1. Vartotojų veiklos procesų modelis

Vartotojų veiklos procesų diagramoje matome, jog katedros vedėjas naudojantis kompiuterių katedros duomenų bazėje saugomą informaciją gali tvarkyti individualius katedros dėstytojų planus, t.y. juos redaguoti ar panaikinti. Dėstytojas prisijungęs prie sistemos gali sudaryti ir peržiūrėti savo individualų planą. Tuo tarpu naudojantis akademinės dalies duomenų bazės saugoma informacija, kaip ir paprastas dėstytojas, katedros vedėjas gali tik sudaryti savo individualų planą ir esant reikalui peržiūrėti. Tuo tikslu yra panaudojama duomenų integracija, kurios uždavinys apjungti tų pačių duomenų įvedimą į skirtingas duomenų bazes (14pav.).



14 pav. Vartotojų veiklos procesų modelis

6.2.2. Specifikacijos kiekvienam panaudojimo atvejui

Lentelėse (6 lentelė – 9 lentelė) aprašyti panaudojimo atvejai, kurie pateikti vartotojų veiklos procesų modelyje (14pav.).

6 lentelė. Dėstytojo veiklos procesų specifikacija, remiantis vartotojų veiklos procesų modeliu (14pav.)

Panaudojimo atvejis	Sudaro savo individualų planą
Aktorius	Dėstytojas
Sistema	KK portalo studijų modulių sistema

Prieš sąlyga	Dėstytojas turi būti įrašytas į informacinės sistemos duomenų bazę
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Suvedami duomenys apie individualų dėstytojo planą į formą 2. Patvirtinimo mygtuko paspaudimas	2.1. Sistema tikrina ar užpildyti būtinieji formos laukai 2.1. Sistema įveda duomenis apie individualų dėstytojo planą į duomenų bazę 2.2. Sistema praneša apie sėkmingą duomenų įvedimą
Po sąlyga	Individualus dėstytojo planas sudarytas. Katedros vedėjas gali peržiūrėti sudarytą planą ir atlikti informacijos apie individualų planą tvarkymą, t.y redaguoti ar panaikinti (šie veiksmai galimi naudojantis tik kompiuterių katedros duomenų bazėje saugoma informacija)
Alternatyvos (nesėkmės atvejai)	2.1.a Neužpildyti visi būtinieji laukai Sistema išmes pranešimą apie neužpildytus būtinuosius laukus ir grįžta į 1 žingsnį 2.1.b Laukai užpildyti neteisingai. Sistema išmes pranešimą apie neteisingai užpildytus laukus ir grįžta į 1 žingsnį
Vykdyto variantai	
Veiklos taisyklės	Vartotojas turi įvesti būtinuosius duomenis ir būtinai teisingai įvesti reikalingus duomenis
Specialūs reikalavimai (nefunkciniai)	Duomenys turi saugiai pasiekti duomenų bazę

7 lentelė. Katedros vedėjo veiklos procesų specifikacija, remiantis vartotojų veiklos procesų modeliu (14pav.)

Panaudojimo atvejais	Peržiūrėti individualų dėstytojo planą
Aktorius	Katedros vedėjas
Sistema	KK portalo studijų modulių sistema
Prieš sąlyga	Katedros vedėjas turi būti registruotas duomenų bazėje ir sudaryti individualūs dėstytojų planai
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Pasirenkamas dėstytojas, kurio individualų planą norime peržiūrėti (t.y jo modulį) 2. Patvirtinimo mygtuko paspaudimas	2.1. Sistema išveda pasirinkto dėstytojo individualaus plano duomenis
Po sąlyga	Peržiūrėti pasirinkto dėstytojo individualų planą, iš kurio galima pereiti į plano redagavimo langą.
Alternatyvos (nesėkmės atvejai)	
Vykdyto variantai	Dėstytojo sukurtą modulį galima peržiūrėti pasirenkant iš sąrašo pateiktų modulių.
Veiklos taisyklės	
Specialūs reikalavimai (nefunkciniai)	

Peržiūrėti individualų dėstytojo planą – analogiška individualaus plano peržiūrai, kuria atlieka katedros vedėjas (žr. lentelė 7), tik šiuo atžvilgiu aktorius yra dėstytojas.

8 lentelė. Katedros vedėjo veiklos procesų specifikacija, remiantis vartotojų veiklos procesų modeliu (14pav.)

Panaudojimo atvejis	Tvarko informaciją apie individualius dėstytojų planus
Aktorius	Katedros vedėjas
Sistema	KK portalo studijų modulių sistema
Prieš sąlyga	Katedros vedėjas turi būti registruotas duomenų bazėje ir sudaryti individualūs dėstytojų planai
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Tvarkoma informaciją apie individualų dėstytojo planą (su modulių aprašymais susijusi informacija pateikiama formoje yra redaguojama arba ištrinama) 2. Paspaudžiamas patvirtinimo mygtukas	2.1. Sistema tikrina ar užpildyti būtinieji formos laukai. 2.2. Sistema atnaujina dėstytojo individualaus plano informaciją duomenų bazėje. 2.3. Sistema praneša apie sėkmingą duomenų atnaujinimą.
Po sąlyga	Individualaus dėstytojo plano informacija sutvarkyta, t.y redaguota arba panaikinta.
Alternatyvos (nesėkmės atvejai)	2.1.a Neužpildyti visi būtinieji laukai Sistema išmes pranešimą apie neužpildytus būtinuosius laukus ir grįžta į 1 žingsnį 2.1.b Laukai užpildyti neteisingai. Sistema išmes pranešimą apie neteisingai užpildytus laukus ir grįžta į 1 žingsnį
Vykdomo variantai	Norimą dėstytojo modulį galima pasirinkti pateiktame modulių sąrašė.
Veiklos taisyklės	Vartotojas turi įvesti būtinuosius duomenis ir būtina teisingai įvesti reikalingus duomenis
Specialūs reikalavimai (nefunkciniai)	Redaguoti duomenys turi saugiai pasiekti duomenų bazę, o panaikinti turi būti pašalinti iš duomenų bazės.

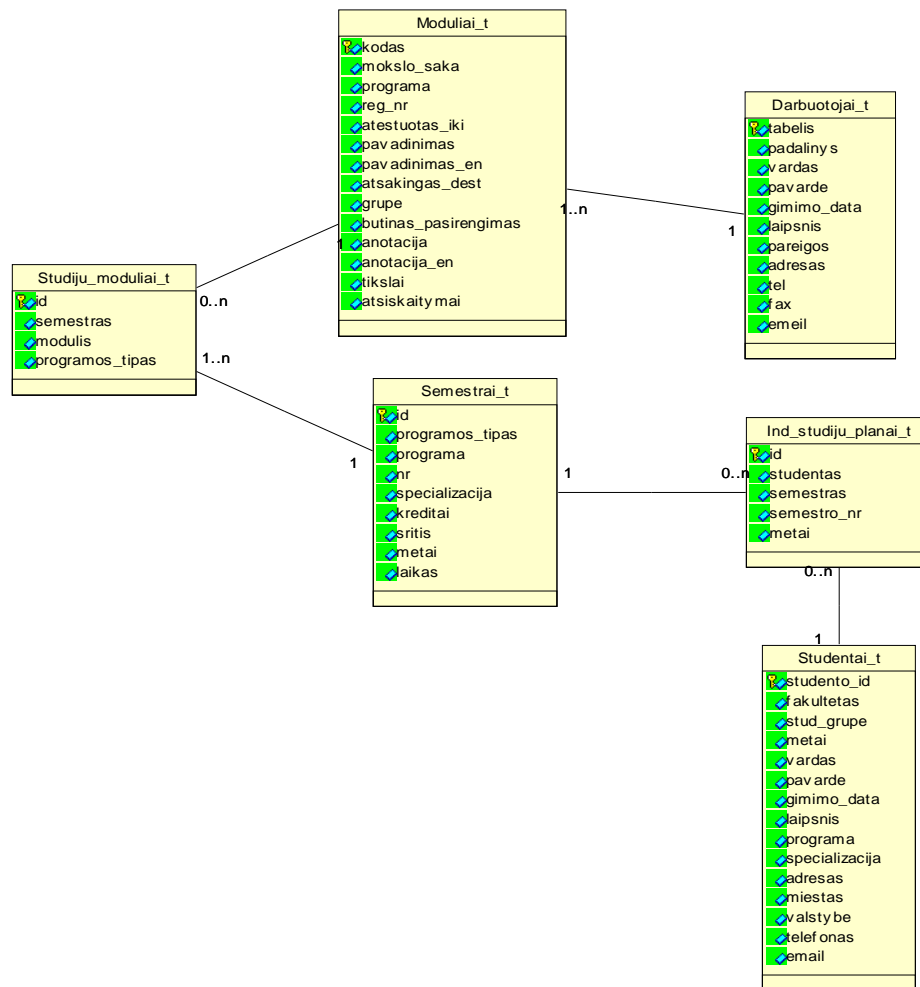
9 lentelė. Katedros vedėjo veiklos procesų specifikacija, remiantis vartotojų veiklos procesų modeliu (14pav.)

