

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Aivaras Sakalas

Konferencijos informacinės sistemos prototipas

Magistro darbas

Darbo vadovas
doc. Lina Nemuraitė

Kaunas, 2005

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

TVIRTINU
Katedros vedėjas
doc. dr. R. Butleris

Konferencijos informacinės sistemos prototipas

Magistro darbas

Kalbos konsultantė
Lietuvių kalbos katedros lektorė
dr. J. Mikėlionienė

Vadovas
doc. L. Nemuraitė

Recenzentas
doc. dr. V. Pilkauskas

Atliko
IFM-9/2 gr. stud.
A. Sakalas

Kaunas, 2005

Turinys

1. ĮVADAS	8
2. INTERNETO SISTEMŲ KŪRIMO TECHNOLOGIJŲ IR METODŲ ANALIZĖ	12
2.1 Konferencijų programinės įrangos poreikių analizė	12
2.2 Java karkasų analizė	12
2.2.1 Java Struts	13
2.2.2 Java Spring MVC	14
2.2.3 Java WebWork	14
2.2.4 Java Tapestry	14
2.2.5 Java JSF	14
2.3 Java technologijų palyginimo rezultatai	15
2.4 Analizės išvados	16
3. KARKASAIŠ GRINDŽIAMA KŪRIMO METODIKA	18
3.1 Karkasų sąvokos	18
3.2 Java karkasinės technologijos	18
3.2.1 Java servletų technologija	18
3.2.2 JSP technologija	19
3.2.3 EJB technologija	19
3.3 Java Tapestry karkasas	20
3.4 Kūrimo metodika naudojant Tapestry karkasą	22
3.4.1 Java Tapestry MVC šablonas	22
3.4.2 Java Tapestry karkaso architektūra	23
3.4.3 Java Tapestry karkaso plėtimas	26
3.4.4 Spindle 3.0 priedas Eclipse programavimo aplinkai	29
4. METODIKOS TAIKYMAS KONFERENCIJŲ PROGRAMINĖS ĮRANGOS PROTOTIPUI SUKURTI	31
4.1 Konferencijų sistemos panaudojimo atvejai	32
4.1.1 Panaudojimo atvejų vaizdas	32
4.1.2 Panaudojimo atvejų aprašymas	32
4.2 Sistemos paketų struktūra	39
4.3 Konferencijų sistemoje vykdomi procesai	42
4.3.1 Naujos konferencijos skelbimas	42
4.3.2 Konferencijos užbaigimo procesas	43
4.3.3 Straipsnio įtraukimo į konferenciją procesas	45
4.3.4 Autorių ir recenzentų įtraukimo į konferenciją procesas	46
4.4 Klasių detalizavimas	48
4.4.1 Vartotojų paketas	49
4.4.2 Konferencijų paketas	51
4.4.3 Prisijungimo paketas	54
4.5 Duomenų vaizdas	55
4.6 Komponentų ir diegimo diagramos	56

5. SUKURTO PRODUKTO VEIKIMO, KOKYBĖS IR TAIKymo GALIMYBIŲ ANALIZĖ	59
5.1 Vartotojų darbas su sistema	59
5.2 Kokybės analizė	67
5.2.1 Kokybės tikrinimas pagal funkcionalumą	68
5.2.2 Kokybės analizės rezultatai	71
5.3 Taikymo galimybių analizė	72
6. IŠVADOS	74
7. LITERATŪRA	75

Paveikslų tūrinys

1.1 pav. Java karkasų lyginimo rezultatai.....	16
1.2 pav. Java ServerPages architektūra.....	19
1.3 pav. Java ServerPage su EJB architektūra. JavaBeans naudojimas JSP puslapiuose.....	20
1.4 pav. „Java Tapestry Framework“ komponentinio skaidymo pavyzdys.....	21
1.5 pav. HTML kodo dalis, kurioje naudojami Tapestry komponentai.	22
1.6 pav. MVC šablonas atskiria vaizdavimą nuo veiklos procesų.....	23
1.7 pav. Tapestry karkasų sistemos komponentų sąveika.....	24
1.8 pav. Detali Tapestry karkasų sistemos komponentų sąveika.....	24
1.8 pav. Tapestry karkasų sistemos klasių architektūra.....	26
1.9 pav. Tapestry karkasų sistemos plėtimas.....	27
1.10 pav. Tapestry karkaso plėtimo veiksmų seka.....	28
1.11 pav. XML konfigūracinio failo fragmentas.....	29
1.12 pav. Eclipse programavimo aplinka.....	30
1.9 pav. Panaudojimo atvejų vaizdas.....	32
1.10 pav. Sistemos paketų sąveikos diagrama.....	39
1.11 pav. Sistemos verslo klasių paketas.....	40
1.12 pav. Vartotojo paslaugų paketas pagal „Tapestry“ karkasą.....	41
1.13 pav. Naujos konferencijos paskelbimo procesas.....	42
1.14 pav. Naujos konferencijos paskelbimo veiksmų sekos diagrama.....	43
1.15 pav. Konferencijos užbaigimo procesas	44
1.16 pav. Konferencijos užbaigimo veiksmų sekos diagrama.....	44
1.17 pav. Straipsnio įkėlimo į konferenciją procesas.....	45
1.18 pav. Straipsnio įkėlimo į konferenciją veiksmų sekos diagrama.....	46
1.19 pav. Dalyvių įtraukimo į konferenciją procesas.....	47
1.20 pav. Dalyvių įtraukimo į konferenciją veiksmų sekos diagrama.....	48
1.21 pav. Verslo klasių paketas.....	48
1.22 pav. Paketo „Vartotojas“ struktūra.....	49
1.23 pav. Paketo „Konferencijos“ struktūra.....	51
1.24 pav. Paketo „Prisijungimas“ struktūra.....	54
1.25 pav. Duomenų modelis.....	56
1.268 pav. Sistemos architektūra.....	57
1.279 pav. Serverio pusės sistemos architektūra.....	58

1.20 pav. Sistemos diegimo žingsniai.....	58
1.28 pav. Konferencijos eigos fazės.....	59
1.29 pav. Prisijungimo prie sistemos langas.....	60
1.30 pav. Informacija apie nepavykusį prisijungimą.....	60
1.31 pav. Sistemos vartotojo sąsaja.....	61
1.32 pav. Informacija apie sistemos vartotoją.....	61
1.33 pav. Sistemos būsenos eilutė.....	61
1.34 pav. Sistemos vartotojų sąrašas.....	62
1.35 pav. Naujo vartotojo pradinė inicializacija.....	63
1.36 pav. Naujos konferencijos pradinė inicializacija.....	63
1.37 pav. Vykstančių konferencijų sąrašas.....	64
1.38 pav. Vartotojui prieinamų konferencijų sąrašas.....	64
1.39 pav. Konferencijos temų sąrašas.....	64
1.403 pav. Vykstančių konferencijų sąrašas.....	64
1.414 pav. Vartotojui prieinamų konferencijų sąrašas.....	64
1.425 pav. Konferencijos temų sąrašas.....	64
1.436 pav. Naujos temos įkėlimas.....	65
1.447 pav. Galimų konferencijos dalyvių sąrašas.....	65
1.458 pav. Konferencijos temos straipsnių sąrašas.....	66
1.469 pav. Recenzijų sąrašas.....	67
1.20 pav. Sugaištas laikas (sekundėmis) funkcijoms atlikti skirtingose sistemoje.....	69
1.21 pav. Laiko sunaudojimas straipsnių skirstymui.....	70
1.22 pav. Užtruktas laikas (minutėmis) straipsnių publikavimui internete.....	70
1.23 pav. Užtruktas laikas (minutėmis) autorių straipsnių priėmimui.....	70
1.24 pav. Užtruktas laikas (minutėmis) konferencijos dalyvių registracijos patvirtinimui.....	71
1.25 pav. Užtruktas laikas (minutėmis) darbui su 4 sistemos funkcijomis.....	71
1.26 pav. Galimos sistemos modifikacijos.....	73

SUMMARY

The project is designed for an actual software area nowadays – automation solutions for conference organization. The product, developed with Java Tapestry technology is a web-based application with document, content management, publication, and conference organization process.

The software for web systems is effective when developers have easy way to make software changes or upgrades. Tapestry framework allows software creation with the good separation between presentation and data exchange levels. This solution allows creating prototype for all customers, and adding special customer's defined functions very simply. The project goal was to improve Tapestry functionality and ability for web applications, and create a web portal framework with content management, document management and users rights management subsystems in it. The design, problems, results and project realization are fully described in this document.

1. ĮVADAS

Šio darbo tikslas – sukurti internete veikiančios konferencijų informacinės sistemos prototipą, kurio programinės įrangos paskirtis – automatizuoti konferencijų organizavimo, rengimo bei informacijos surinkimo funkcijas. Kuriama sistema skirta sumažinti iki minimumo žmogaus pastangas organizuojant konferencijos veiklos procesus internetu. Siekiama, kad sukurta programinė įranga, veikianti interneto naršyklės lange, leistų kiekvienam konferencijos dalyviui saugiai atlikti visus jam reikalingus veiksmus paprasčiausiu keliu, o visos konferencijos sklandaus darbo organizavimui užtektų nedidelio administratoriaus įsikišimo. Duomenų bazėje bus saugoma visa informacija, susijusi su konferencija bei jau anksčiau pasibaigusių konferencijų istorija. Interneto svetainės vartotojai per sąsają bendraus su duomenų baze; potencialūs autoriai galės peržiūrėti konferencijos informaciją, registruotis ir tapti dalyviais, keisti savo duomenis, siųsti straipsnius; recenzentai – rašyti recenzijas; administratorius galės peržiūrėti visą informaciją ir atlikti administravimo veiksmus.

Šiais laikais informacinių technologijų raida itin greita ir daugumą uždavinių automatiškai gali atlikti specialios sistemos. Siekiamos sistemos prototipo tikslas – lanksti, itin lengvai pritaikoma bei modifikuojama sistema, kuri derinama prie konkrečių poreikių bei tobulinama paprasčiausiu keliu, sukuriant papildomus modulius ir juos lengvai prijungiant bei integruojant į esamą sistemą, negriaunant ir nestabdant pastarosios darbo bei atliekamų procesų.

Sukurtame prototipe naudojamos specifiniam duomenų karkasui pritaikytos standartinės kūrimo, atvaizdavimo, redagavimo bei trynimo (CRUD – *Create-Retrieve-Update-Delete*) galimybės. Šių funkcijų rinkinį pateikia dauguma interneto sistemų, leidžiančių atlikti anksčiau minėtas duomenų operacijas, tačiau prototipas naudoja specifines duomenų struktūras, todėl jau sukurtų sistemų pritaikymas yra gana sudėtingas.

Sukurtos prototipinės konferencijų organizavimo sistemos tikslas – pateikti šio tipo programinės įrangos modelį, panaudojant karkasus, kurie leidžia atlikti sistemos modifikacijas lengviausiu keliu.

Sistemos analizės dalyje nagrinėjamas siekiamos sistemos funkcionalumas bei jo techninės realizacijos būdai, aprašomi galimi vartotojo veiksmai bei sistemos vidinės funkcijos. Analizuojant ir renkantis kūrimo platformą, svarstyta keletas alternatyvių technologijų, leidžiančių pilnai arba dalinai atskirti pateikimo vartotojui bei skaičiavimo dalis: .NET, PHP ir

Java. Kadangi vienas iš reikalavimų buvo kuriamos sistemos nepriklausomumas nuo operacinės sistemos, .NET kalbos variantas iš karto buvo palaikytas netinkamu. Prioritetas atiduotas Java technologijoms dėl plataus kūrimo, palaikymo priemonių ir komponentų pasirinkimo, nepriklausomumo nuo platformos. PHP buvo atmesta dėl silpniau išvystytos objektinio programavimo koncepcijos, didesnio primityvumo, mažesnės erdvės eksperimentinei projekto daliai realizuoti.

Tačiau Java technologijos pasirinkimas irgi nėra vienareikšmiškas, kadangi Java pagrindu yra sukurta daug patobulintų priemonių – taip vadinamų karkasų, kurie leidžia sukurti geresnės struktūros programos kodą atskiriant sistemos modelį nuo jos pateikties (vaizdo) ir valdymo. Atlikta galimų JAVA karkasinių technologijų analizė pagal aštuonis kriterijus ir remiantis jos išvadamis pasirinktas geriausias duomenų/rezultatų pateikties ir skaičiavimo logikos atskyrimo problemą panaikinantį sprendimas – Tapestry karkasas [2], kuris nors ir nedideliu skirtumu, tačiau pasirodė pranašesnis prieš Struts.

Sistemos kūrimo pagrindimui buvo remtasi literatūros šaltiniais. Konferencijos sistemos funkcionalumas buvo suformuluotas remiantis užsakovo pageidavimais ir tarptautinių VLDB, ADBIS, ISD [1] konferencijų sistemų galimybių analize. Ne visos konferencijos, tame tarpe ir KTU konferencija „Informacinės Technologijos“, turi tokio tipo įrangą, todėl kuriama sistema buvo numatyta kaip prototipas tokiai sistemai realizuoti.

Sistemos architektūra ir programinė realizacija aprašyta, pateikiant fragmentus iš programinės įrangos specifikacijų, detaliam aprašomam jos panaudojimo atvejai, pateikiami konferencijų vykdymo proceso vaizdai, nagrinėjamos paketų diagramos, klasių sąveikos. Pateiktos diagramos leidžia suvokti visą konferencijos procesą, apibrėžia vartotojų lygius, sąlygojančius jo funkcijas sistemoje.

Sistemos kokybės įvertinimui buvo remtasi kokybės įvertinimo faktoriais ir kriterijais. Sistemos kūrėjų siekis, įgyvendinti programinę konferencijų automatizavimo sistemos realizaciją, kuri automatizuotų konferencijų eigos procesus, taip sutaupant resursų naudojimą konferencijų automatizavimo sistemoms. Sistema laikoma kokybiška, nes jos realizacijos detalus funkcijų tyrimas parodė, jog panaudojus šią programinę įrangą, resursų naudojimo sumažėjimo koeficientas yra itin didelis. Kokybės tyrime atlikti detalūs skaičiavimai bei pateiktos skaičiavimų vizualizacijos, parodančias automatizavimo naudas konferencijų organizatoriams.

Darbą sudaro šešios dalys:

Įvadas. Įžanginė darbo dalis, aprašanti sistemos kūrimo tikslus, principus, realizacijoje panaudotus sprendimus bei taikymo galimybes.

Interneto sistemų kūrimo technologijų ir metodų analizė. Pateikiama sistemų, kuriomis galima realizuoti šį projektą, analizė [3], [4], [6], [7], nagrinėjami konkrečių sistemų privalumai ir trūkumai, atliekamas lyginimas bei dėstomas pasirinktos technologijos pagrindimas.

Karkasais grindžiama kūrimo metodika. Gilinamasi į Tapestry karkaso architektūrą, pateikiamas analitinis jo veikimo modelis, analizuojamos karkaso valdiklių galimybės, išryškinami jo privalumai, aiškinamas programinės įrangos, sukurtos su Tapestry technologija, modifikacijos galimybės.

Metodikos taikymas konferencijų programinės įrangos prototipui sukurti. Skyriuje analizuojamas sukurtas konferencijų informacinės sistemos prototipas, aprašomas jo architektūrinis sprendimas, pateikiama funkcionalumo analizė. Pateikiami ryšiai tarp komponentų, sąveikos diagramos, aprašomi konferencijoje vykstantys procesai. Programinė įranga nagrinėjama pasitelkiant klasių, veiklos, sekų diagramas, pateikiami detalūs panaudojimo atvejų aprašai, atliekamas sukurtų paketų ir klasių detalizavimas.

Sukurto produkto veikimo ir taikymo galimybių analizė. Skyriuje pateikiami sukurto produkto sąsajos vaizdai, susiejant juos su funkcijomis, kad būtų galima vaizdžiai suvokti aprašytus veikimo principus. Pateikiama detali kelių funkcijų naudojimo efektyvumo analizė, atliekami resursų taupymo skaičiavimai, pagal kuriuos įvertinama produkto kokybė. Pateikiama produkto galimybių rinkoje analizė, svarstomos jo taikymo galimybės ir kryptys, į kurias galima orientuotis modifikuojant sukurtą sistemą ir plečiant jos funkcijas.

Išvados. Pateikiamos išvados, apibendrinančios bei apibrėžiančios projekto rezultata, panaudotas technologijas, sprendimus, jų pasirinkimą, fiksuojami projekto privalumai.

Literatūra. Pateikiamos literatūros ir informacijos šaltinių sąrašas, kuriuo buvo naudotasi, dirbant su projektu, priimant sprendimus ar sprendžiant kūrimo eigoje iškilusias problemas.

Atliktas darbas turi daug privalumų, lyginant su kitomis internete veikiančiomis programomis. Visų pirma, sistema sukurta, jai panaudojus pakankamai netradicinę internetinės programoms koduoti sukurtą Java technologiją – „Java Tapestry“. Technologija sukurta ir adaptuota vaizdo pateikimui atskirti nuo veiklos procesų, nes šių dviejų iš esmės skirtingų veiklų maišymasis yra pagrindinė priežastis, daranti interneto programinės įrangos modifikacijas pakankamai sudėtingomis. Panaudojus karkasinę kūrimo technologiją, kokia ir yra Tapestry, kodas atskiriamas nuo vaizdavimo, todėl programos tampa atviros pakeitimams ir įvairių lygių modifikacijoms.

Kitas darbo privalumas – jo paliesta sritis. Kuriamas konferencijų automatizavimo projektas – pagalbininkas konferencijų organizatoriams, nes daugelis šiuo metu vykstančių konferencijų yra neautomatizuotos ir jų vykdymas reikalauja daug žmoniškųjų resursų. Sistemos kūrėjų tikslas – automatizuoti šiuos veiklos procesus, didžiąją dalį darbo perkeliant sistemai ir taip atnešant tiesioginę suskaičiuojamą naudą organizatoriams – administratorių laiko taupymą, o tai tiesiogiai įtakoja organizatorių kaštus. Programa turi savyje panašumų ir į kitas interneto sistemas (turinio valdymo [4], dokumentų valdymo, nuotolinio mokymosi sistemos), todėl įmanomas jos perkėlimas į kitą sritį. Lengvas modifikavimas leidžia siūlyti sistemą įvairių rinkos segmentų atstovams, nes ji pati ir jos įmanomos transformacijos yra aktualios šios dienos verslo, mokslo, valstybinėms organizacijoms.

2. INTERNETO SISTEMŲ KŪRIMO TECHNOLOGIJŲ IR METODŲ ANALIZĖ

2.1 Konferencijų programinės įrangos poreikių analizė

Išanalizavus analogiškų jau sukurtų sistemų veiklą, po konsultacijų su konferencijų dalyviais, skaitytojais bei administratoriais buvo suformuoti būsimų sistemos vartotojų poreikiai ir sukurtas bendras sistemos veiklos procesų vaizdas. Konferencijų informacinės sistemos kūrimo pagrindinis tikslas – realizuoti lengvai ir paprastai tobulinamą programinę įrangą, kuri nebūtų pririšta prie platformos ir kurios tobulinimas nereikalautų didelių jos naudotojų kaštų. Produkto kūrėjų siekis – konferencijų organizavimo, rengimo bei informacijos surinkimo pilnas automatizavimas. Kuriama sistema yra skirta tam, kad sumažinti iki minimumo žmogaus pastangas organizuojant konferencijos veiklos procesus. Kuriama programinė įrangą, veikianti interneto naršyklės lange, turi leisti kiekvienam konferencijos dalyviui atlikti reikiamus veiksmus paprasčiausiu keliu, o visos konferencijos sklandaus darbo organizavimui užtenka nedidelio administratoriaus įsikišimo. Duomenų bazėje saugoma visa informacija, susijusi su konkrečia konferencija bei jau anksčiau pasibaigusių konferencijų istorija. Ši informacija dinamiškai pateikiama konferencijos svetainėje, kurioje vartotojas gali ne tik peržiūrėti informaciją, bet ir pats tapti jos aktyviu dalyviu. Šio tipo sistemų bendrą veiklą, galėtume skirstyti į dvi dalis: konferencijos valdymą ir prieigų prie sukauptos istorijos valdymą. Konferencijos eigos valdymo uždaviniai, tai jau minėtas uždavinys – mažinti administratorių resursų poreikį, visas įmanomas funkcijas automatizuojant. Sistemoje, ypač po ilgo jos naudojimo laiko, bus sukaupta informacija apie visas įvykusias konferencijas, kuri toli gražu ne visa bus neprisiregistravusiems vartotojams, todėl kita kūrimo sritis, tai sukurti lengvą informacijos istorijos kaupimo, publikavimo ir prieigų nustatymą. Šiuos procesus turės atlikinėti konferencijos administratorius, tačiau sistemos kūrėjai kuria įrankius, leidžiančius tai atlikti lengvai ir paprastai.

2.2 Java karkasų analizė

Renkantis konferencijų valdymo portalo kūrimo platformą, buvo svarstomi keli variantai: .NET, PHP, JAVA [5], [13], [15]. Kadangi turėjome viziją, jog kuriama programinė įrangą privalo būti nepriklausoma ir veikti tiek „UNIX“, tiek „Windows“ šeimų operacinėse sistemose,

ASP kalbos variantas buvo greitai atmestas. .NET technologija, nors ir pilnai būtų tikusi reikiamam programinės įrangos funkcionalumui realizuoti, veikia ir yra suderinama tik su „Microsoft“ platformos produktais, todėl mums netiko. Prioritetas buvo paskirtas Java technologijai dėl plataus kūrimo, palaikymo priemonių ir komponentų pasirinkimo, visiško nepriklausomumo nuo platformos. PHP nebuvo pasirinktos dėl silpniau išvystytos objektinio programavimo koncepcijos, didesnio primityvumo, kuris itin yra jaučiamas rašant didelius projektus, sudarytus iš daugybės kodo eilučių, išskaidytų į atskirus modulius.

Pasirinkus Java programavimo kalbą, buvo renkama technologija, kuria būtų galima vykdyti projektą, kad jo realizacija atitiktų sistemos kūrėjų iškeltus pagrindinius tikslus – lengvas ir paprastas esamos sistemos adaptavimas naujiems užsakovų poreikiams bei nedaug resursų reikalaujantis sistemos tobulinimas. Nors Java šiuo metu šiek tiek užleidžia savo pozicijas paprastų interneto sistemų kūrimo PHP technologijai, tačiau jos karkasinių technologijų projektavimo galimybės kur kas didesnės. Peržiūrėjus nemokamus Java karkasus internetinių projektų kūrimui, detaliam analizei buvo atrinkti šie produktai [8], [9]:

- **Java Struts;**
- **Java Spring MVC;**
- **Java WebWork;**
- **Java Tapestry;**
- **Java JSF.**

Toliau pateikiama šių karkasų analizė, įvertinanti savybes, leidžiančias spręsti apie technologijos tinkamumą kuriamai konferencijų informacinei sistemai.

2.2.1 Java Struts

Tai šios dienos JAVA internetinių projektų standartas, itin populiarus Europoje ir Azijos šalyse. Lengvai išmokstama technologija, nes labai daug pavyzdžių, literatūros bei informacijos internete [10], [14], o tai yra labai svarbus faktorius, mokantis koduoti visiškai nauja technologija. Itin patogiai išspręstas HTML kalbos žymių (angl. tags) naudojimas. Technologijos trūkumas – sudėtingai naudojamas „ActionForms“ kontroleris, įnešantis daug neaiškumo, nes Struts technologija nepasižymi lengvu testavimu ir klaidų aptikimu, ypač kai kuriama su „StrutsTestCase“ įrankiais, kurie atlieka pilną integravimą, tačiau ne testavimą.

2.2.2 Java Spring MVC

Ši internetinių sistemų technologija pasižymi lengvu testavimu, palengvinančių kūrėjų darbą, nes galima panaudoti atvirkštinį testavimą. Taip pat šis šablonas turi integruotą savybę informacijos eksportui: technologija turi modulius, leidžiančius duomenis ar rezultatus eksportuoti į kitas sistemas žinomais formatais: PDF, XML, EXCEL. Sistemos trūkumas yra tai, jog projektuose, realizuotose su Java Spring MVC neįmanoma išvengti didelio kiekio kodo JSP failuose, o tai daro sistemą painesne ir sunkiau modifikuojama. Ši sistema nėra labai populiari ir labiau taikoma mažesniems projektams, nes tai tik tobulesnis paprasto JSP variantas.

2.2.3 Java WebWork

Ši technologija pasižymi paprasta, aiškia modulių jungimo į karkasą architektūra – tai pagrindinė sąlyga, norint turėti lengvai plečiamą ir tobulinamą sistemą. HTML žymių (angl. Tags) biblioteka lengvai modifikuojama, galima įsidėti savo modulius bei žymes. Šiai technologijai nėra išleista daug pavyzdžių, teorinės medžiagos, todėl sunku ją išmokti, o tai ir sąlygoja jos nedidelį populiarumą internetinių programų kūrime.

2.2.4 Java Tapestry

Technologija, akivaizdžiai konkuruojanti su Java Struts dėl Java šeimos technologijų lyderio pozicijų interneto sistemų ir portalų kūrime. Tapestry leidžia dirbti itin našiai, nes kodas yra gerai atskirtas nuo HTML šablonų. Sistemos lengvai modifikuojamos, nes HTML tarpai įtraukiami kaip dalys, o tai įgalina paprastai keisti sistemos išvaizdą dizaineriams, nes jie gali daryti išvaizdos pakeitimus visai be programuotojų įsikišimo. Sistemų kūrimas, naudojantis šia technologija, reikalauja patirties, kurią įgavus, darbas tampa našesnis, greitesnis nei dirbant kitomis Java karkasų sistemomis. Tobulinimas bei modulių atnaujinimas atliekamas gana paprastai, aprašant naujas modulių sąveikas XML konfigūraciniuose failuose. Itin įdomi ir kitose technologijose neturinti analogų galimybė – konfigūracinių failų valdymas. Karkasas leidžia naudoti keletą konfigūracinių failų, o jų naudojimas pagal situacijas aprašomas kontroleryje. Tai leidžia išspręsti darbo aplinkos personalizavimo klausimą, apibrėžti vartotojui galimų veiksmų sekas, visiškai nenaudojant teisių valdymo kontrolerio.

2.2.5 Java JSF

Technologija, paremta J2EE standartu. Šio standarto technologijos ypatumus nesunku

perprasti, nes pavyzdžių internete apstu. Technologija gana primityvi, nesudėtinga ir tuo pačiu apribotas jos naudojimas dideliuose projektuose, nes pats karkasas turi nedaug valdymo ir modulių funkcijų. Tinkama mažiems uždaviniams, nes nesudėtingas karkasas leidžia greit kurti nedidelės apimties projektus. Ši technologija nėra išbaigta, todėl jos trūkumus autoriai vis dar šalina ir ją tobulina.

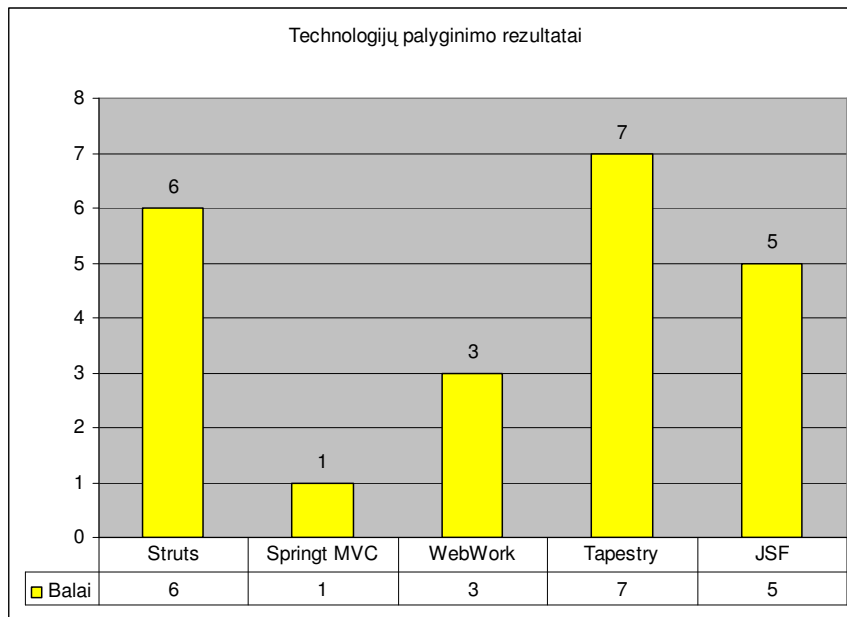
2.3 Java technologijų palyginimo rezultatai

Atlikus bendrą Java karkasų analizę, buvo atrinkti aštuoni kriterijai, labiausiai liečiantys konferencijų valdymo informacinės sistemos kūrimą, ir pagal juos atliktas technologijų sulyginimas. Lyginimo rezultatai fiksuoti 1.1 lentelėje bei grafine forma (1.1 pav.), kur aiškiai matyti Tapestry technologijos tinkamumas projekto tikslams įgyvendinti.