Panaudojimo atvejis	Integruoja duomenis
Aktoriai	Katedros vedėjas
Sistema	Studijų modulių planavimo ir valymo sistema
Prieš sąlyga	Vartotojas turintis teisę naudotis šia sistema
Pagrindinis įvykių srautas	Sistemos reakcija
1. Vartotojas pasirenka norimos duomenų bazės XML srauto suformavimą	1.1 Tikrinama ar yra suformuotas XML srautas norimos duomenų bazės, jei taip, pereinama prie 1.2 punkto 1.1.1 Jungiamasi prie KTU akademinės dalies duomenų bazės ir nuskaitomi reikiami duomenys 1.1.2 Jungiamasi prie KTU kompiuterių katedros duomenų bazės ir nuskaitomi reikiami duomenys 1.1.3 Nuskaityti duomenys susiejami i vieną bendrą reikiamos formos struktūrą 1.1.4 Suformuoti duomenys saugomi XML formate

	1.2 Nuskaitomas XML srautas 1.3 Iš XML srauto formuojami įrašai į vieną iš duomenų bazių.
Po sąlyga	Pakeista viena iš duomenų bazių
Alternatyvos (nesėkmės atvejai)	Nepavyko prisijungti prie vienos iš duomenų bazių, sistema siūlo jungimąsi pakartoti arba nutraukti;
Vykdomo variantai	Jei XML srautas suformuotas, jis nėra formuojamas
Veiklos taisyklės	Turi būti sudarytas priėjimas prie abiejų duomenų bazių; Turi būti įvesti duomenų bazių vartotojų slaptažodžiai;
Specialūs (nefunkciniai) reikalavimai	Patogi sąsaja su vartotoju, saugus ir savalaikis duomenų bazių pakeitimas.

6.2.3. Dalykinės srities statinis modelis

Dalykinės srities statinis modelis skirtas duomenų (data) lygmeniui, kuris nurodo saugomų duomenų klases. Žemiau pavaizduota KTU akademinės IS dalykinės srities duomenų klasės (15pav.).



15 pav. Dalykinės srities statinis modelis (KTU akademinės IS)

Klasė Darbuotojai_t aprašo KTU darbuotojus (tuos kurie gauna atlyginimą, tokie kaip dėstytojai), projektuojant DB atsižvelgti reikia į tai, kad galima būtų prijungti reikalingus laukus (10 lentelė).

10 lentelė. Klasė Darbuotojai_t.

Atributai	Reikšmė
tabelis	darbuotojo id
padalinys	padalinys, kuriame darbuotojas dirba
vardas	darbuotojo vardas
pavarde	darbuotojo pavardė
gimimo_data	darbuotojo gimimo data

Studijų modulių planavimo ir valdymo sistema. Magistro baigiamasis darbas.

laipsnis	žodžiu parašytas darbuotojo laipsnis
pareigos	darbuotojo pareigos. pvz, vedėjas
adresas	darbuotojo adresas
tel	nurodo darbuotojo telefono numerį
fax	nurodo darbuotojo fakso numerį
email	nurodo darbuotojo elektroninio pašto adresą

Klasė Moduliai_t aprašo dėstomus modulius, pvz.. Informacinės sistemos (11 lentelė).

11 lentelė. Klasė Moduliai_t.

Atributai	Reikšmė
kodas	nurodomas modulio kodas;
mokslo šaka	nurodoma mokslo šaka
programa	nurodoma modulio programa
reg_nr	nurodomas modulio registracijos numeris
atestuotas_iki	nurodoma data iki kada modulis atestuotas
pavadinimas	nurodo modulio pavadinimą
pavadinimas_en	nurodo modulio pavadinimą anglų kalba
atsakingas_dest	nurodo už modulį atsakingo dėstytojo kodą
grupė	nurodo modulio grupę
butinas_pasirengimas	aprašo moduliui būtiną pasirengimą
anotacija	nurodo informaciją apie modulį (anotaciją)
anotacija_en	nurodo informaciją apie modulį anglų kalba
tiksiai	nurodo modulio tikslus
atsiskaitymai	nurodo atsiskaitymus už modulį

Klasė Studiju_moduliai_t aprašo studijų modulius (12 lentelė).

12 lentelė. Klasė Studiju_moduliai_t.

Atributai	Reikšmė
id	nurodomas modulio id
semestras	nurodo modulio semestrą
modulis	nurodo modulio kodą
programos_tipas	nurodo modulio programos tipą

Klasė Semestrai_t aprašo informaciją apie semestrus (13 lentelė).

13 lentelė. Klasė Semestrai_t.

Atributai	Reikšmė
id	nurodomas semestro id
programos_tipas	nurodo programos tipą
programa	nurodo studijų programą
nr	nurodo semestro numerį
specializacija	nurodo specializaciją
sritis	nurodo studijų programos sritį
metai	nurodo metus
laikas	nurodo pavasario ar rudens semestrą

Klasė Specializaciju_moduliai_t aprašo specializacijų modulius (14 lentelė).

14 lentelė. Klasė Specializaciju_moduliai_t.

Atributai	Reikšmė
id	nurodomas specializacijų modulio id
semestras	nurodo specializacijų modulio semestrą
modulis	nurodo modulio kodą
specializacija	nurodo specializaciją
fakultetas	nurodo fakultetą

Klasė Indv_studiju_planai_t aprašo studento individualius studijų planus (15 lentelė).

15 lentelė. Klasė Indv_studiju_planai_t.

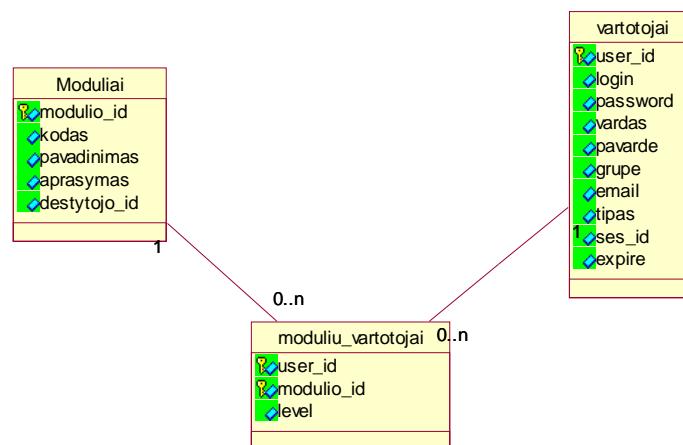
Atributai	Reikšmė
id	nurodomas individualaus studijų plano id
studentas	nurodo studento pažymėjimo numerį
semestras	nurodo semestrą
semestro	nurodo semestro numerį
metai	nurodo metus

Klasė Studentai_t aprašo informaciją apie semestrus (16 lentelė).

16 lentelė. Klasė Studentai_t.

Atributai	Reikšmė
Studento_id	nurodomas studento pažymėjimo numeris
fakultetas	nurodo fakultetą, kuriame mokosi studentas
stud_grupe	nurodo studento grupę
metai	nurodo studento įstojimo metus
vardas	nurodo studento vardą
pavarde	nurodo studento pavardę
gimimo_data	nurodo studento gimimo datą
laipsnis	nurodo studento laipsnį
programa	nurodomas studento programą
specializacija	nurodo studento specializaciją
adresas	nurodo studento adresą
miestas	nurodo miestą, iš kurio yra studentas
valstybe	nurodo valstybę, kurioje gimė studentas
telefonas	nurodo studento telefono numerį
email	nurodo studento elektroninio pašto adresą

Taip pat pateikta KTU kompiuterių katedros IS dalykinės srities duomenų klasės ir jų trumpas aprašymas, naudojamos šiame darbe (16 pav.).



16 pav. Dalykinės srities statinis modelis (KTU kompiuterių katedros IS)

Klasė Moduliai aprašo studijų modulius (17 lentelė).

17 lentelė. Klasė Moduliai.

Atributai	Reikšmė
modulio_id	nurodo modulio id

Kodas	nurodo modulio kodą
Pavadinimas	nurodo modulio pavadinimą
Aprašymas	nurodo modulio aprašymą
destrytojo_id	nurodo modulio destrytoją

Klasė modulio_vartotojai aprašo studijų modulių vartotojus (18 lentelė).

18 lentelė. Klasė modulio_vartotojai.

Atributai	Reikšmė
user_id	nurodo vartotojo id
modulio_id	nurodo modulio id
level	nurodo vartotojo lygį

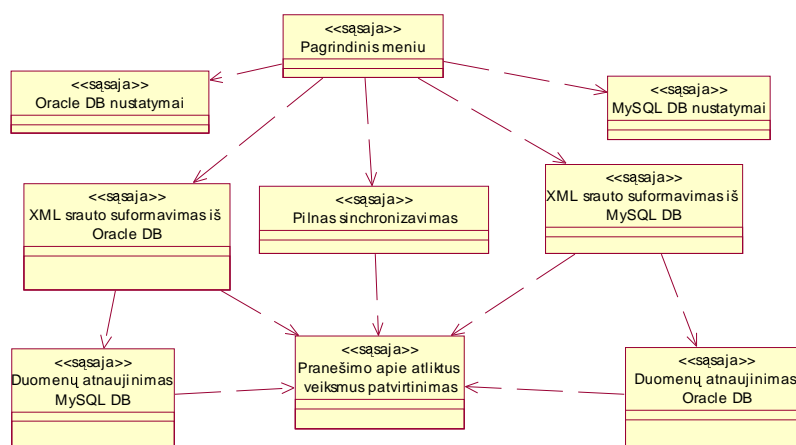
Klasė vartotojai aprašo sistemos vartotojus (19 lentelė).

19 lentelė. Klasė vartotojai.

Atributai	Reikšmė
user_id	nurodo vartotojo id
login	nurodo vartotojo prisijungimo vardą
password	nurodo prisijungimo prie sistemos slaptažodį
vardas	nurodo vartotojo vardą
pavarde	nurodo vartotojo pavardę
grupe	nurodo vartotojo grupę
email	nurodo vartotojo elektroninio pašto adresą
tipas	nurodo vartotojo tipą
ses_id	nurodo sesijos id
expire	nurodo sesijos pasibaigimo laiką

6.2.4. Vartotojo sąsajos modelis

Apibendrintas vartotojo sąsajos funkcionavimas yra parodytas žemiau esančiame statiniame modelyje (17pav.)



17 pav. Vartotojo sąsajos modelis

6.2.5. Papildomi reikalavimai sistemai

Funkcionalumui

Sistema turi apjungti tam tikrus duomenis tarp heterogeninių duomenų bazių, atliekant XML srautų formavimo procedūrą pagal nustatytą algoritmą, sudaryti sąlygas duomenų atnaujinimui norimu laiku.

Sistema turi dirbti ypač tiksliai. Formuojant duomenų srautus, bet kokios informacijos praradimas nėra leistinas, kadangi dirbama su informacija, kurios netekus atsiranda trikdžių dirbant su KTU akademinė ar KTU kompiuterių katedros IS.

Sistema turi integruoti tam tikrus duomenis iš KTU akademinės ir KTU kompiuterių katedros duomenų bazėse saugomų duomenų. Tarpiniai duomenų srautai yra atliekami ir saugomi panaudojus XML standartą.

Turi būti užtikrintas priėjimo prie duomenų saugumas. Suformuotus XML dokumentus gali peržiūrėti tik šios sistemos vartotojas, kadangi tarpiniai duomenys yra išsaugomi vartotojo nurodytoje vietoje. Kiti vartotojai, be šios sistemos naudotojo leidimo matyti duomenų negali.

Patikimumui

Sistema turi pilnai funkcionuoti. Apsisaugojimui nuo nenumatytų gedimų ir duomenų praradimų, integruojami duomenys turi būti saugomi kitame archyviname kataloge. Taip suformuojama duomenų atstatymo galimybė. Klaidingų duomenų įvedimo atvejais sistema privalo pateikti atitinkamą pranešimą ir duomenų į duomenų bazes neįrašyti, apsaugant sistemas kurios dirba su integruojamais duomenimis.

Patogumui

Sistemos išvaizdai specialių reikalavimų nėra, vartotojo sąsaja turi būti lengvai suprantama, pritaikyta įvairioms aplinkoms. Vartotojo programinei įrangai turi būti keliami minimalūs reikalavimai.

Efektyvumui

XML dokumentų formavimas, saugojimas ir persiuntimas turi užimti kuo mažiau laiko. Formuojant XML dokumentą negali būti sugaišta daugiau kaip 3 – 4 minutes. Visi taikomųjų programų uždaviniai gali būti vykdomi darbo stotyje (serveryje) ar net kliento kompiuteryje.

Perkeliamumui

Sistema turi turėti įdiegimo programą. Numatytas įdiegimas skirtingose operacinėse platformose.

Priežiūros savybės

Sistema turi dirbti stabiliai, įvykus gedimams turi būti numatyta duomenų atstatymo galimybė.

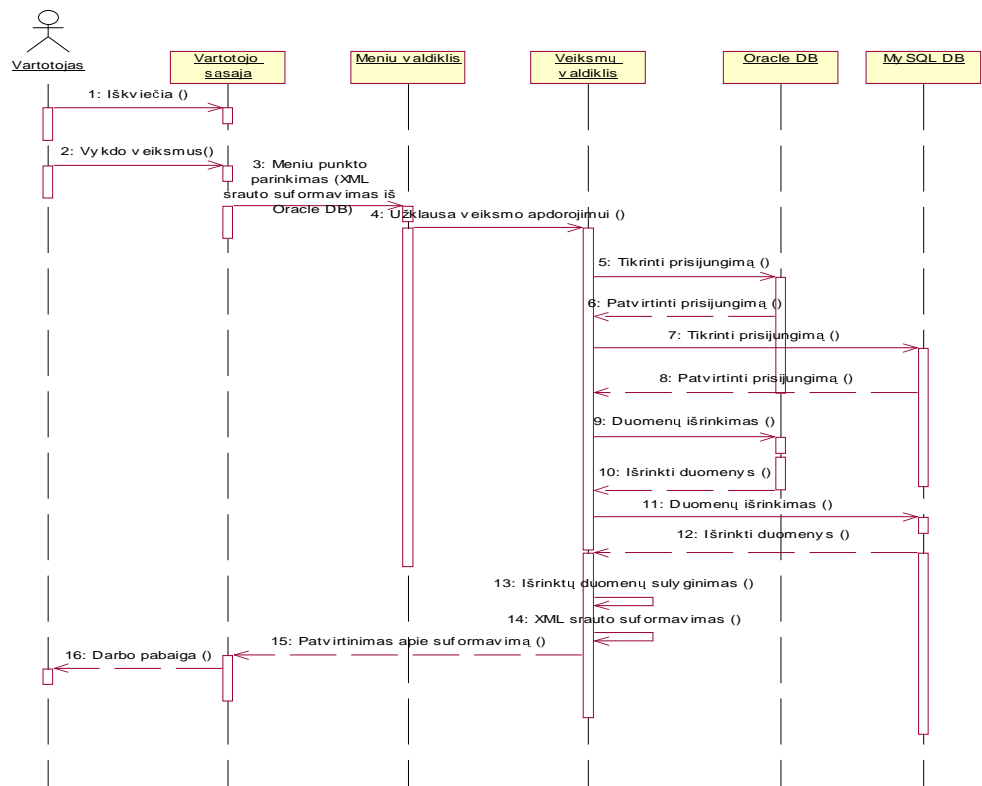
7. Sistemos struktūra ir realizacija

7.1. Duomenų integracijos panaudojimų atvejų ir sekų (arba bendradarbiavimo) diagramos

7.1.1. XML srauto sudarymo iš KTU akademinės DB veiklos procesų sekų modelis

Norėdami suformuoti XML srautą iš KTU akademinės duomenų bazės, vartotojas turintis tam teises, turi pasileisti sistemą ir pasirinkti reikimą meniu punktą operacijos atlikimui (18pav.).

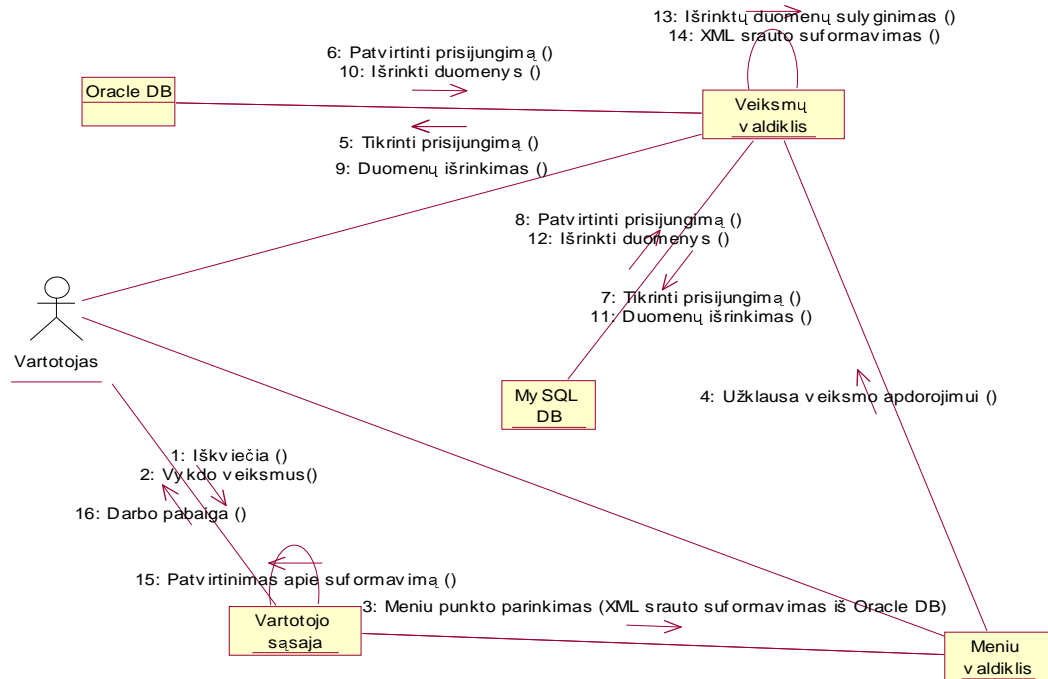
Sistema tikrina ar yra suformuotas XML srautas. Jei ne, tuomet sistema jungiasi prie KTU akademinės duomenų bazės ir užklausią reikiamų duomenų. Tada jungiamasi prie KTU kompiuterių katedros duomenų bazės ir užklausiama atitinkamų duomenų. Iš gautų duomenų formuojamas XML duomenų struktūra. Vartotojui ekrane išvedamas pranešimas apie pavykusį arba nepavykusį duomenų persiuntimą.