1.1 lentelė

Java karkasų lyginimo rezultatai

Technologija/ Savybė	Struts	Spring MVC	WebWork	Tapestry	JSF
Tinkamumas dideliems projektams	X			X	
Lengva modifikacija	X			X	X
Mažas kodo kiekis JSP puslapiuose				X	
Lengvas testavimas		X	X	X	X
Lengvas technologijos perpratimas	X				X
Lengvas HTML keitimas	X		X	X	
Papildomi kodavimo/integravimo įrankiai	X		X	X	X
Literatūros pakankamumas internete	X			X	X



2.1 pav. Java karkasų lyginimo rezultatai

Atlikus analizę, buvo pasirinktas Java Tapestry karkasas. Ištirtos ir atrastos technologijos savybės mums leidžia teigti, jog šios sistemos funkcionalumas bei galimybės pilnai tenkina konferencijų valdymo informacinės sistemos poreikius. Karkasas tinkamas dideliems projektams, lengvai modifikuojami pakeitimai, atnaujinimai, tai leis kuriamą produktą padaryti lanksčiu ir paprastai adaptuojamu pagal užsakovo poreikius. Mažas kodo kiekis JSP puslapiuose – privalumas, leisiantis siūlyti užsakovui keletą ar daugiau sistemos dizaino variantų, nes dizaino kūrimo darbus atliekantiems žmonėms bus paprasta atlikti stiliaus pakeitimus. Tapestry labai nedaug pasirodė pranašesnė prieš Struts, neveltui šios sistemos yra laikomos labai konkurencingos viena kitai. Vienas iš esminių postūmių, nulėmusių pasirinkimą ne Struts karkaso naudai – pastarojo populiarumas, nes šis informacinės sistemos kūrimas turi ir eksperimentinę pusę – naujos, dar nenaudotos technologijos išbandymą ir perpratimą kuriant didelės apimties sudėtingą sistemą, atliekančią internete vykstančios konferencijos automatizavimo darbus.

2.4 Analizės išvados

Analizės metu buvo pagrįstas kūrimo metodologijos pasirinkimas ir pateikti jos principai.

Pirmiausia buvo nagrinėjamos platformos, tinkamos konferencijų sistemai realizuoti. Kadangi svarbus faktorius buvo platformos nepriklausomumas ir konceptualus aiškumas, iš trijų

kandidatų – .Net, PHP ir Java – buvo pasirinkta pastaroji.

Kadangi kūrimas Java platformoje turi daug variantų, buvo išskirti 8 kriterijai, kurie padėjo palyginti tarpusavyje eilę pažangių Java karkasų ir pasirinkti tinkamiausią – Tapestry karkasą. Jo galimybės plečiamumui užtikrinti buvo įvertintos geriausiai, kadangi jame reikia rašyti mažiausiai kodo, didžiausias nepriklausomumas tarp modelio – vaizdo – kontrolerio kodo dalių, patogūs šablonai ir konfigūracijos failai. Be to, domino pačios technologijos įsisavinimas ir galimybė sukurti geresnės architektūros sistemą.

3. KARKASAIŠ GRINDŽIAMA KŪRIMO METODIKA

Šiame skyriuje nagrinėjami karkasais grindžiamo kūrimo principai, detaliau paliečiant Tapestry karkasą.

3.1 Karkasų sąvokos

Karkasas yra tarpusavyje susijusių klasių ir komponentų visuma, kuri sudaro tam tikros klasės uždavinių pakartotino panaudojimo projektą. Karkasas skiriasi nuo paprasto klasių rinkinio, kuriame kiekviena klasė atlieka izoliuotą vaidmenį. Priešingai, karkasas apsprendžia programos architektūrą. Jis pateikia projektuotojui pradinį sprendimą ir leidžia susikoncentruoti ties individualiomis dalykinės srities savybėmis. Naudojant karkasus, kūrimas prasideda nuo pakartotino naudojimo klasių, kurias galima plėsti. Idealiu atveju reikia pateikti keletą specifinių poklasių, ir sistema sukurta. Taigi karkasai leidžia rašyti mažiau kodo, duoda geresnę programos struktūrą, ir didesnę jos elementų suderinamumą.

Tačiau karkasai turi ir trūkumų – reikia papildomo laiko juos įsisavinti, kartais lėtesnis veikimas. Tačiau konferencijų sistemos atveju greitas veikimas nėra toks svarbus, be to, veikimo skirtumas nėra toks didelis. Žymiai svarbiau užtikrinti sistemos plečiamumą, kadangi konferencijų veiklos taisyklės dažnai keičiasi. Skirtingų organizacijų rengiamų konferencijų veikla iš esmės turi tas pačias funkcijas, tačiau jų naudojimo taisyklės šiek tiek skiriasi. Todėl norint užtikrinti platesnį sistemos panaudojimą, kūrimui tikslinga taikyti nors ir sudėtingesnę, bet labiau ilgalaikę technologiją.

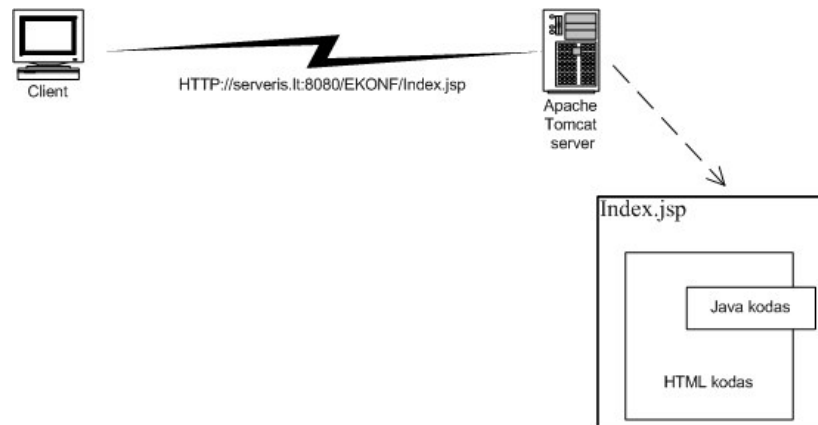
3.2 Java karkasinės technologijos

3.2.1 Java servletų technologija

Servletų technologija sukurta Java programavimo kalbos taikymams internete. Tai serverio funkcionalumus praplečiantys papildomi komponentai, leidžiantys generuoti HTML kodą. Su dinaminėmis HTML puslapių atsiradimų (JSP technologija) šie papildomi komponentai atlikdavo skaičiavimus, logikos operacijas, o visus savo rezultatus pateikdavo generuodami HTML failą. Generavimui pasibaigus, jis nusiunčiamas į kliento naršyklę. Šio tipo sistemos buvo labai painios, nes skaičiavimų logika glaudžiai persipynusi su duomenų išvedimu, todėl didelių sistemų modifikavimą sunku atlikti, o neturint detalios dokumentacijos ir išvis neįmanoma.

3.2.2 JSP technologija

Dinaminių HTML puslapių kūrimo technologijos supaprastino internetinių svetainių kūrimą. JavaServerPages (JSP) technologija leidžia įterpinėti Java kalbos kodą tiesiai HTML dokumente, aprašančiame duomenų ir rezultatų pateikimo formą. Naudojant servletus, kiekvieną kartą, gavus kliento užklausą, kodas yra kompiliuojamas ir tada kuriamas HTML dokumentas. JavaServerPages neatlieka kaskart naujo kompiliavimo, jei jis jau buvo atliktas, o kodas nebuvo keistas. Nors kodo įterpimas į HTML dokumentą labai padidino Java interneto svetainių verslo paslaugų galimybes, Java programavimo kalbos kodo rašymas tiesiai HTML dokumente vis tiek įneša daug painumo, ypač dideliuose projektuose. Tokiu būdu parašytas kodas išlieka sunkiai skaitomas ir jaučiamas skaidrumo trūkumas HTML ir JAVA programinių kodų atskyrimui (3.1 pav.).



3.1 pav. Java ServerPages architektūra. Aiškiai matomas JAVA ir HTML kodų maišymasis

3.2.3 EJB technologija

EJB (JavaBeans) technologijos atsiradimas suteikė galimybę didelę dalį kodo perkelti į išorines klases, ten iškeliant visas veiklos funkcijas ir procesus. Šias klases Java ServerPages puslapiai naudoja kaip atskirus objektus ar komponentus [7]. Kiekviena Java klasė gali būti traktuojama kaip EJB komponentas, jei tenkina tokius reikalavimus:

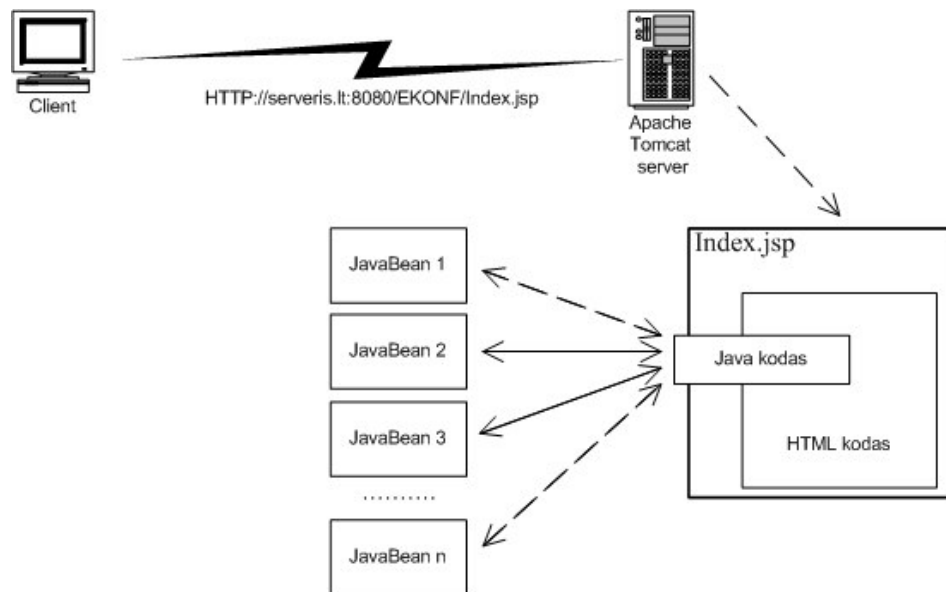
- Turi bendrus vidinius klasės kintamuosius, kuriais perduodamos reikšmės;
- Kiekvienam iš vidinių kintamųjų turi metodą `NustatytiReikšmę()` (`setValue()`), kuriuo inicializuojama arba nustatoma kintamojo reikšmė.
- Kiekvienam iš vidinių kintamųjų turi metodą `gautiReikšmę()` (`getValue()`),

kuriuo sužinoma nagrinėjamojo kintamojo reikšmė.

- Turi vieną ar daugiau nei vieną klasės konstruktorių (inicializatorių).

JavaBeans komponentai naudojami daugumoje karkasų, tame tarpe ir Tapestry, nes be šios technologijos karkasų naudojimas nebūtų populiarus, nes netektų didžiosios daugumos savo teikiamų patogumų kodo kūrėjams.

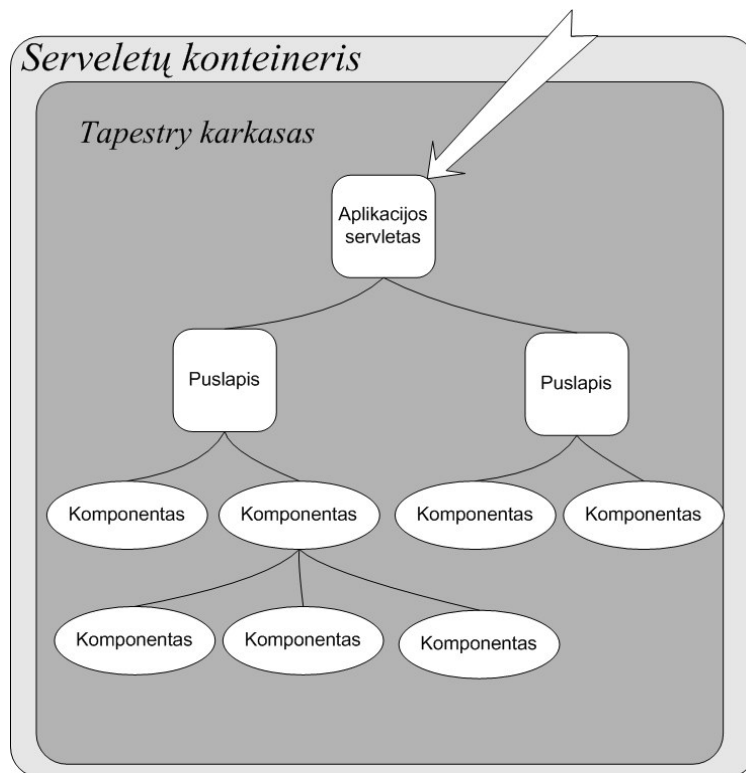
Koduojant internetines sistemas standartine JSP technologija ir naudojant Java kodo gabalus išorėje (EJB technologija), vis tiek nėra išspręstas duomenų atvaizdavimo ir skaičiavimo atskyrimas. Tai supaprastina darbą, lyginant su paprastu JSP, tačiau maišymasis vis tiek išlieka ir sukelia sunkumų dideliuose projektuose (3.2 pav.).



3.2 pav. Java ServerPage su EJB architektūra. JavaBeans naudojimas JSP puslapiuose

3.3 Java Tapestry karkasas

Java Tapestry karkasinė technologija, pritaikyta JSP, yra sprendimas, norint atskirti skaičiavimų procesus nuo pateikties procesų, nes joje gerai išspręstas duomenų ir veiklos atskyrimo klausimas [2], [6], [11]. Tai komponentinė technologija (3.3 pav.), o atskirų komponentų sąveikos aprašomos XML konfigūraciniuose failuose. Tapestry kontrolieris apdoroja vartotojo naršyklės užklausą ir grąžina jai atsakymą. Kiekvienas Tapestry karkaso principu parašytos programos puslapis turi unikalų pavadinimą, kuriuo jis yra identifikuojamas karkaso kontrolierio, pagal šį unikalų pavadinimą yra atrenkami HTML šablonai (angl. Templates), kuriuos puslapis naudoja pagal XML konfigūracijoje aprašytą struktūrą.



3.3 pav. „Java Tapestry Framework“ komponentinio skaidymo pavyzdys

Itin mažas Java kodo kiekis JSP puslapiuose leidžia paprastai ir aiškiai kurti sistemas, paliekant paprastą ir aiškų sistemos tobulinimo kelią ateityje. Žemiau pateiktame pavyzdyje aiškiai matosi, kaip į HTML kodą įrašomi kreipiniai į metodus bei nustatomas konfigūracinis failas, pagal kurį bus atliekami tolesni informacijos pateikimo veiksmai. Jwcid žymė nurodo unikalų pavyzdžio komponentą pavadinimu insertDate, kuris ir atliks puslapio informacijos mainus su kitais komponentais (3.4 pav.).

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Simple</title>
</head>
<body>
  <p>This application demonstrates some dynamic
    behavior using Tapestry components.
  <p>The current date and time is:
    <b><span jwcid="insertDate">This Text Will Be Replaced</span></b>
  <p>Click <a jwcid="refresh">here</a> to refresh.
</body>
</html>

```

3.4 pav. HTML kodo dalis, kurioje naudojami Tapestry komponentai

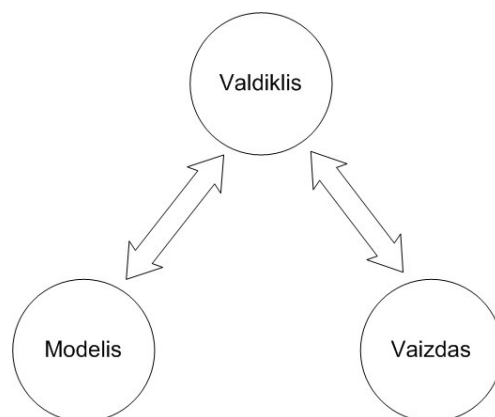
Dirbant šia technologija, galima grįžti ir prie paprasto Java ServerPages. Naudojant Tapestry karkasą komponentų sąveikai aprašinėti, galima tuose pačiuose HTML puslapiuose įterpinti JAVA kodą. Savaimė suprantama, tai nėra gera praktika ir ji tikrai nerekomenduojama, nes įneša daug painumo programų sistemos architektūroje, tada Tapestry karkaso naudojimo tikslas – sukurti kuo aiškiau skaitomą ir tobulinamą sistemą – nebetenka prasmės.

3.4 Kūrimo metodika naudojant Tapestry karkasą

3.4.1 Java Tapestry MVC šablonas

Dauguma karkasinių sistemų naudoja modelio, vaizdo, kontrolerio (angl. MVC – Model, View, Controller) šablonus su įvairiomis jų modifikacijomis. MVC šablonas skirstomas į tris dalis:

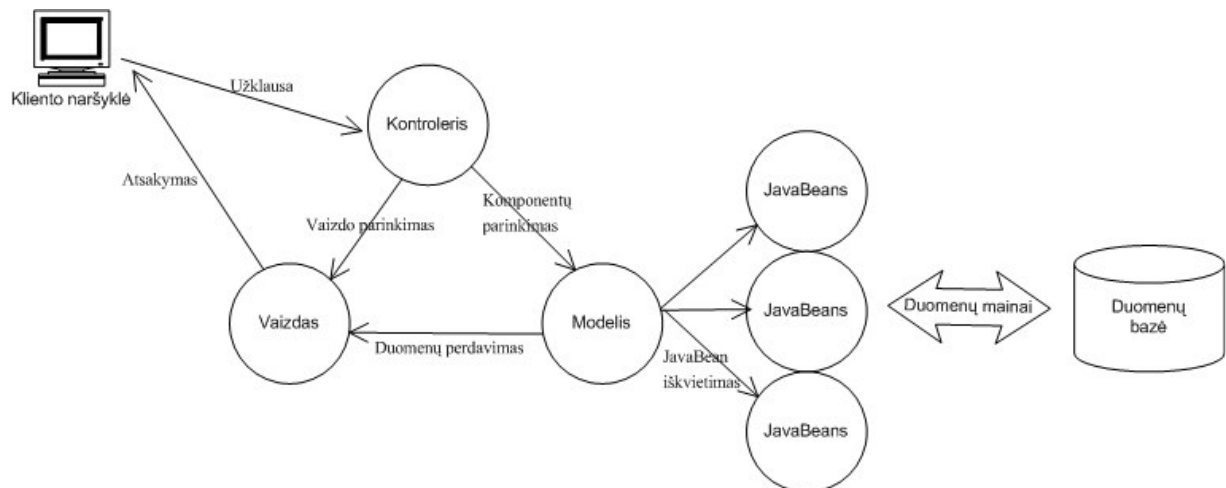
- Komponentas **Modelis** (Model) atsakingas už sistemos sąveikas su JavaBeans komponentais, savo moduluose saugo programų sistemos būsenas.
- Komponentas **Vaizdas** (View) atsakingas už duomenų pateikimą programos komponentams, atliekantiems vaizdavimą.
- Komponentas **Kontroleris** (Controller) šablone atlieka dvi pagrindines funkcijas (3.5 pav.). Jis yra informacijos valdiklis, duomenų mainų tarp Vaizdo ir Modelio kontroleris, užtikrinantis teisingą duomenų kaitą. Antra funkcija – tai vartotojo užklausų interpretavimas, jų persiuntimas modeliui bei atsakymų kliento naršyklei valdymas.



3.5 pav. MVC šablonas atskiria vaizdavimą nuo veiklos procesų

3.4.2 Java Tapestry karkaso architektūra

Konferencijų automatizavimo programinės įrangos sistema buvo kuriama pasinaudojus Tapestry karkasu, todėl pateiksime paties karkaso analizę. Java Tapestry karkasas taip pat kaip ir dauguma karkasinių sistemų paremtas MVC (modelis – vaizdas – kontroleris) šablonu, žemiau pateiktame pavaizduotoje komponentų sąveikos diagramoje (3.6 pav.) regime, koku principu veikia sistema.



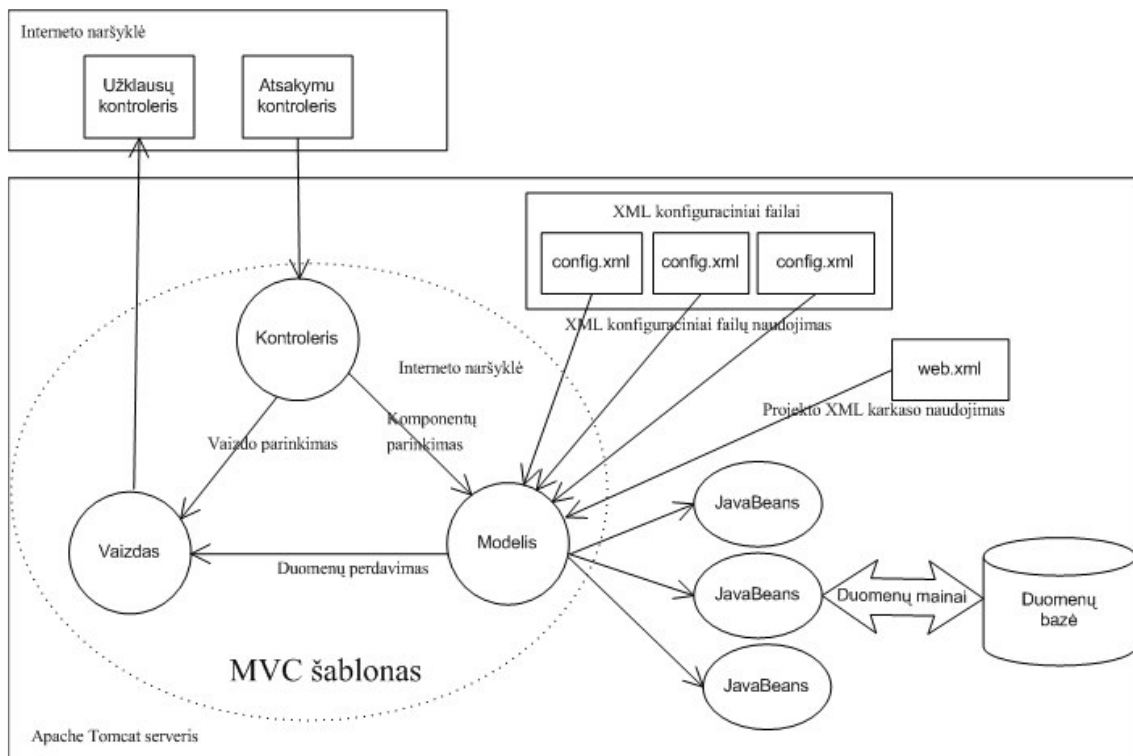
3.6 pav. Tapestry karkasų sistemos komponentų sąveika

Sistemos ašis – tai trikampiu išdėstytas modelio – vaizdo – kontrolerio šablonas, kuris ir valdo visą sistemos darbą. Interneto naršyklė, siunčia užklausą (request) sistemos kontrolieriui, kuris pasinaudodamas būsenų sąveikų valdikliu, iškviečia atitinkamus sistemos komponentus (tai gali būti tiek kiti puslapiai, tiek JavaBeans), ir rezultatus persiunčia į vaizdavimo kontrolierį. Vaizdas atsakingas už JSP puslapio suformavimą, jo dinaminį generavimą į HTML, kuris ir persiunčiamas vartotojo naršyklei kaip atsakymas (response). Informacijos mainai su duomenų bazėmis vyksta JavaBeans komponentų lygyje, ir aukštesnio lygio kontrolierių jie neliečia.

Šioje principinėje MVC panaudojimo schemoje nėra detalios pavaizduotos komponentų sąveikos bei konfigūracinių failų naudojimas, kuris taip pat yra labai svarbi Tapestry karkaso dalis, daranti jį išskirtiniu iš kitų karkasinių sistemų. Žemiau pateiktoje diagramoje (3.7 pav.) jau galima matyti ir konfigūracijų naudojimą bei valdymą ir kitas sistemos sąveikas:

- Sistemos būseną saugoma MVC šablono komponento modelio objektuose, modelis taip pat atsakingas už sąveikas su JavaBeans komponentais.

- Kai vartotojas atlieka veiksmus, naršyklės kontrolieris siunčia užklausą serveriui, kuri yra apdorojama MVC karkaso kontrolierio pagal web.xml faile aprašytus veiksmus. Taip pat kontrolieris gali naudoti ir kitus XML konfigūracinius failus, jei sistemos teisių valdiklis nurodo, jog naudojamas kitas, o ne pagrindinis konfigūracinis failas.
- Komponentas „vaizdas“ valdomas dviejų komponentų (Model, Controler), jam pateikiančių atvaizdavimo duomenis, kuriuos komponentas siunčia kliento naršyklės kontrolieriui kaip atsakymą (response).

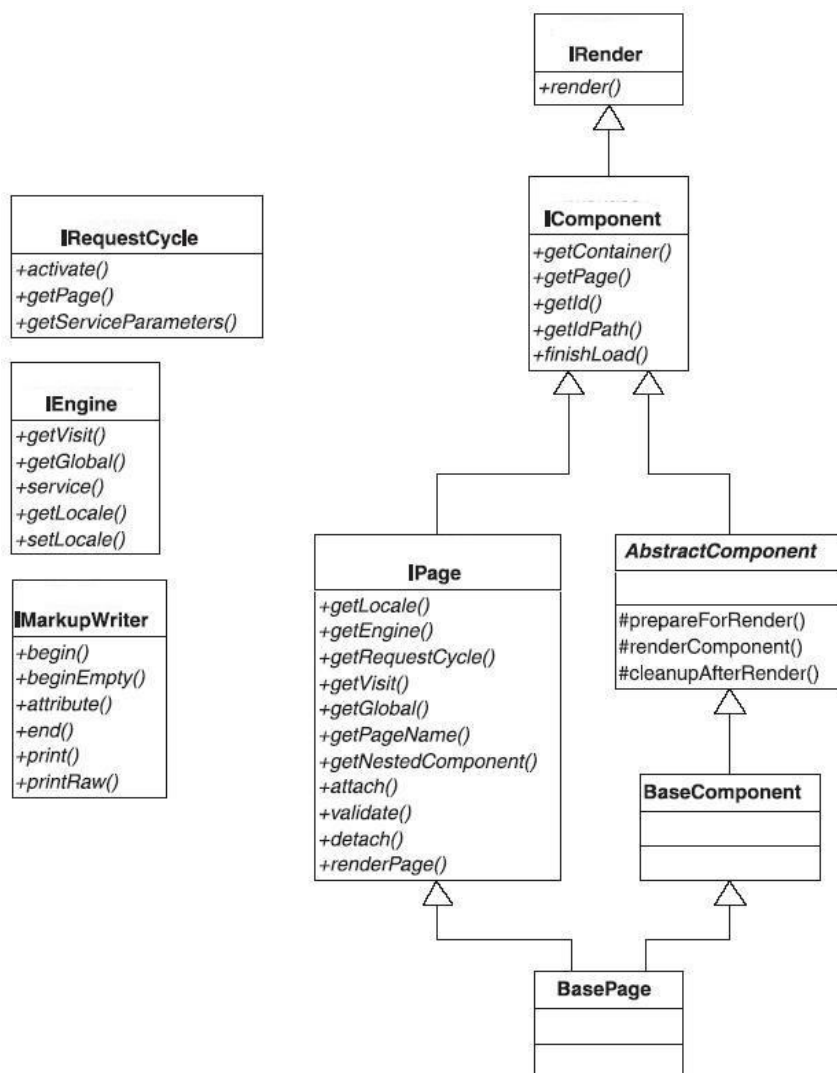


3.7 pav. Detali Tapestry karkasų sistemos komponentų sąveika