18 pav XML srauto sudarymo veiklos procesų sekų modelis (KTU akademinė DB)

7.1.2. XML srauto sudarymo iš KTU akademinės DB bendradarbiavimo modelis.

Bendradarbiavimo modelis (19pav.) atvaizduoja sąveiką vykdymo laiku (run-time) tarp elementų labai panašiais būdais, kaip ir veiklos procesų sekų modelyje (18pav.).

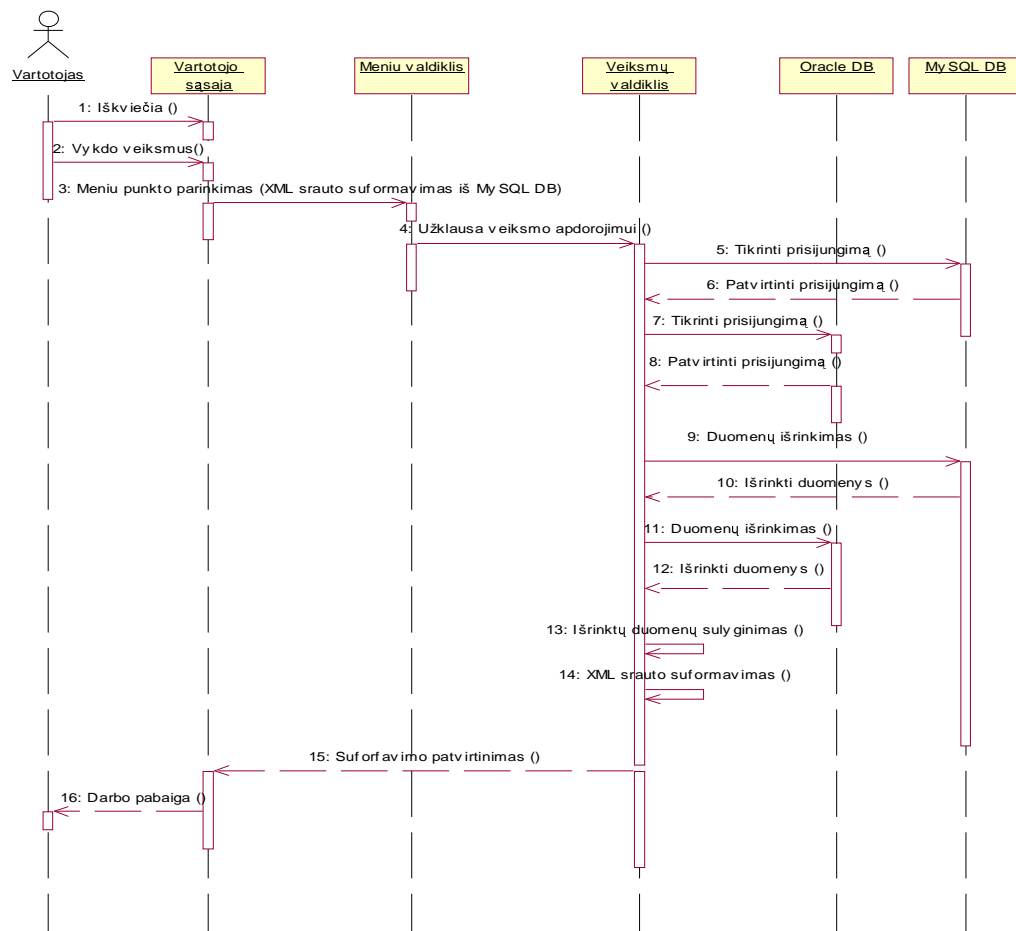


19 pav XML srauto sudarymo bendradarbiavimo modelis (KTU akademinė DB)

7.1.3. XML srauto sudarymo iš KTU kompiuterių katedros DB veiklos procesų sekų modelis

Norėdami suformuoti XML srautą iš KTU kompiuterių katedros duomenų bazės, galima tai atlikti pasirinkus reikiamą meniu punktą operacijos įvykdymui (20pav.).

Aktyvavus sistemą yra tikrinama ar suformuotas XML srautas. Jei ne, tuomet sistema jungiasi prie KTU kompiuterių katedros bazės ir užklausią reikiamų duomenų. Tada jungiamasi prie KTU akademinės duomenų bazės ir užklausiama atitinkamų duomenų. Iš gautų duomenų formuojamas XML duomenų struktūra. Vartotojui ekrane išvedamas pranešimas apie pavykusį arba nepavykusį duomenų persiuntimą.

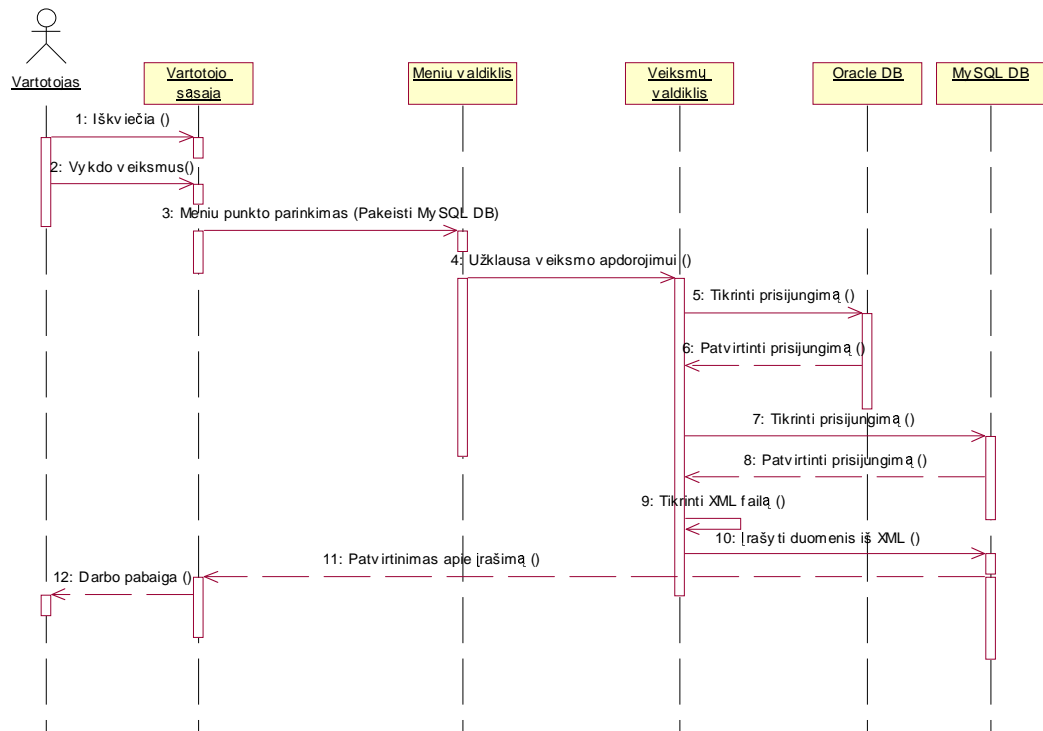


20 pav XML srauto sudarymo veiklos procesų sekų modelis (KTU KK)

7.1.4. Duomenų atnaujinimo KTU akademinės DB veiklos procesų sekų diagrama

Norėdami atnaujinti duomenis KTU akademinės duomenų bazės, galima tai atlikti pasirinkus reikimą meniu punktą operacijos įvykdymui (21 pav.).

Aktyvavus sistemą yra tikrinama ar suformuotas XML srautas. Jei ne, tuomet reikia suformuoti, sistemoje pasirinkus atitinkamą operaciją, kuri reikiamus duomenis patalpina į XML dokumentą. Jai XML duomenų struktūra yra suformuota sistema jungiasi prie KTU akademinės duomenų bazės ir iš XML dokumento įrašo naujus duomenis į DB. Vartotojui ekrane išvedamas pranešimas apie pavykusį arba nepavykusį duomenų atnaujinimą.



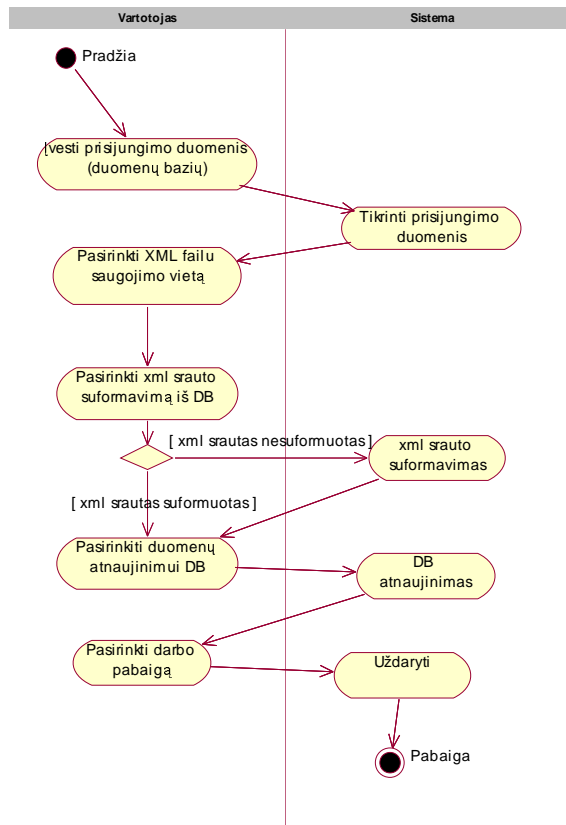
21 pav Individualaus dėstytojo plano peržiūros veiklos procesų sekų diagrama

Analogiška sekų diagrama gaunama duomenų atnaujinimui KTU kompiuterių katedros duomenų bazėj, kai tai atlieka vartotojas turintis tam teisę.