Išanalizavus karkaso paslaugų pasiskirstymo modelį, galime toliau pažvelgti į jo klasių architektūrą. Žemiau esančioje klasių sąveikos diagramoje pavaizduota, kaip karkaso paslaugų funkcijos paskirstytos tarp jo klasių. Šakninė objektų hierarchijoje karkaso klasė *IRender* skaito konfigūracinius failus ir apibrėžia sistemoje naudojamus objektus ir komponentus pasinaudodama kitomis paveldimomis klasėmis. *IPage* ir *IComponent* klasės naudojamos komponentų reikšmių gavimui ir persiuntimui į šakninę klasę. *AbstractComponent* yra

reikalinga, jei joks verslo klasių komponentas nėra naudojamas, tačiau *IRender* klasė negali atlikti numatytų funkcijų be komponento (bent vieno), todėl šis objektas sukuria abstraktų objektą, neatliekanti jokio funkcionalumo. Jei komponentas yra, tada tiesiog naudojama *IComponent*, kurioje yra visi apdorojamo puslapio parametrai. *BaseComponent* klasė yra komponentų valdiklis, kurią visos kitos anksčiau paminėtos klasės naudoja komponentui išrinkti jo parametrus nustatyti. Visus sukurtus ir konfiguracioniuose failuose aprašytus puslapius, Tapestry karkasas iškviečia būtent panaudodamas *BasePage* klasę.

Apdorodamas Tapestry puslapį, kontrolieris jam automatiškai inicializuoja dar tris papildomas klases: *IRequestCycle*, *IEngine*, *IMarkupWriter*. *IRequestCycle* klasė saugo informaciją apie puslapio, kuriame ji buvo inicializuota, užklausą iš kliento naršyklės, ši klasės informacija ir metodai gali būti naudojami atsakymo (response) išsiuntimui vartotojui. *IEngine* – tai objektas, kurį naudoja karkasas savo vidinių, puslapius apdorojančių klasių valdymui ir inicializacijai. *IMarkupWriter* klasė atlieka galutinio HTML kodo generaciją, po kurios jis siunčiamas vartotojui.

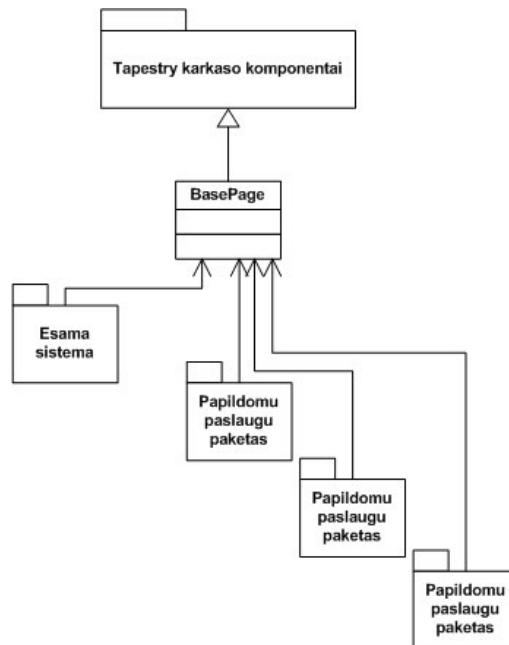


3.8 pav. Tapestry karkasų sistemos klasių architektūra

3.4.3 Java Tapestry karkaso plėtimas

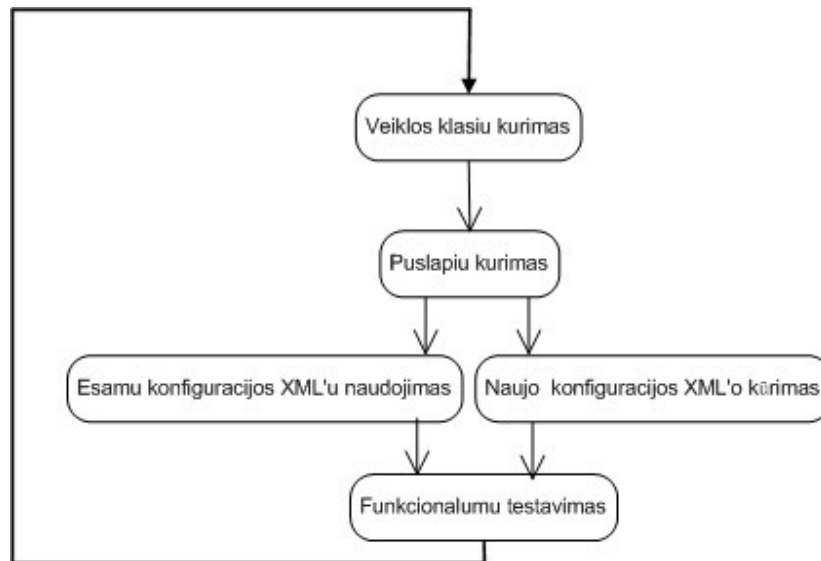
Karkasinių sistemų kūrimas vyksta etapais, kuriant naujus funkcionalumus realizuojančius komponentus ir jungiant juos prie esamos sistemos. Nauji moduliai jungiami aprašinėjant jų sąveikas su esamais sistemos komponentais konfigūraciniuose XML failuose, kuriais naudojasi Tapestry karkaso objektai ir kontrolieriai, atsakingi už veiksmų ir procesų valdymą. Naują funkcionalumo realizaciją turi sudaryti puslapiai, atliekantys vaizdavimą vartotojui bei verslo klasės atliekančios šio funkcionalumo procesus bei naudojamos informacijos valdymą. Šių dviejų rūšių komponentų naudojimas aprašomas konfigūracijose, sukurti puslapiai pasiekiami Tapestry karkasui naudojant jau anksčiau minėtą klasę *BasePage*,

kuri yra žemiausias karkaso vienetas, jungiantis sukurtus komponentus ir Tapestry.



3.9 pav. Tapestry karkasų sistemos plėtimas

Žemiau pateiktame paveiksle vaizduojama veiksmų seka, kuria turi būti atliekamas naujų modulių integravimas į sukurtas Java Tapestry sistemas. Pirmiausia atliekama verslo klasių realizacija, turint jų išbaigtumą, pradedami kurti svetainės puslapiai, kurie bus naudojami naujų funkcijų pateikimui atlikti. Natūralu, jog kaž kurios iš anksčiau paminėtų dalių gali ir nebūti. Jei kuriami atvaizdavimo papildiniai, tai jiems dažniausiai nėra reikalingos naujos verslo klasės, nes jie neįneša naujų funkcionalumų, ir atvirkščiai: kartais kuriamos veiklos klasės, kaip papildomi moduliai jau esamoms klasėms, todėl šiuo atveju pateikimo puslapiai irgi nėra kuriami. Konfigūracijų keitimas (XML failų redakcija pagal pakeitimus) būtų neatliekamas vienu vieninteliu atveju – tada, kai kuriamos tik verslo klasės, kurios nėra naudojamos puslapiuose, o savo darbo rezultatus sistemai pateikia pasinaudodamos jau sukurtomis projekto verslo klasėmis, atliekančiomis ir duomenų pateikimo atvaizdavimui procesus.



3.10 pav. Tapestry karkaso plėtimo veiksmų seka

Konfigūracinių failų valdymas ir veiksmų aprašymas nėra sudėtingas Tapestry sistemoje. Šis karkasas yra turi pranašumą prieš kitas karkasines technologijas, nes leidžia naudoti daug konfigūracinių failų, kurių naudojimą kontroliuoja vartotojo inicializuotas konfigūracijų kontrolieris. Taip skaidant į atskirus gabalus bendrą sistemos karkasą, žymiai palengvėja sistemos tobulinimas bei kūrimas, nes tada konfigūracija kapojama pagal veiklos sritis ir lengviau surasti ieškomas sąveikas. Didelių sistemų sukurti ir aprašyti sąveikos failai būna labai ilgi, todėl tampa sunkiau perprantami. Visos Tapestry konfigūracijos aprašomos su dvejomis žymėmis: *servlet* ir *servlet mapping*. Pirmojo tipo žymės aprašo servletą, jo šakninę klasę, o *Servlet mapping* žymė naudojama objektų sąveikoms apibrėžti. Žemiau pateiktas fragmentas iš konferencijų darbo automatizavimo prototipo konfigūracinio failo, kuris naudojamas kiekvieno lygio vartotojo pradinės aplinkos inicializacijai.