7.2. Sistemos elgsenos modelis

7.2.1. Sistemos elgsenos modelis , kai atnaujinama viena iš duomenų bazių

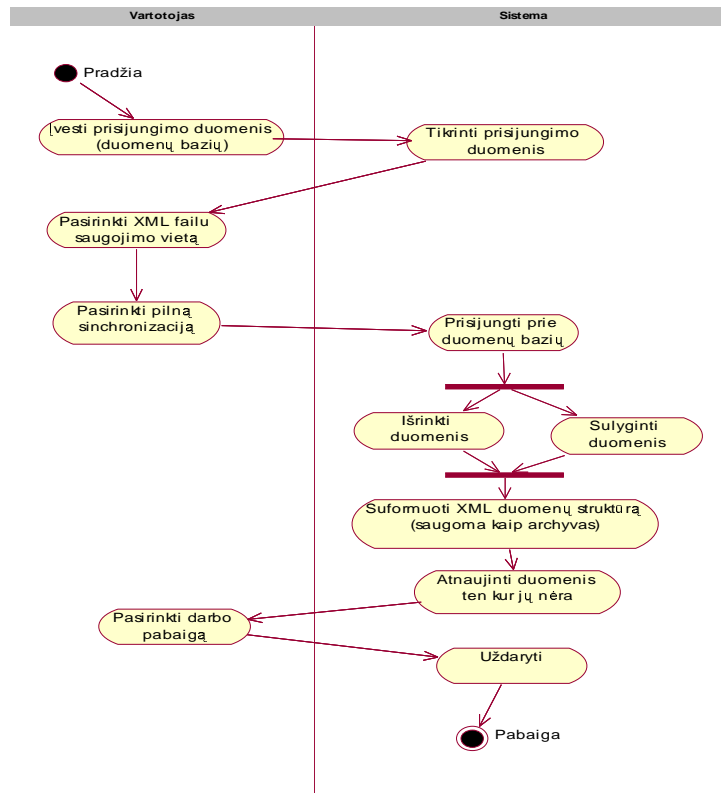
Žemiau pateiktas sistemos elgsenos modelis, kuriame atvaizduota, kaip atnaujinami vienos iš heterogeninių duomenų bazių duomenys, kurių elementai laikui einant keičia būsenas. Taip pat leistinus perėjimus ir jų sąlygas.



22 pav Sistemos elgsenos modelis, kai atnaujinama viena iš duomenų bazių

7.2.2. Sistemos elgsenos modelis , kai atliekama pilna sinchronizacija

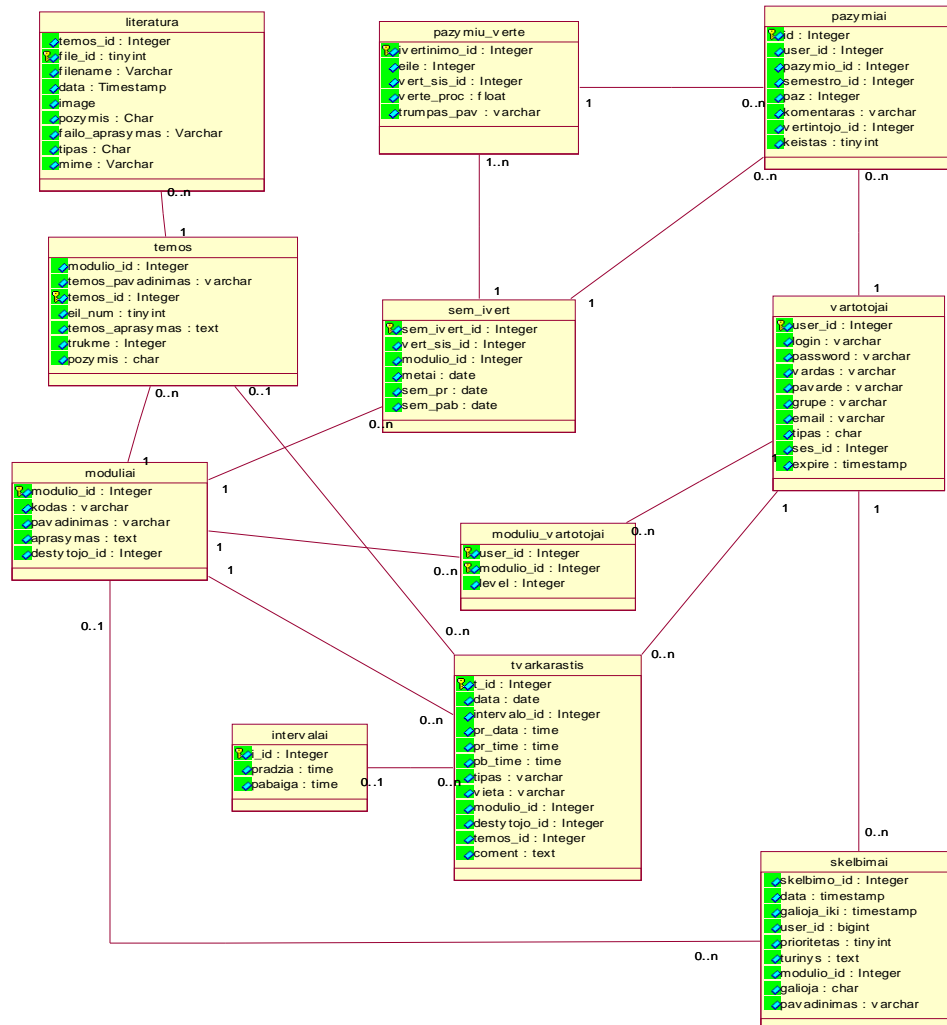
Pasirinkus pilną sinchronizaciją, vartotojui belieka patikslinti prisijungimo prie duomenų bazių parametrus ir sistema pati atlieka duomenų integraciją tarp heterogeninių duomenų bazių. Tuo pačiu yra išsaugomi tarpiniai integracijos duomenys į atitinkamą katalogą XML dokumentu, kad užtikrinti duomenų atstatymo galimybę įvykus nenumatytiems atvejams (23pav.).



23 pav Sistemos elgsenos diagrama, kai atliekama pilna sinchronizacija tarp heterogeninių duomenų bazių

7.3. Duomenų bazių modelis

KTU kompiuterių katedros duomenų bazės fizinis modelis (MySQL):



24 pav. KTU KK duomenų bazės fizinis modelis.

Duomenų bazę sudaro tokios lentelės:

- moduliai;
- tvarkarastis;
- intervalai;
- skelbimai;
- temos;
- literatūra;
- pazymiu_verte;
- pazymiai;

- sem_ivert;
- vartotojai;
- moduliu_vartotojai

Lentelė *moduliai*:

Šioje lentelėje saugoma konkretų modulį apibūdinanti informacija.

20 lentelė. DB lentelė *moduliai*

Laukas	Aprašymas	Tipas
modulio_id (PK)	Modulio identifikacijos numeris	Sveikasis skaičius
kodas	Modulio kodas	Tekstinė eilutė iki 20 simbolių
pavadinimas	Modulio pavadinimas	Tekstinė eilutė iki 100 simbolių
aprasymas	Trumpas modulio aprašymas	Tekstinė eilutė iki 255 simbolių
destytojo_id	Dėstytojo identifikacijos numeris	Sveikasis skaičius

Lentelė *tvarkarastis*:

Šioje lentelėje saugoma tvarkaraščio organizavimui skirta informacija.

21 lentelė. DB lentelė *tvarkarastis*

Laukas	Aprašymas	Tipas
t_id (PK)	Tvarkaraščio identifikacijos numeris	Sveikasis skaičius
data	Įvykio data	Datos formatas
intervalo_id	Intervalo identifikacijos numeris	Sveikasis skaičius
modulio_id	Modulio identifikacijos numeris	Sveikasis skaičius
pr_data	Įvykio pradžia	Laiko žymė
pb_time	Įvykio pabaiga	Laiko žymė
tipas	Užsiėmimo pobūdis (paskaita, laboratoriniai, konsultacijos ir kt.)	Tekstinė eilutė iki 10 simbolių
vieta	Užsiėmimui skirta vieta	Tekstinė eilutė iki 100 simbolių
destytojo_id	Dėstytojo identifikacijos numeris	Sveikasis skaičius
temos_id	Temos identifikacijos numeris	Sveikasis skaičius
coment	Įvykio aprašymas	Tekstinė eilutė

Lentelė *skelbimai*:

Šioje lentelėje saugoma skelbimų organizavimui skirta informacija.