```

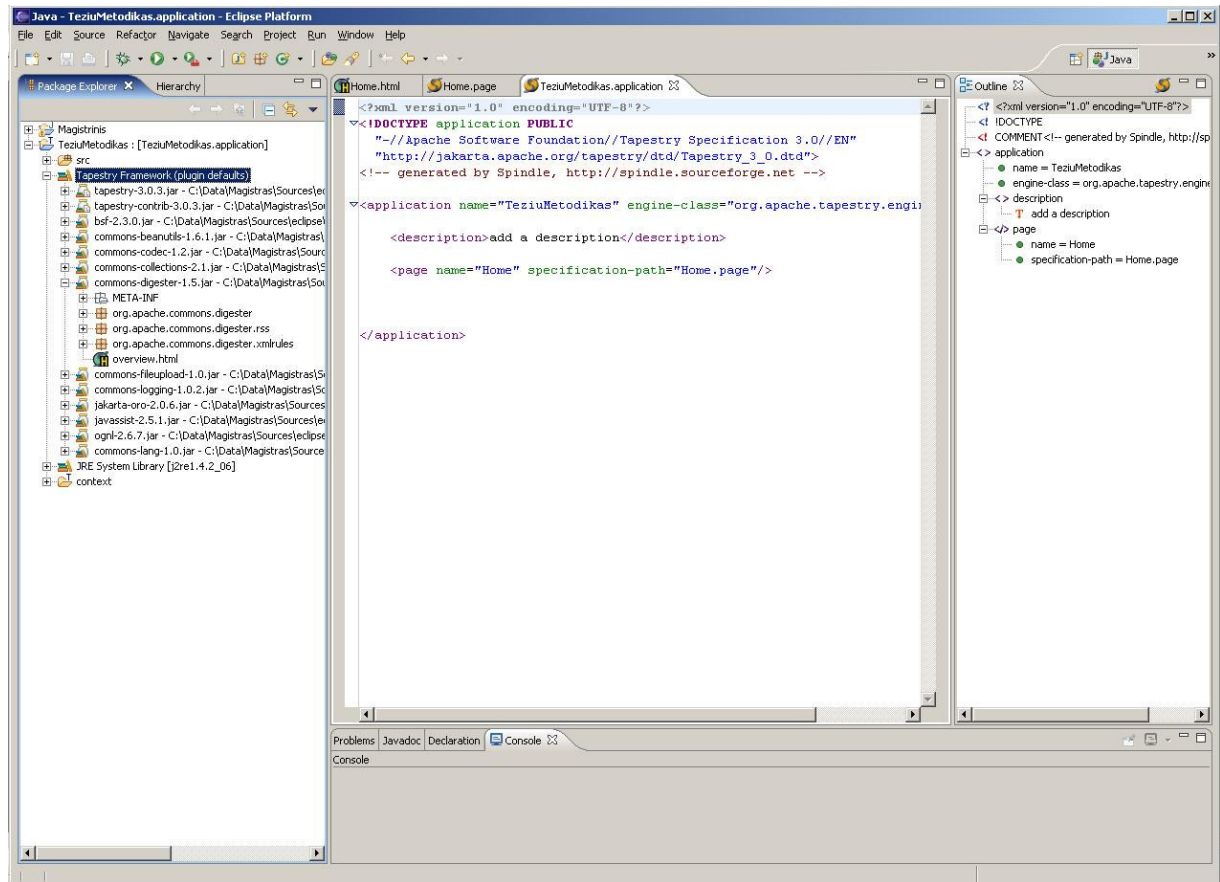
- <filter-mapping>
  <filter-name>redirect</filter-name>
  <url-pattern>/</url-pattern>
</filter-mapping>
- <servlet>
  <servlet-name>Anonimas</servlet-name>
  <servlet-class>org.apache.tapestry.Konferencija.PublicRoot</servlet-class>
  <load-on-startup>/BendrasPuslapis</load-on-startup>
</servlet>
- <servlet-mapping>
  <servlet-name>Anonimas</servlet-name>
  <url-pattern>/Konferencija</url-pattern>
</servlet-mapping>
- <servlet>
  <servlet-name>Autorius</servlet-name>
  <servlet-class>org.apache.tapestry.AutoriusRoot</servlet-class>
  <load-on-startup>AutoriusPuslapis</load-on-startup>
</servlet>
- <servlet-mapping>
  <servlet-name>Autorius</servlet-name>
  <url-pattern>/AutoriusAplinka</url-pattern>
</servlet-mapping>
- <servlet>
  <servlet-name>Recenzentas</servlet-name>
  <servlet-class>org.apache.tapestry.Konferencija.RecenzentasRoot</servlet-class>
  <load-on-startup>RecenzentasPuslapis</load-on-startup>
</servlet>
- <servlet-mapping>
  <servlet-name>Recenzentas</servlet-name>
  <url-pattern>/RecenzentasAplinka</url-pattern>
</servlet-mapping>
- <servlet>
  <servlet-name>Administratorius</servlet-name>
  <servlet-class>org.apache.tapestry.Konferencija.AdminRoot</servlet-class>
  <load-on-startup>AdministratoriusPuslapis</load-on-startup>
</servlet>
- <servlet-mapping>
  <servlet-name>Administratorius</servlet-name>
  <url-pattern>/AdministratoriusAplinka</url-pattern>
</servlet-mapping>

```

3.11 pav. XML konfigūracinio failo fragmentas

3.4.4 Spindle 3.0 priedas Eclipse programavimo aplinkai

Java Tapestry karkasinei sistemai egzistuoja įrankiai, palengvinantys internetinės programinės įrangos kūrimo darbą. Projekto kūrime dauguma etapų buvo realizuojami naudojant Elipse programų konstruktoriaus priedą *Spindle 3.0*. Tai programavimo aplinkos išplėtimas (ang. plugin), leidžiantis automatizuoti konfigūracinių failų kūrimą bei verslo klasių integraciją su interneto puslapiais. Ši programavimo aplinka taip pat tiesiogiai integruojasi su „Apache Tomcat“ serverių, todėl aplinkos konfigūracijoje galima nustatyti automatinį projektų pakeitimų sinchronizavimą su programine įranga, jau įdiegta Apache Tomcat serveryje. Tai sutaupo daug laiko, nes nereikia rūpintis integracija, versijavimu ir diegimu. Eclipse (www.eclipse.org) kūrimo aplinka pasirinkta dėl jos atviro kodo koncepcijos, šiai programavimo aplinkai yra sukurta daug išplėtimų, tame tarpe ir išplėtimų karkasinėms sistemoms kurti.



3.12 pav. Eclipse programavimo aplinka

Auksčiau pateiktame programos vaizdo lange matome, išplėtimo *Spindle 3.0* teikiamas automatizavimo galimybes sistemos kūrėjui. Kairėje atvaizduojami projekto resursai, paprasta naršyti po projekto objektus, konfigūracijas, aplinka praneša apie konfigūracijų kūrimo klaidas bei dažniausiai nurodo jų kilmės priežastis, taip pat egzistuoja patogus techninės dokumentacijos generatorius, kuriantis *javadocs* failus. Patogus dalykas – Tapestry žyniai, automatizuojantys naujų komponentų kūrimą ir bei jų integraciją.

Ši nemokama aplinka yra tinkamas įrankis Tapestry karkaso programinės įrangos projektams kurti, nes aplinkos teikiamos galimybės pilnai patenki automatizavimo poreikius šio tipo programų kūrimo.

4. METODIKOS TAIKYMAS KONFERENCIJŲ PROGRAMINĖS ĮRANGOS PROTOTIPUI SUKURTI

Šiame skyriuje aprašoma konferencijų darbo automatizavimo sistemos prototipo realizacija, sistemos reikalavimai, apibrėžiama bendra projekto veiklos sfera. Atliekama sukurtos sistemos architektūrinė analizė įvairiais aspektais, detalizuojami šie sistemos kūrimo pjūviai:

- Panaudojimo atvejai. Aprašomi sistemos aktoriai ir jų atliekami veiksmai.
- Loginis vaizdas. Pateikiama dalykinės srities klasių diagrama, paketų, objektų sąveikos.
- Proceso vaizdas. Atvaizduojamos objektų scenarijų ir bendradarbiavimo diagramos.
- Realizacijos vaizdas. Pateikiama programinių sistemos komponentų diagrama.
- Duomenų vaizdas. Pateikiama duomenų bazės schema.

Taip pat yra nagrinėjamas glaudus reikalavimų ir architektūrinių vaizdų aprašymas, susijimas, svarbiausios realizacijos metu sudarytos diagramos.

4.1 Konferencijų sistemos panaudojimo atvejai

4.1.1 Panaudojimo atvejų vaizdas



4.1 pav. Panaudojimo atvejų vaizdas.

4.1.2 Panaudojimo atvejų aprašymas

Žemiau pateikiamose lentelėse yra aprašomi visi konferencijų sistemos prototipo

panaudojimo atvejai, detalizuojant jų veikimo sąlygas bei rezultatai.

1. PANAUDOJIMO ATVEJIS: Paskelbti konferenciją

Vartotojas/Aktorius: Sistemos administratorius

Aprašas: Sukuriama sistemoje nauja konferencijos skiltis bei duomenų bazėje sukuriamas naujas įrašas

Prieš sąlyga: Autorizacijos tikrinimas; Aktorius turi turėti administratoriaus teises

Sužadinimo sąlyga: Paspauštas sukūrimo mygtukas

Po-sąlyga: Sukurtas naujas konferencijos objektas ir atvertas naujai sukurtos konferencijos administravimo langas

2. PANAUDOJIMO ATVEJIS: Skirti konferencijos administratorių

Vartotojas/Aktorius Sistemos administratorius

Aprašas: Apima procesą, kurio metu sistemos administratorius paskiria asmenį administruoti tam tikrą konferenciją bei suteikia jam tokias teises sistemoje.

Prieš sąlyga: Autorizacijos tikrinimas. Aktorius turi administratoriaus teises, konferencijos objektas sukurtas.

Sužadinimo sąlyga: Sužadinama konferencijos administratoriaus kūrimo funkcija.

Po-sąlyga: Atveriamas speciali aplinka. Paskirtas konferencijos administratorius.

3. PANAUDOJIMO ATVEJIS: Tvarkyti istoriją

Vartotojas/Aktorius: Sistemos administratorius

Aprašas: Apima procesą, kurio metu sistemos administratorius sutvarko duomenų bazę bei užrakina įvykusias konferencijas

Prieš sąlyga: Autorizacijos tikrinimas. Konferencijos objektas sukurtas. Konferencija baigta

Sužadinimo sąlyga: Sužadinama konferencijos tvarkymo funkcija.

Po-sąlyga: Atveriamas patvirtinimo langas. Konferencijos redaguoti neįmanoma.

4. PANAUDOJIMO ATVEJIS: Nustatyti datas