22 lentelė. DB lentelė *skelbimai*

Laukas	Aprašymas	Tipas
skelbimo_id (PK)	Skelbimo identifikacijos numeris	Sveikasis skaičius
modulio_id	Modulio identifikacijos numeris	Sveikasis skaičius
data	Skelbimo idėjimo data	Laiko žymė
galioja_iki	Skelbimo galiojimo laikas	Laiko žymė
user_id	Skelbimo autorius	Sveikasis skaičius
prioritetas	Skelbimo prioritetas (skalė nuo labai svarbu iki nesvarbu)	Sveikasis skaičius
turinys	Skelbimo turinys	Tekstinė eilutė
pavadinimas	Skelbimo pavadinimas	Tekstinė eilutė 100 simbolių

Lentelė *temos*:

Šioje lentelėje saugoma modulio paskaitų temų informacija.

23 lentelė. DB lentelė *temos*

Laukas	Aprašymas	Tipas
temos_id (PK)	Paskaitos temos identifikacijos numeris	Sveikasis skaičius
modulio_id (FK)	Modulio identifikacijos numeris	Sveikasis skaičius
temos_pavadinimas	Temos pavadinimas	Tekstinė eilutė iki 100 simbolių
eil_num	Eilės numeris	Sveikasis skaičius
temos_aprasymas	Temos apibūdinimas	Tekstinė eilutė
trukme	Valandų kiekis skirtas šiai temai	Sveikasis skaičius
pozymis	Požymis paskaitos tipui (laboratorinis, teorine paskaita, pratybos, kita)	Tekstinė eilutė vieno simbolio

Lentelė *literatura*:

Šioje lentelėje saugoma su kiekviena tema susijusi literatūra.

24 lentelė. DB lentelė *literatura*

Laukas	Aprašymas	Tipas
file_id (PK)	Failo identifikacijos numeris	Sveikasis skaičius
temos_id	Paskaitos temos identifikacijos numeris	Sveikasis skaičius
Filename	Failo vardas	Tekstinė eilutė iki 100 simbolių
Data	Įdėjimo laikas	Laiko žymė
Failas	Konkretus failas	Blob tipas (maksimalu dydį nustato duomenų bazės administratorius)
pozymis	Literatūros tipas	Tekstinė eilutė iki 3 simbolių (failas, URL nuoroda, HTML dokumentas, kita)
failo_aprasymas	Konkreto failo trumpas aprašymas	Tekstinė eilutė iki 255 simbolių

Lentelė *pazymiu_verte*:

Šioje lentelėje saugoma konkrečios vertinimo sistemos sudėtis, t.y. vertinimo kriterijai ir jų procentinė vertė.

25 lentelė. DB lentelė *pazymiu_verte*

Laukas	Aprašymas	Tipas
ivertinimo_id (PK)	Įvertinimo identifikacijos numeris	Sveikasis skaičius
vert_sist_id	Vertinimo sistemos identifikacijos numeris	Sveikasis skaičius
vertė_proc	Pažymio vertė procentais (kiekvieno atsiskaitymo ar laboratorinio darbo)	Slankaus kablelio skaičius
trumpas_pav	Vertinimo kriterijaus pavadinimas	Tekstinė eilutė iki 255 simbolių
eile	Įvertinimų eiliškumas (įvertinimas su nuline eile yra galutinis)	Sveikasis skaičius

Lentelė *pazymiai*:

Šioje lentelėje saugomi konkretūs pažymiai už konkretų kriterijų konkrečiam vartotojui bei komentaras prie pažymio.

26 lentelė. DB lentelė *pazymiai*

Laukas	Aprašymas	Tipas
user_id (PK)	Vartotojo identifikacijos numeris	Sveikasis skaičius
pazymio_id (PK)	Pažymio identifikacijos numeris	Sveikasis skaičius
semestro_id (PK)	Semestro identifikacijos numeris	Sveikasis skaičius
paz	Galutinis pažymys	Sveikasis skaičius
komentaras	Komentaras prie pažymio	Tekstinė eilutė iki 100 simbolių
vertintojo_id	Vertintojo identifikacijos numeris	Sveikasis skaičius

Lentelė *sem_ivert*:

Šioje lentelėje saugomi semestrų įvertinimo sistemos duomenys.

27 lentelė. DB lentelė *sem_ivert*

Laukas	Aprašymas	Tipas
sem_ivert_id (PK)	Semestro įvertinimo identifikacijos numeris	Sveikasis skaičius
vert_sist_id	Vertinimo sistemos identifikacijos numeris	Sveikasis skaičius
modulio_id	Modulio identifikacijos numeris	Sveikasis skaičius
sem_pr	Semestro pradžia	Datos formatas
sem_pab	Semestro pabaiga	Datos formatas

Lentelė *moduliu_vartotojai*:

Šioje lentelėje saugoma informacija apie vartotojo priklausymą vienam ar kitam moduliu bei vartotojo teisių lygis tame modulyje, kuris apsprendžia jų teises atliekant veiksmus su modulių informacija.

28 lentelė. DB lentelė *moduliu_vartotojai*

Laukas	Aprašymas	Tipas
user_id (PK)	Vartotojo identifikacijos numeris	Sveikasis skaičius
modulio_id (PK)	Modulio identifikacijos numeris	Sveikasis skaičius
level	Vartotojo lygis	Sveikasis skaičius

Lentelė *intervalai*:

Šioje lentelėje saugomi tvarkaraščio užsiėmimų intervalai.

29 Lentelė. DB lentelė *intervalai*

Laukas	Aprašymas	Tipas
i_id (PK)	Intervalo identifikacijos numeris	Sveikasis skaičius
pradžia	Įvykio pradžia	Laiko žymė
pabaiga	Įvykio pabaiga	Laiko žymė

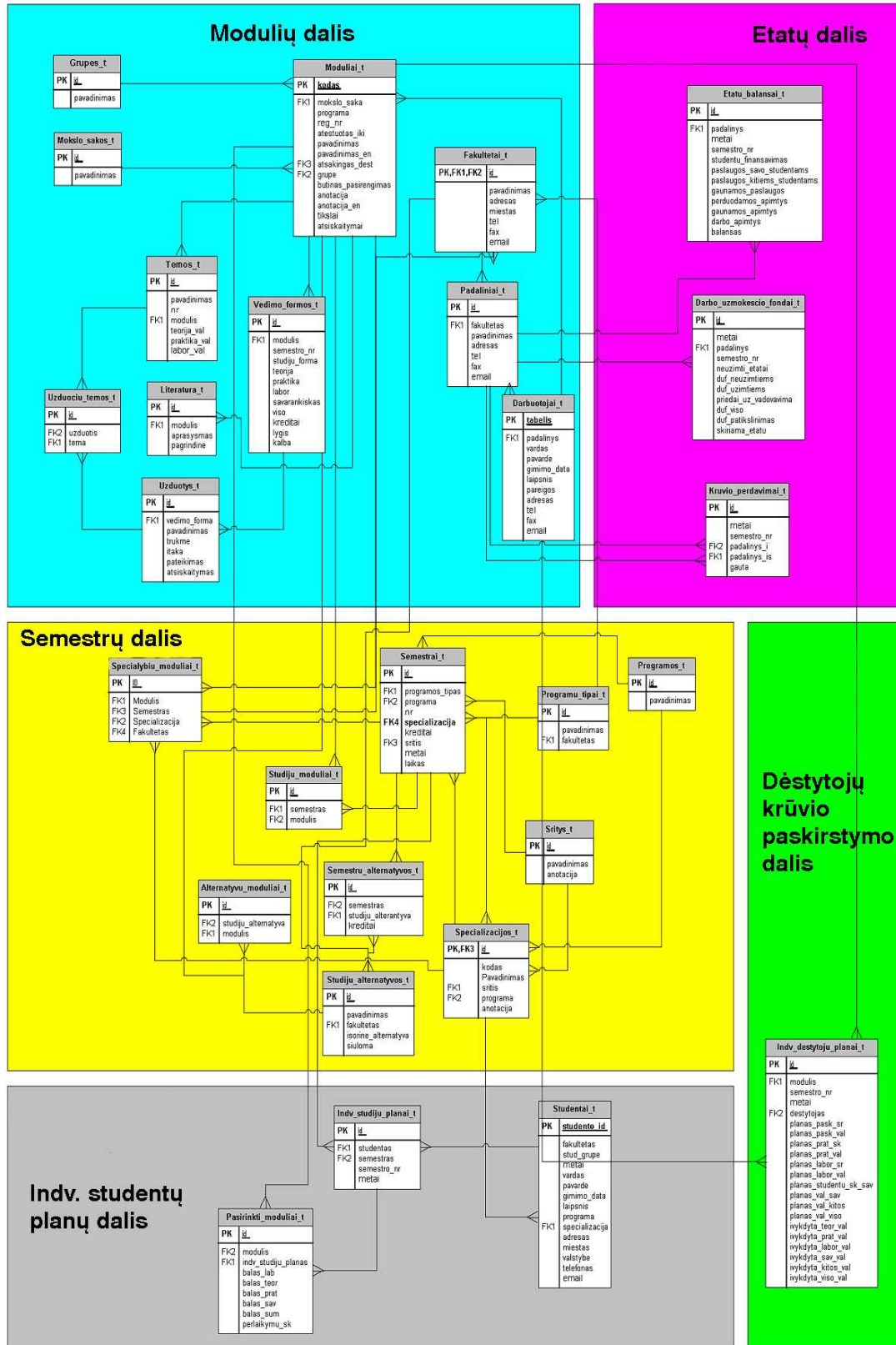
Lentelė *virtotojai*:

Šioje lentelėje saugoma sistemos vartotojų informacija.