Vartotojas/Aktorius: Sistemos administratorius

Aprašas: Nustatomos svarbios datos, kurioms atėjus bus siunčiami sisteminiai pranešimai visiems užsiregistravusiems bei kitiems suinteresuotiems asmenims.

Prieš sąlyga: Autorizacijos tikrinimas. Konferencijos objektas sukurtas. Datos nėra nustatytos.

Sužadavimo sąlyga: Sužadinama konferencijos datų nustatymo funkcija.

Po-sąlyga: Datos išsaugomos bei perduodamos sistemos mechanizmui, kuris atėjus laikui siųs pranešimus.

5. PANAUDOJIMO ATVEJIS: Šalinti administratorius

Vartotojas/Aktorius: Sistemos administratorius

Aprašas: Apima procesą, kurio metu sistemos administratorius panaikina administravimo teisę iš ją turėjusio vartotojo.

Prieš sąlyga: Autorizacijos tikrinimas. Konferencija užrakinta. Vartotojas turi konferencijos administratoriaus teises.

Sužadavimo sąlyga: Sužadinama konferencijos administratoriaus šalinimo funkcija.

Po-sąlyga: Vartotojui suteikiamos paprasto vartotojo (autoriaus) teisės.

6. PANAUDOJIMO ATVEJIS: Skirti recenzentus

Vartotojas/Aktorius: Konferencijos administratorius

Aprašas: Apima procesą, kurio metu konferencijos administratorius paskiria recenzentus atitinkamiems darbams recenzuoti.

Prieš sąlyga: Autorizacijos tikrinimas. Darbas nėra priskirtas kitam recenzentui.

Sužadavimo sąlyga: Sužadinama recenzento skyrimo funkcija.

Po-sąlyga: Atidaroma speciali aplinka bei pasirenkami sisteminiai pranešimai, kurie bus išsiusti.

7. PANAUDOJIMO ATVEJIS: Šalinti recenzentus

Vartotojas/Aktorius: Konferencijos administratorius

Aprašas: Apima procesą, kurio metu konferencijos administratorius pašalina recenzavimo teisę pasirinktam recenzentui.

Prieš sąlyga: Autorizacijos tikrinimas. Recenzentas parašė recenziją ir jis vis dar turi recenzavimo teisę

Sužadavimo sąlyga: Sužadinama recenzento šalinimo funkcija.

Po-sąlyga: Recenzentui priskiriamos paprasto vartotojo (autoriaus) teisės.

8. PANAUDOJIMO ATVEJIS: Skelbti temas

Vartotojas/Aktorius: Konferencijos administratorius

Aprašas: Paskelbiamos temos, kuriomis vyks konferencija.

Prieš sąlyga: Autorizacijos tikrinimas. Tema nėra paskelbta.

Sužadinimo sąlyga: Sužadinama temos skelbimo funkcija.

Po-sąlyga: Tema įtraukta. Padaromas įrašas duomenų bazėje.

9. PANAUDOJIMO ATVEJIS: Skelbti konferencijos pradžia

Vartotojas/Aktorius: Konferencijos administratorius

Aprašas: Paskelbiama konferencijos pradžia bei pradeda veikti automatinis sistemos valdymas

Prieš sąlyga: Autorizacijos tikrinimas. Konferencija nėra pradėta. Yra sukurtos temos.

Sužadinimo sąlyga: Sužadinama konferencijos pradžios skelbimo funkcija.

Po-sąlyga: Siunčiami sisteminiai pranešimai. Konferencija atidaryta. Daromi įrašai duomenų bazėje.

10. PANAUDOJIMO ATVEJIS: Skelbti konferencijos pabaigą

Vartotojas/Aktorius: Konferencijos administratorius

Aprašas: Paskelbiama konferencijos pabaiga.

Prieš sąlyga: Autorizacijos tikrinimas. Konferencija nėra užbaigta.

Sužadinimo sąlyga: Sužadinama konferencijos pabaigos skelbimo funkcija.

Po-sąlyga: Siunčiami sisteminiai pranešimai. Konferencija uždaryta. Daromi įrašai duomenų bazėje. Redagavimas leidžiamas tik sistemos administratoriui kol konferencija neužrakinta.

11. PANAUDOJIMO ATVEJIS: Siųsti info recenzentams

Vartotojas/Aktorius: Konferencijos administratorius

Aprašas: Apima procesą, kurio metu nustatomi sisteminiai pranešimai, kurie bus siunčiami automatizuotai.

Prieš sąlyga: Autorizacijos tikrinimas. Konferencija paskelbta bei neuždaryta.

Sužadinimo sąlyga: Sužadinama informacijos siuntimo recenzentams funkcija.

Po-sąlyga: Pranešimus siunčia sistema pagal administratoriaus nustatytas taisykles.

12. PANAUDOJIMO ATVEJIS: Registruotis

Vartotojas/Aktorius: Autorius

Aprašas: Apima procesą, kurio metu vartotojas užregistruoja savo duomenis sistemoje.

Prieš sąlyga: Gavo konferencijos arba sistemos administratoriaus siųstą sutikimą registracijai. Tokio vartotojo dar nėra. Autorizacijos tikrinimas.

Sužadinimo sąlyga: Jungiamasi į sistemą administratoriaus duotu adresu ir duomenimis

Po-sąlyga: Sistemoje užregistruojamas naujas vartotojas. Atnaujinami duomenys.

13. PANAUDOJIMO ATVEJIS: Išsiregistruoti

Vartotojas/Aktorius: Autorius

Aprašas: Apima procesą, kurio metu vartotojas atsisako sisteminių pranešimų.

Prieš sąlyga: Autorizacijos tikrinimas. Pranešimai buvo siunčiami.

Sužadinimo sąlyga: Sužadinama sisteminių pranešimų atsisakymo funkcija.

Po-sąlyga: Pranešimai šiam vartotojui nebus siunčiami. Vartotojas neištrintas.

14. PANAUDOJIMO ATVEJIS: Įkelti straipsnį

Vartotojas/Aktorius: Autorius

Aprašas: Apima procesą, kurio metu vartotojas įkrauna į sistemą parašytą straipsnį.

Prieš sąlyga: Autorizacijos tikrinimas. Straipsnis dar nėra įkeltas. Straipsnio formatas atitinka keliamus reikalavimus.

Sužadinimo sąlyga: Sužadinama straipsnio įkėlimo funkcija.

Po-sąlyga: Straipsnis įkeltas bei įtrauktas į duomenų bazę – redagavimas galimas.

15. PANAUDOJIMO ATVEJIS: Pasiūlyti temą

Vartotojas/Aktorius: Autorius

Aprašas: Straipsnio pavadinimo ir santraukos bei raktinių žodžių įvedimas.

Prieš sąlyga: Autorizacijos tikrinimas. Tokia tema dar nebuvo pasiūlyta.

Sužadinimo sąlyga: Sužadinama temos siūlymo funkcija.

Po-sąlyga: Siunčiamas pranešimas sistemos administratoriui.

16. PANAUDOJIMO ATVEJIS: Pildyti anketą

Vartotojas/Aktorius: Autorius

Aprašas: Apima procesą, kurio metu vartotojas pildo anketą.

Prieš sąlyga: Autorizacijos tikrinimas. Vartotojas dar nėra pildęs anketos tokiam straipsniui.

Sužadinimo sąlyga: Sužadinama anketos pildymo funkcija.

Po-sąlyga: Siunčiamas pranešimas konferencijos administratoriui apie užpildytą anketą.

17. PANAUDOJIMO ATVEJIS: Peržiūrėti straipsnius

Vartotojas/Aktorius: Recenzentas

Aprašas: Apima procesą, kurio metu recenzentas gali peržiūrėti sistemoje esančius straipsnius.

Prieš sąlyga: Autorizacijos tikrinimas. Pasirinktas pasijungimo tipas: Recenzentas