30 lentelė. DB lentelė *virtotojai*

Laukas	Aprašymas	Tipas
user_id (PK)	Vartotojo identifikacijos numeris	Sveikasis skaičius
login	Vartotojo prisijungimo vardas	Tekstinė eilutė iki 50 simbolių
password	Prisijungimo prie sistemos slaptažodis	Tekstinė eilutė iki 50 simbolių
vardas	Vartotojo vardas	Tekstinė eilutė iki 50 simbolių
pavarde	Vartotojo pavardė	Tekstinė eilutė iki 50 simbolių
grupe	Vartotojo grupė	Tekstinė eilutė iki 10 simbolių
email	Vartotojo elektroninio pašto adresas	Tekstinė eilutė iki 255 simbolių
tipas	Vartotojo tipas (administratorius, dėstytojas, studentas)	A (Administratorius), D (Dėstytojas) ar S (Studentas)
ses_id	Sesijos identifikacijos numeris	Sveikasis skaičius
expire	Sesijos pasibaigimo laikas	Laiko žymė
asm_kod	Vartotojo asmens kodas (dėstytojų – tabelis; studentų – pažymėjimo numeris)	Tekstinė eilutė iki 11 simbolių

KTU akademinės duomenų bazės fizinis modelis (Oracle):



23 pav. KTU akademinės duomenų bazės modelis

Duomenų bazę sudaro tokios lentelės (aprašytos svarbiausios, naudojamos darbe):

- Moduliai_t;
- Darbuotojai_t;
- Studentai_t;

Lentelė *Moduliai_t*:

Šioje lentelėje saugoma informacija apie dėstomus modulius, pvz.:

Informacinių sistemų modulis.

31 lentelė. DB lentelė *Moduliai_t*

Laukas	Aprašymas	Tipas
kodas	modulio kodas	Tekstinė eilutė
mokslo_saka	modulio mokslo šaka	Tekstinė eilutė
programa	modulio programos numeris	Tekstinė eilutė
reg_nr	modulio registracijos numeris	Tekstinė eilutė
atestuotas_iki	data iki kada modulis atestuotas	Datos formatas
pavadinimas	modulio pavadinimas	Tekstinė eilutė iki 60 simbolių
pavadinimas_en	modulio pavadinimas anglų kalba	Tekstinė eilutė iki 60 simbolių
atsakingas_dest	už modulį atsakingo dėstytojo kodas	Tekstinė eilutė iki 10 simbolių
grupe	modulio grupė	Tekstinė eilutė
butinas_pasirengimas	moduliui būtinas pasirengimas	Tekstinė eilutė iki 120 simbolių
anotacija	informaciją apie modulį (anotaciją)	Tekstinė eilutė iki 480 simbolių
anotacija_en	informaciją apie modulį anglų kalba	Tekstinė eilutė iki 480 simbolių
tikslai	informacija apie modulio tikslus	Tekstinė eilutė iki 420 simbolių
atsiskaitymai	informacija apie atsiskaitymus už modulį	Tekstinė eilutė iki 240 simbolių

Lentelė *Darbuotojai_t*:

Šioje lentelėje saugoma informacija apie KTU darbuotojus (pvz.: dėstytojus)

32 lentelė. DB lentelė *Darbuotojai_t*

Laukas	Aprašymas	Tipas
tabelis	darbuotojo id	Tekstinė eilutė iki 10 simbolių
padalinys	padalinys, kuriame darbuotojas dirba	Sveikasis skaičius
vardas	darbuotojo vardas	Tekstinė eilutė iki 20 simbolių
pavarde	darbuotojo pavardė	Tekstinė eilutė iki 30 simbolių
gimimo_data	darbuotojo gimimo data	Datos formatas
laipsnis	žodžiu parašytas darbuotojo laipsnis	Tekstinė eilutė iki 50 simbolių
pareigos	darbuotojo pareigos. pvz., vedėjas	Tekstinė eilutė iki 50 simbolių
adresas	darbuotojo adresas	Tekstinė eilutė iki 50 simbolių
tel	darbuotojo telefono numeris	Tekstinė eilutė iki 15 simbolių
fax	darbuotojo fakso numeris	Tekstinė eilutė iki 15 simbolių
email	darbuotojo elektroninio pašto adresas	Tekstinė eilutė iki 15 simbolių
pass	darbuotojo slaptažodis prisijungti prie DB	Tekstinė eilutė iki 30 simbolių

Lentelė *Studentai_t*:

Šioje lentelėje saugoma informacija apie KTU studentus

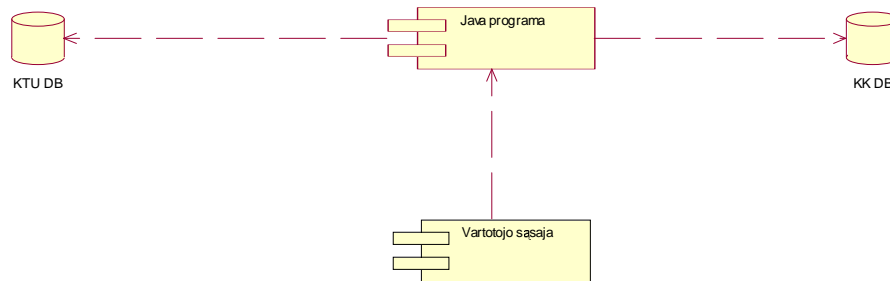
33 lentelė. DB lentelė *Studentai_t*

Laukas	Aprašymas	Tipas
Studento_id	nurodomas studento pažymėjimo numeris	Tekstinė eilutė
fakultetas	nurodo fakultetą, kuriame mokosi studentas	Tekstinė eilutė
Stud_grupe	nurodo studento grupę	Tekstinė eilutė iki 10 simbolių
metai	nurodo studento įstojimo metus	Sveikasis skaičius
Vardas	nurodo studento vardą	Tekstinė eilutė iki 20 simbolių
pavarde	nurodo studento pavardę	Tekstinė eilutė iki 30 simbolių
Gimimo_data	nurodo studento gimimo datą	Datos formatas
laipsnis	nurodo studento laipsnį	Tekstinė eilutė iki 20 simbolių
programa	nurodomas studento programą	Tekstinė eilutė
specializacija	nurodo studento specializaciją	Tekstinė eilutė
adresas	nurodo studento adresą	Tekstinė eilutė iki 50 simbolių
miestas	nurodo miestą, iš kurio yra studentas	Tekstinė eilutė iki 20 simbolių
valstybe	nurodo valstybę, kurioje gimė studentas	Tekstinė eilutė iki 20 simbolių
telefonas	nurodo studento telefono numerį	Tekstinė eilutė iki 15 simbolių
email	nurodo studento elektroninio pašto adresą	Tekstinė eilutė iki 40 simbolių
password	nurodo studento prisijungimo slaptažodį prie sistemos	Tekstinė eilutė iki 20 simbolių
kursas	nurodo studento kursą	Sveikasis skaičius

8. Realizacijos modelis

8.1. Komponentų diagrama

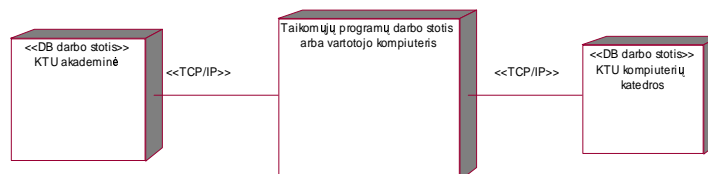
Sistemą sudaro šie pagrindiniai komponentai: heterogeninės duomenų bazės, tarp kurių atliekama tam tikros informacijos duomenų integracija; taikomoji programa, kuri paleidžiama darbo stotyje arba vartotojo pusėje ir vartotojo sąsaja skirta darbui su sistema (24pav.).



24pav. Komponentų diagrama

8.2. Paskirstymo diagrama

Vartotojas gali pasileisti sistemą ir savo kompiuteryje, tačiau tam, kad neapkrautų papildomais resursais, sistema gali būti vykdoma ir darbo stotyje. JAVA technologija valdo jungimąsi prie MySQL ir ORACLE duomenų bazių, kuriuose informacija gali būti atnaujiniama bet koku metu (26pav.).



25pav. Paskirstymo diagrama

8.3. Testavimo modelis

Testavimo metu tikrinama kaip funkcionuoja sukurta informacinė sistema, ar teisingai atliekamos reikalavimuose apibrėžtos funkcijos:

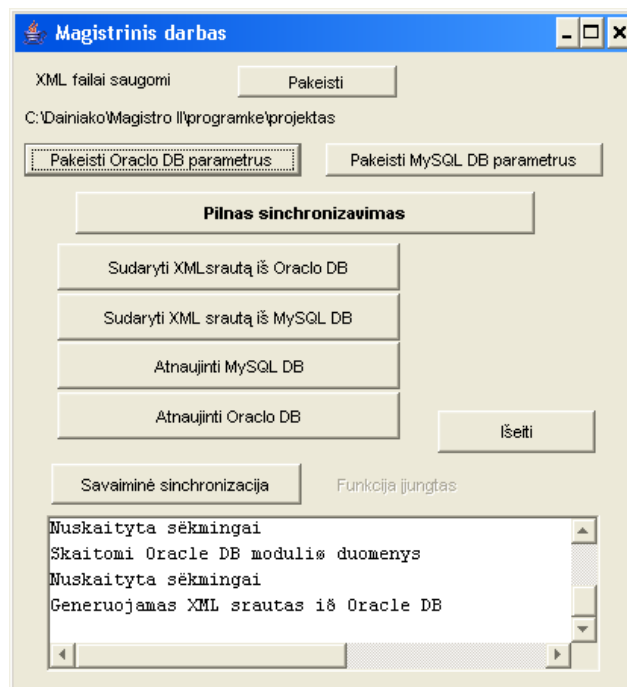
- Prisijungimo prie heterogeninių duomenų bazių korektiškumas.
- Korektiška duomenų integracija, duomenų persiuntimas.
- XML duomenų struktūrų formavimo laikas (turi atitikti reikalavimus).
- Archyvinių duomenų kaupimo korektiškumas, suteikiantis duomenų atstatymo galimybę.

8.4. Reikalavimai sistemos funkcionavimo palaikymui.

Sistamai paleisti reikalinga techninė įranga – kompiuteris. Galimos kompiuterio operacinės sistemos Windows, Linux, Solaris ir Mac šeimos operacinės sistemos. Rekomenduojami kompiuterio parametrai: procesoriaus greitis $\geq 500\text{Mhz}$, RAM $\geq 256\text{MB}$, laisvos vietos kietame diske – 5GB, 10/100 MB Ethernet adapteris.

8.5. Sistemos panaudojimo atvejis

Duomenų integracijai atlikti yra sukurta programinė įranga, kuri apdoroja XML formato duomenis ir juos išsaugo reliacinėje duomenų bazėje. Sistemos vartotojo sąsajos forma pateikiama 26 paveiksle :



26 pav. Sistemos vartotojo sąsaja

Vartotojas savo nuožiūra gali sudaryti XML srautą ir atnaujinti jį dominančią duomenų bazę. Pasirinkus pilną sinchronizaciją, sistema atlieka duomenų integraciją abiejų heterogeninių duomenų bazių ir sudaro XML duomenų struktūrą išsaugant tarpinius duomenis, suteikdama atstatymo galimybę praradus duomenis. Sistema turi ir savaiminės sinchronizacijos funkciją, kurios pagalba nustatomas laiko intervalas duomenų automatizuotam integravimui, t.y. pagal vartotojo nustatymus (data ir laikas nuo kada bus pradėtas vykdyti šis procesas, bei pasikartojimo intervalas) sistema pati atliks pilną duomenų bazių sinchronizavimą. Duomenims persiųsti naudojami XML duomenys ir operuojama tik tam tikrais duomenimis.

Darbo išvados

- Darbe panaudotas analizės metodas, taikant UML priemonę, leidžiančią pilnai aprašyti kuriamos sistemos probleminę sritį, reikalavimus bei realizacijos procesą.
- Išnagrinėta XML programavimo kalbos specifiška, pagrindiniai jos sandaros komponentai, aptartas panaudojimas, išvardinti privalumai ir trūkumai. Yra pateikiamas XML dokumento korektiškas pavyzdys.
- Išnagrinėta heterogeninių duomenų bazių integracija ir tarpininkavimo programų taikymas.
- Atlikus heterogeninių duomenų bazių integraciją nustatyta, kad visų pirma reikia atsižvelgti į vardų konfliktus. Norint išvengti šios problemos yra sukuriamas globalus žodynas, kuriame vienodos sąvokos atitinka tą patį objektą. Tai panaudojus sukurta studijų modulių planavimo ir valdymo sistema.
- Realizuota heterogeninių duomenų bazių integracija panaudojant XML duomenų struktūrą.
- Darbe nagrinėjamos transformacijos ir heterogeninių duomenų bazių integracijos metodas buvo praktiškai pritaikytas KTU kompiuterių katedros veikloje

Naudota literatūra

[1] An exploration of XML in database management systems [žiūrėta 2004-06-03]. Prieiga per Internetą:

<http://www.topxml.com/xml/articles/xmlldb/default.asp>

[2] IBM Framework for e-business Technology, Solution and Design Overview. IBM Redbook [žiūrėta 2003-11-20]. Prieiga per Internetą: <http://ibm.com/redbooks>

[3] Igor Dayen. Storing XML in Relational Databases [žiūrėta 2004-12-01]. Prieiga per Internetą: <http://www.xml.com/pub/a/2001/06/20/databases.html>

[4] Introduction to XML Schema [žiūrėta 2003-12-03]. Prieiga per Internetą: http://www.w3schools.com/schema/schema_intro.asp

[5] J2EE Integration (JInt) Demo [žiūrėta 2003-11-25]. Prieiga per Internetą: http://otn.oracle.com/sample_code/tech/java/j2ee/jintdemo/index.html

[6] Kontrimas V. Java sąsaja su reliacinėmis duomenų bazėmis // Informacinės technologijos 2003 : konferencijos pranešimų medžiaga. Kaunas : Technologija, 2003. p. III-31 - III-34.

[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] Ronald Bourret, XML Database Products [žiūrėta 2004-10-16]. Prieiga per internetą: <http://www.rpbourret.com/xml/XMLDatabaseProds.htm>

[9] Using XML in Oracle Database Applications [žiūrėta 2004-06-05]. Prieiga per Internetą: http://otn.oracle.com/tech/xml/info/htdocs/otnwp/about_oracle_xml_products.htm

[10] W3C®. W3C Technical Reports and Publications [žiūrėta 2004-01-28]. Prieiga per Internetą: <http://www.w3.org/TR/>

[11] XML Tutorial [žiūrėta 2003-12-03]. Prieiga per Internetą:

<http://www.zvon.org/xxl/XMLTutorial/General/book.html>

Priedai

Priedas Nr 1. Duomenų integravimas

DUOMENŲ INTEGRAVIMAS

Dainius Jurčikonis, Aurimas Paršonis, Egidijus Kazanavičius

Kauno Technologijos Universitetas, Kompiuterių katedra

Studentų g. 50-213, LT-3031 Kaunas

Straipsnyje 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 keisti 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 keisti 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 nepriklaustytų nuo atitinkamos sistemos ar duomenų bazės. Šiuo atveju patogiausia naudoti XML (EXtensible Markup Language) [7].

XML technologija

- XML skirta keitimuisi duomenimis;
- XML skirta keisti 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 patogu 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 schemą, 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 sąlygas 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.

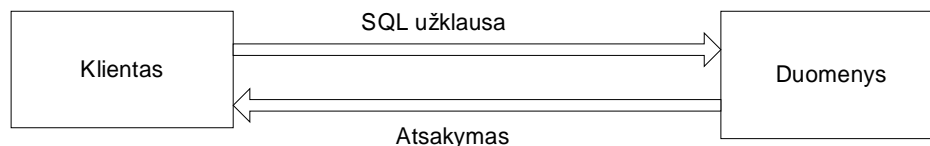
Tarpininkavimo programos - bet kokio tipo programinė įranga, kuri palengvina komunikaciją tarp dviejų ar daugiau programų sistemų. TP turi mechanizmą, kuris vienai esybei (taikomajai programai ar duomenų bazei) leidžia komunikuoti su kita esybe ar esybėmis, leidžia paslėpti pirminio šaltinio ir adresato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

		(Shareware)			
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.rpbouret.com/xml/XMLDatabaseProds.htm>

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

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.

Priedas Nr 2.Eksperimento XML dokumento fragmentas:

```
<?xml version="1.0" encoding="windows-1257" ?>
= <ExportMySQL>
= <studentas>
  <AsmKod />
  <Pavarde>t120b015</Pavarde>
  <Vardas>t120b015</Vardas>
  <Grupe>IFC-2</Grupe>
  </studentas>
= <studentas>
  <AsmKod />
  <Pavarde>T120B008</Pavarde>
  <Vardas>T120B008</Vardas>
  <Grupe>IF</Grupe>
  </studentas>
= <studentas>
  <AsmKod />
  <Pavarde>Armonas</Pavarde>
  <Vardas>Andrius</Vardas>
  <Grupe>IFM-9/1</Grupe>
  </studentas>
= <modulis>
= <Kodas>T120B016</Kodas>
  <Pavadinimas>Projektavimas ir inovacijos</Pavadinimas>
  <Anotacija>Projekto inicijāvimas.Projekto koncepcijos analiz?.Kainos ir
    investicijø analiz?.Projektø vykdymas,pristatymas ir
    dokumentacija.Komandinis darbas.Inovacijos.Produktø vystymo
    procesai.Inovacijø valdymas ir vadovai.</Anotacija>
= <Destytojas>
  <Tabelis />
  <Email>pranas@ifko.ktu.lt</Email>
  <Pavarde>Kanapeckas</Pavarde>
  <Vardas>Pranas</Vardas>
  </Destytojas>
  </modulis>
</ExportMySQL>
```