Sužadinimo sąlyga: Sužadinama straipsnių peržiūros funkcija.

Po-sąlyga: -

18. PANAUDOJIMO ATVEJIS: Pasiūlyti temą

Vartotojas/Aktorius: Recenzentas

Aprašas: Straipsnio pavadinimo ir santraukos bei raktinių žodžių įvedimas.

Prieš sąlyga: Autorizacijos tikrinimas. Tokia tema dar nebuvo pasiūlyta. Pasirinktas pasijungimo tipas: Autorius.

Sužadinimo sąlyga: Sužadinama temos siūlymo funkcija.

Po-sąlyga: Siunčiamas pranešimas sistemos administratoriui.

19. PANAUDOJIMO ATVEJIS: Pildyti recenziją

Vartotojas/Aktorius: Recenzentas

Aprašas: Apima procesą, kurio metu rašoma recenzija konkrečiam darbui.

Prieš sąlyga: Autorizacijos tikrinimas. Recenzija neužrakinta. Pasirinktas pasijungimo tipas: Recenzentas

Sužadinimo sąlyga: Sužadinama recenzijos pildymo funkcija.

Po-sąlyga: Jei recenzija nebaigta – tik išsaugojimas, o jei užbaigta – recenzija užrakinama bei siunčiamas pranešimas konferencijos administratoriui.

20. PANAUDOJIMO ATVEJIS: Registruotis

Vartotojas/Aktorius: Recenzentas

Aprašas: Apima procesą, kurio metu vartotojas užregistruoja savo duomenis sistemoje.

Prieš sąlyga: Gavo konferencijos arba sistemos administratoriaus siųstą sutikimą registracijai. Tokio vartotojo dar nėra. Autorizacijos tikrinimas.

Sužadinimo sąlyga: Jungiamasi į sistemą administratoriaus duotu adresu ir duomenimis.

Po-sąlyga: Sistemoje užregistruojamas naujas vartotojas. Atnaujinami duomenys.

21. PANAUDOJIMO ATVEJIS: Išsiregistruoti

Vartotojas/Aktorius: Recenzentas

Aprašas: Apima procesą, kurio metu vartotojas atsisako sisteminių pranešimų.

Prieš sąlyga: Autorizacijos tikrinimas. Pranešimai buvo siunčiami.

Sužadinimo sąlyga: Sužadinama sisteminių pranešimų atsisakymo funkcija.

Po-sąlyga: Pranešimai šiam vartotojui nebus siunčiami. Vartotojas neištrintas.

22. PANAUDOJIMO ATVEJIS: Peržiūrėti konferencijas

Vartotojas/Aktorius: Anoniminis sistemos vartotojas

Aprašas: Apima procesą, kurio metu anoniminis sistemos vartotojas turi galimybę peržiūrėti jau baigtas konferencijas bei juose buvusius straipsnius.

Prieš sąlyga: Autorizacijos tikrinimas.

Sužadinimo sąlyga: Sužadinama konferencijų peržiūros funkcija.

Po-sąlyga: -

23. PANAUDOJIMO ATVEJIS: Siūlyti temą

Vartotojas/Aktorius: Anoniminis sistemos vartotojas

Aprašas: Pasiūloma tema būsimom konferencijoms.

Prieš sąlyga: Autorizacijos tikrinimas. Tokia tema dar nebuvo pasiūlyta.

Sužadinimo sąlyga: Sužadinama temos siūlymo funkcija.

Po-sąlyga: Siunčiamas pranešimas sistemos administratoriui.

24. PANAUDOJIMO ATVEJIS: Įtraukti straipsnį į programą

Vartotojas/Aktorius: Konferencijos administratorius

Aprašas: Apima procesą, kurio metu į konferencijos programą įtraukiamas darbas, jeigu jo recenzija yra teigiama.

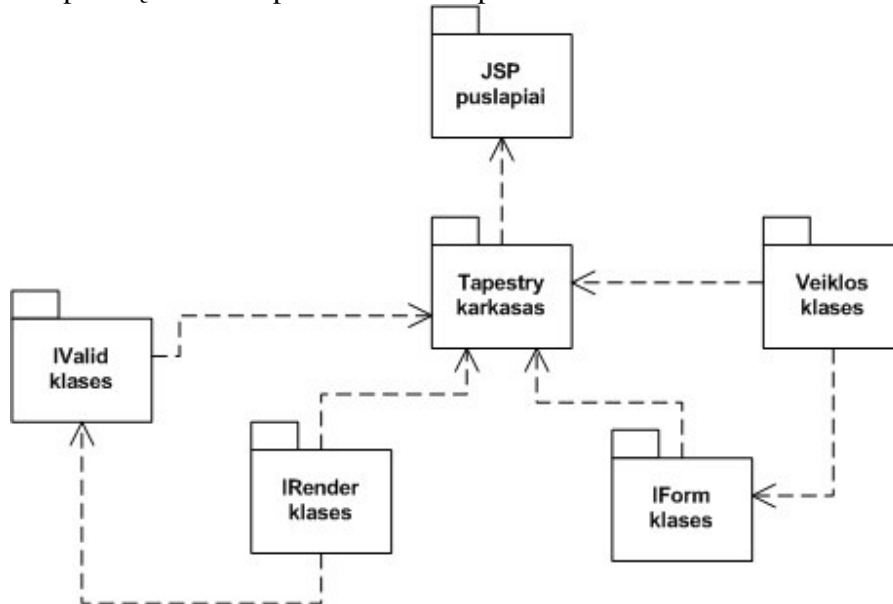
Prieš sąlyga: Autorizacijos tikrinimas. Konferencija paskelbta bei neuždaryta. Darbas recenzuotas.

Sužadinimo sąlyga: Sužadinama informacijos straipsnio įtraukimo funkcija.

Po-sąlyga: Straipsnis įtrauktas į konferencijos programą.

4.2 Sistemos paketų struktūra

Sistemos paketų struktūra pavaizduota 4.2 paveiksle.

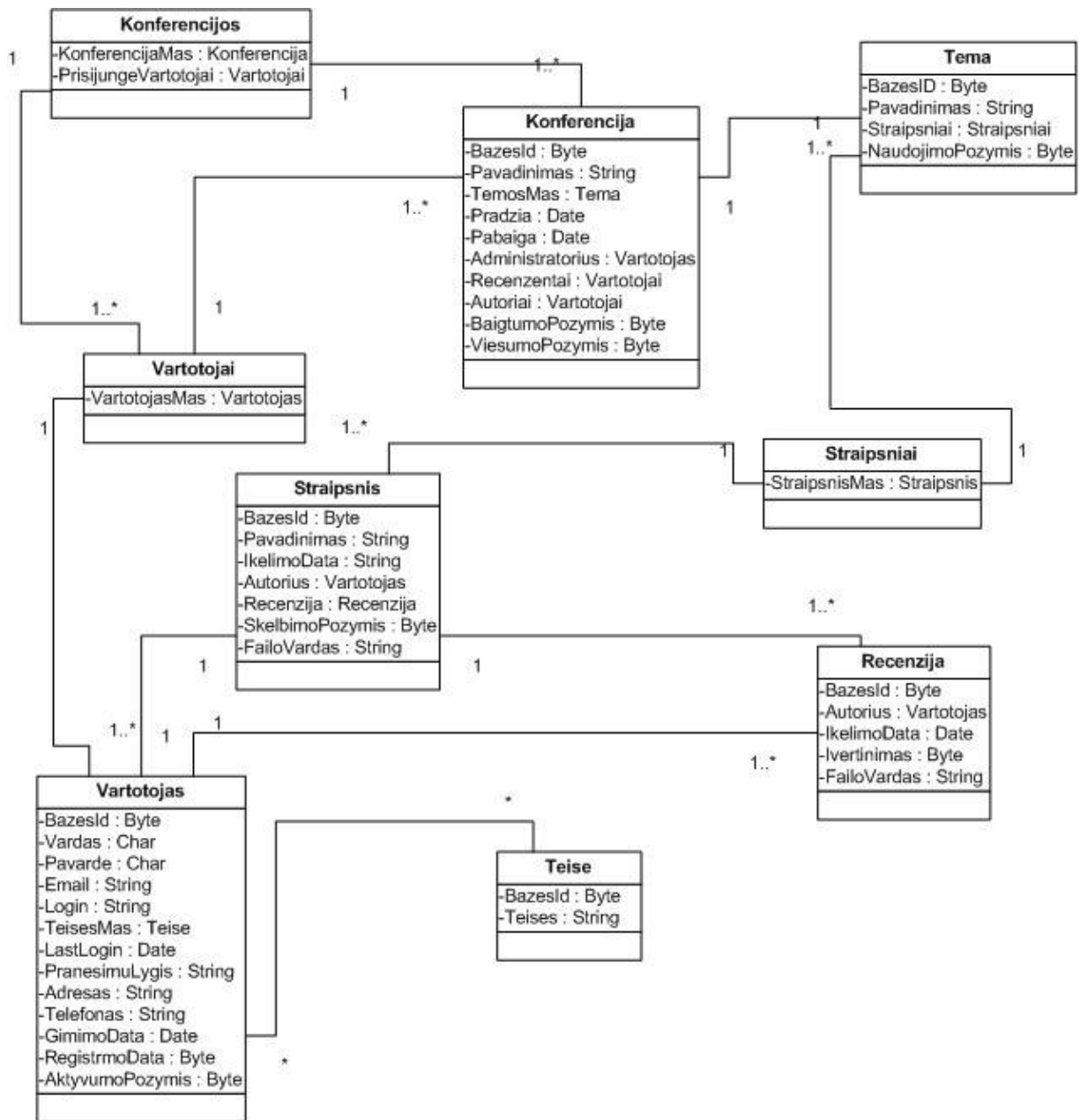


4.2 pav. Sistemos paketų sąveikos diagrama

Sistemos klasės yra suskirstytos į paketus:

- „Tapestry framework“ IRender klasės, vykdomos atlikus tam tikrą veiksmą.
- „Tapestry framework“ IForm klasės, vykdomos įvedant, redaguojant duomenis į formą.
- Tapestry framework IValid klasės, vykdomos aptikus kažkokią klaidą.
- JSP puslapiai – vartotojo interfeisas, atliekantys vaizdavimą ir apimantys minimalų kiekį kodo.
- Verslo klasių paketas teikiantis visą logiką apimančio duomenų apdorojimo paslaugas, ryšį su duomenų baze.

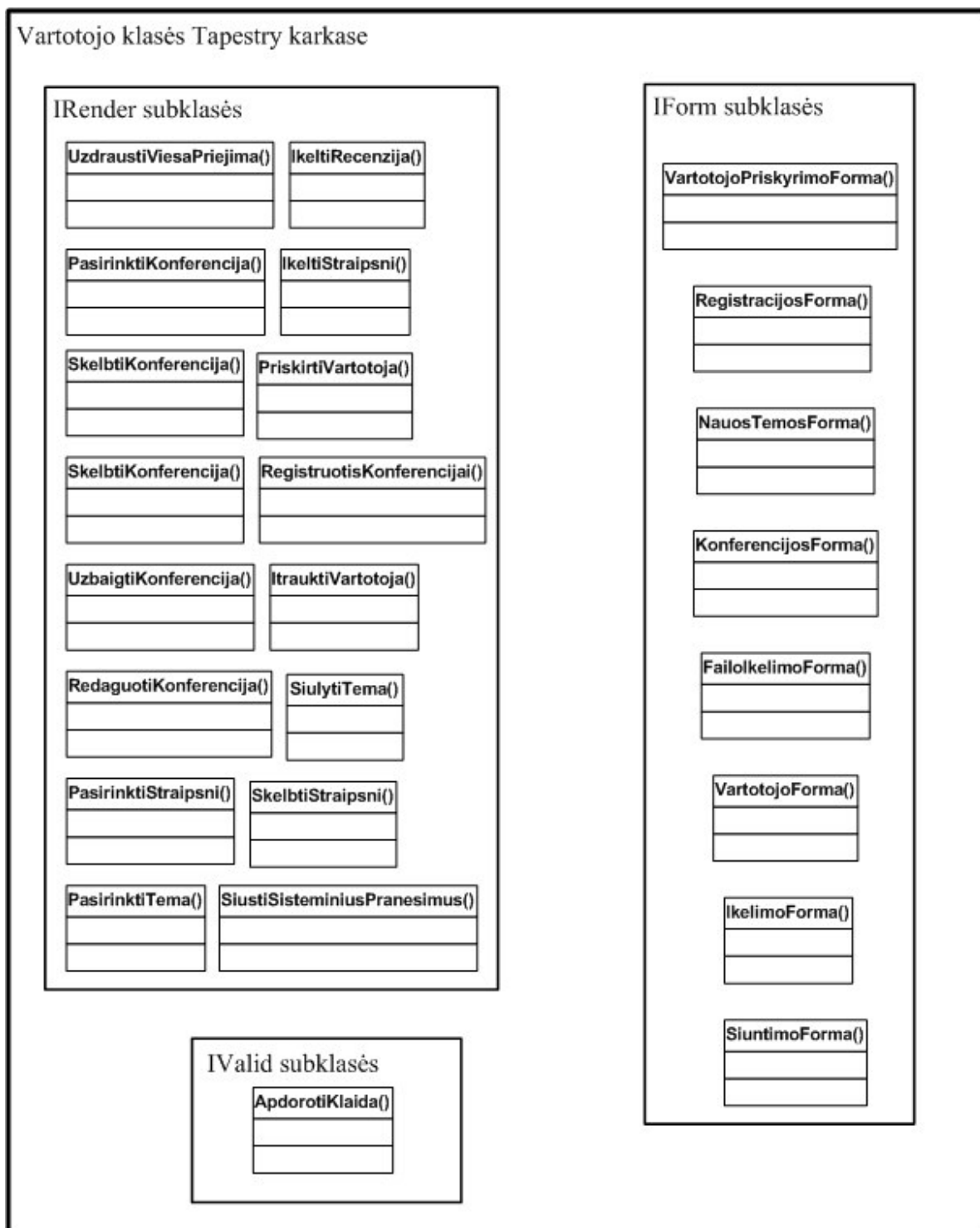
Sistemos architektūra sukurta pagal Tapestry karkasą. Klasės suskirstytos pagal tai, kaip jos bus apdorojamos pagal naudojamojo šablono taisykles. Verslo klasių pakete talpinamos visą logiką apimančios bei pagrindines paslaugas atliekančios duomenų apdorojimo klasės (4.3 pav.).



4.3 pav. Sistemos verslo klasių paketas

Pagal Tapestry karkasą sukurtos valdymo ir pateikties klasės pateiktos 4.4 paveiksle.

Vartotojo klasės Tapestry karkasė

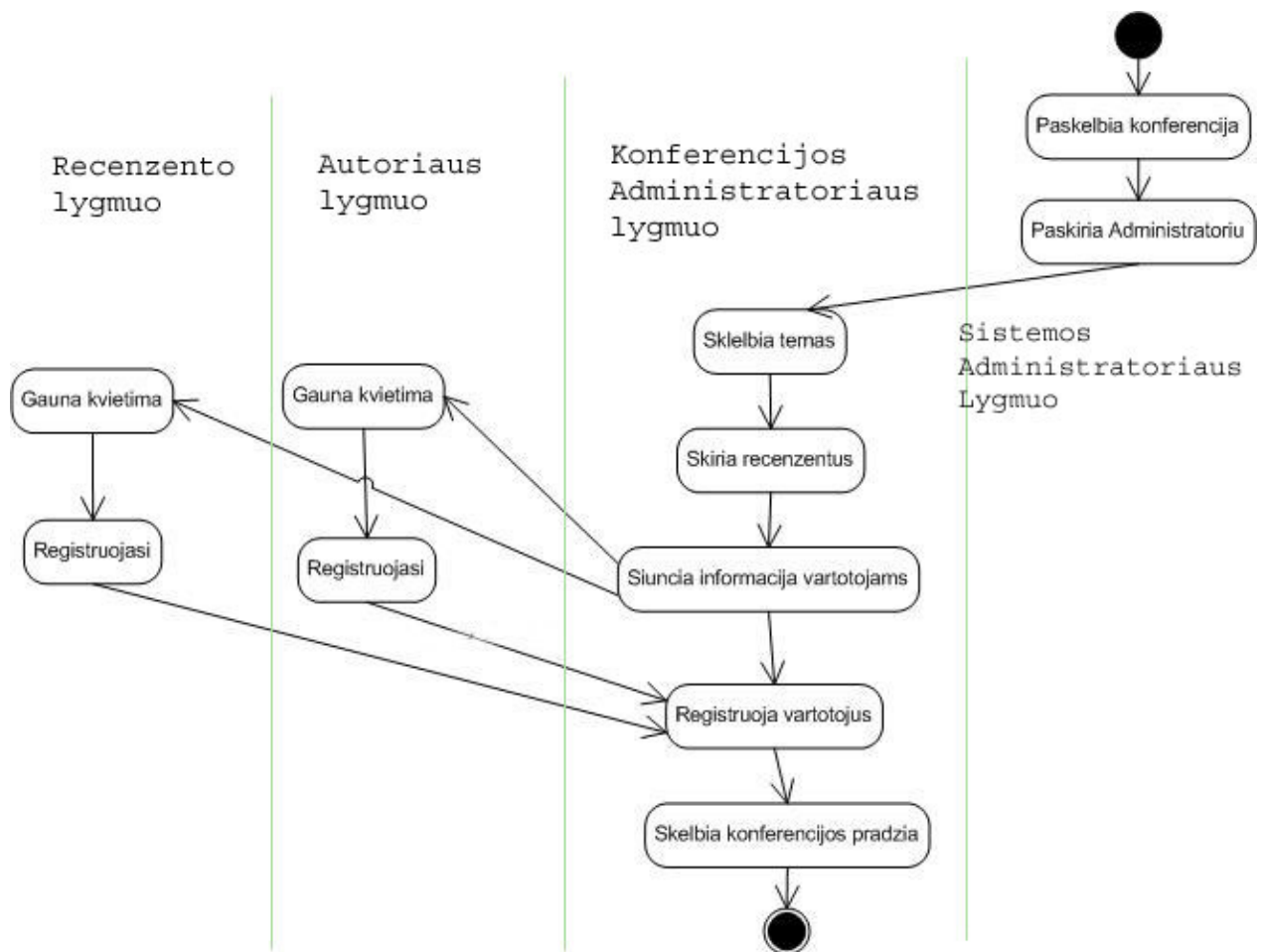


4.4 pav. Vartotojo paslaugų paketas pagal „Tapestry“ karkasą

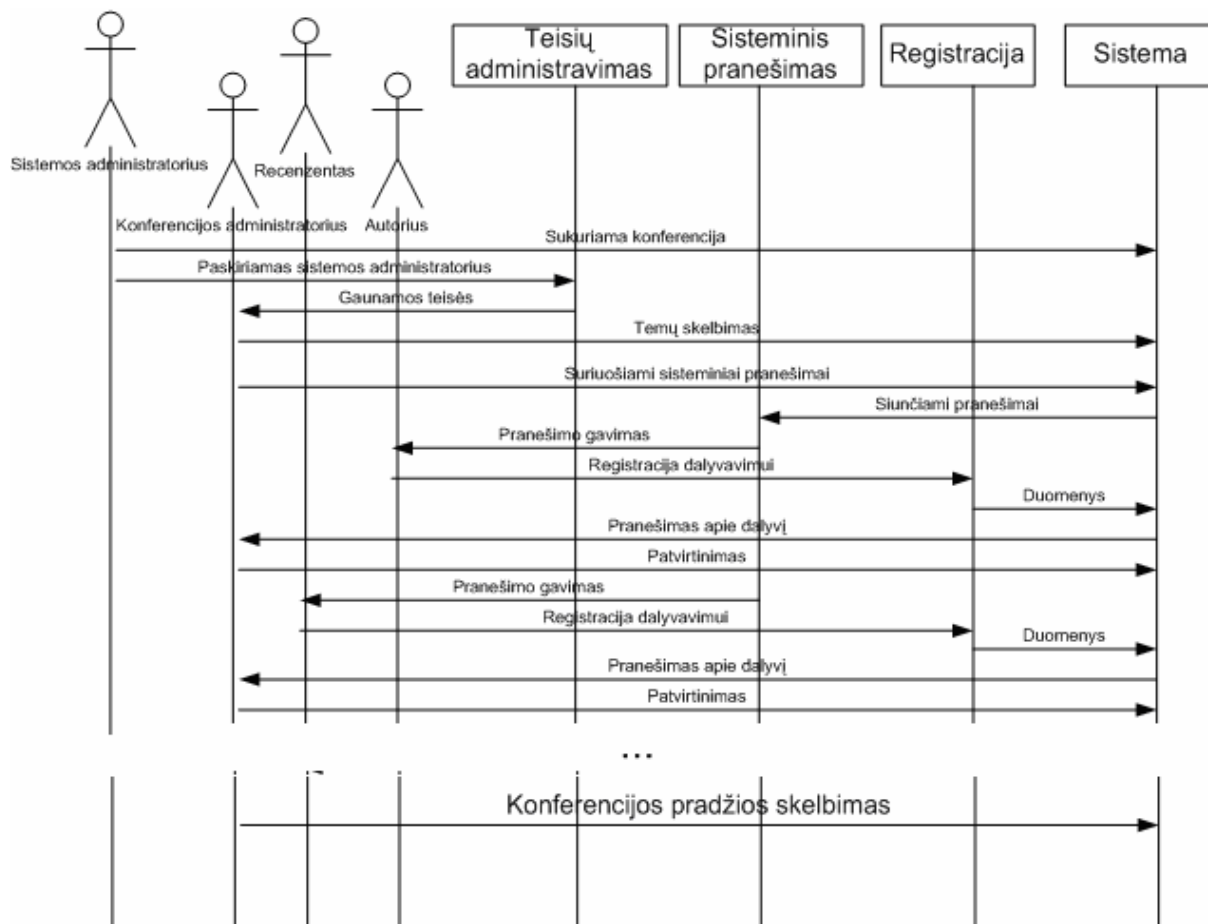
4.3 Konferencijų sistemoje vykdomi procesai

4.3.1 Naujos konferencijos skelbimas

Kiekvieną naują konferenciją inicijuoja sistemos administratorius, nuroydamas konferencijos inicializavimo duomenis bei jos administratorių, kuris ir tęsia visą konferencijos vykdymo procesą iki pat jo užbaigimo.



4.5 pav. Naujos konferencijos paskelbimo procesas



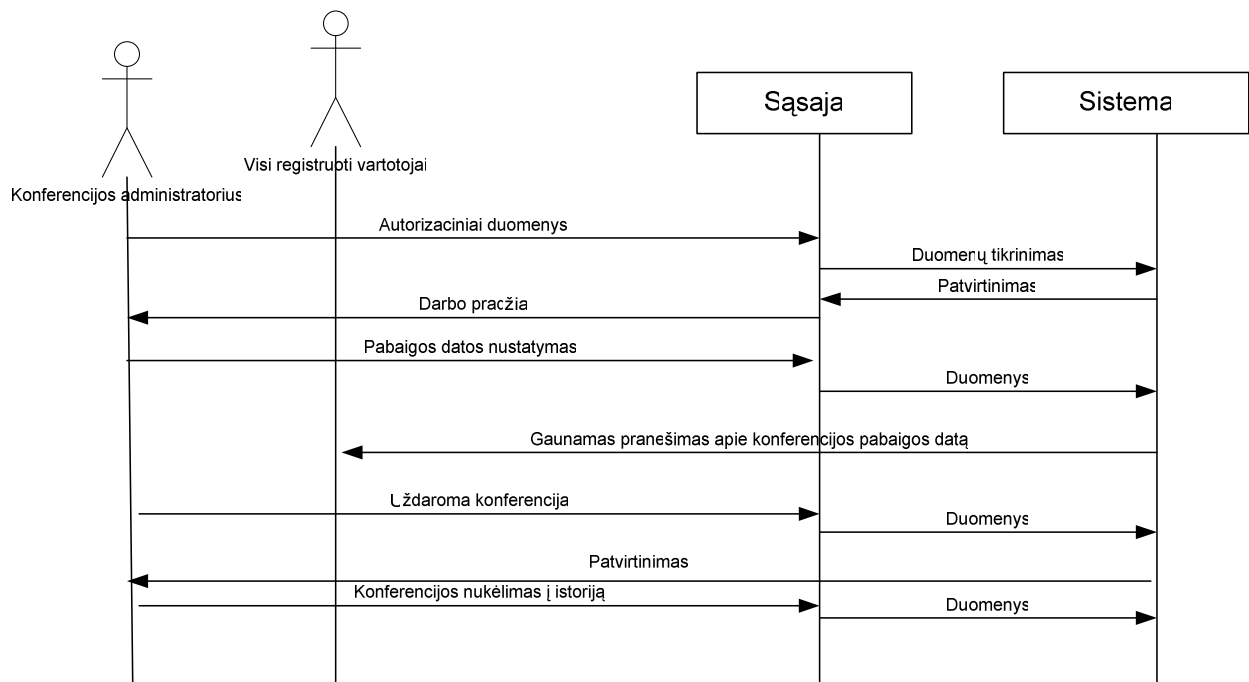
4.6 pav. Naujos konferencijos paskelbimo veiksmų sekos diagrama

4.3.2 Konferencijos užbaigimo procesas

Konferencijos užbaigimo procesą vykdo jos administratorius. Prieš konferencijos užbaigimą, visi jos dokumentai sutvarkomi, nustatoma prieiga prie konferencijos metu sukauptos informacijos, ji padaroma privati, prieinama tik tam tikro rango vartotojams, arba vieša, prieinama visiems.



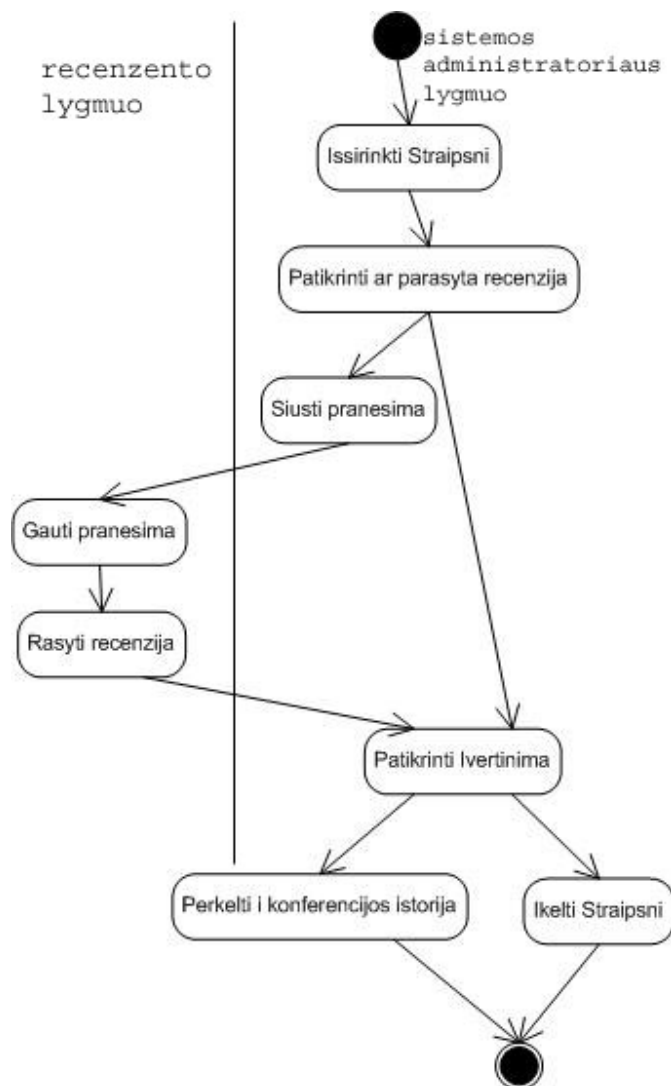
4.7 pav. Konferencijos užbaigimo procesas



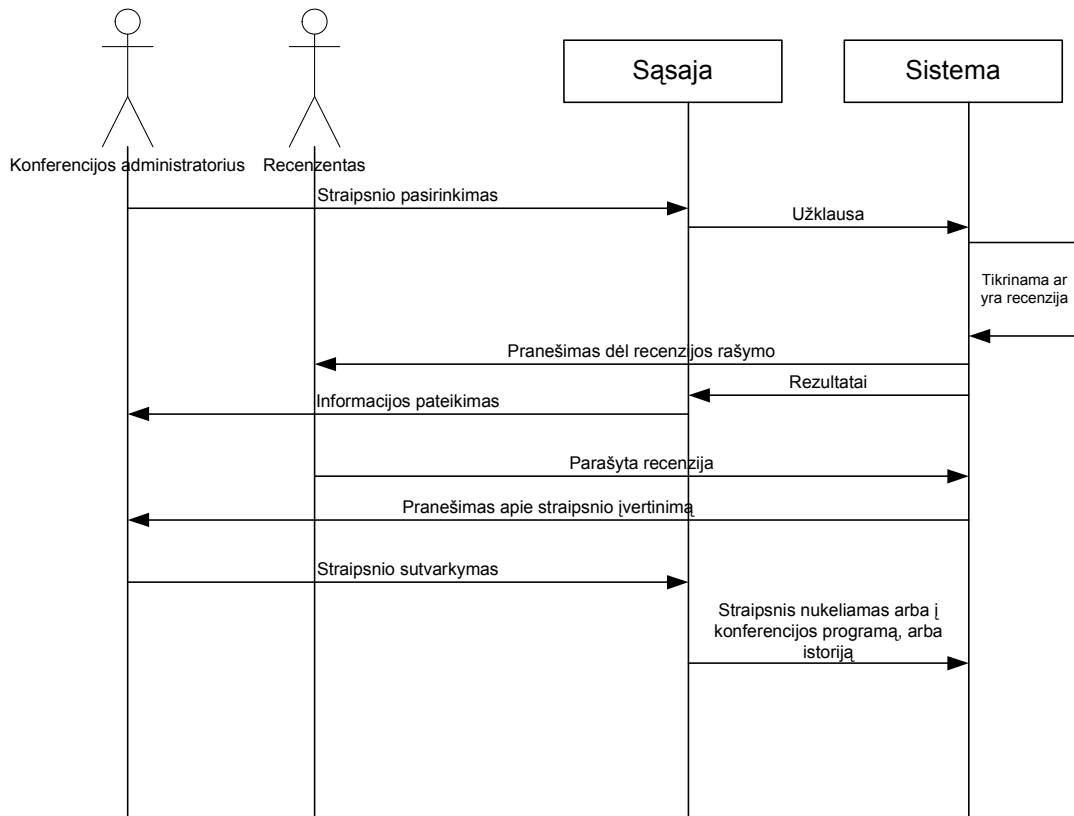
4.8 pav. Konferencijos užbaigimo veiksmų sekos diagrama

4.3.3 Straipsnio įtraukimo į konferenciją procesas

Konferencijos administratorius mato sąrašą atsiustų straipsnių, kurie turi būti recenzijuojami ir įtraukti į konferenciją, jei įvertinimas tinkamas, arba patalpinami į istoriją prie straipsnių, kurie nebuvo priimti į konferenciją. Kiekvienam atsiustam straipsniui konferencijos administratorius paskiria recenzentą, kuris gauna pranešimą, jog jam yra paskirtas straipsnis, kuriam jis turi parašyti recenziją. Recenzijos įvertinimas nulemia tolesnį straipsnio kelią – jis arba talpinamas prie konferencijos straipsnių, arba tiesiog įkeliamas į konferencijos istoriją kaip netinkamas konferencijai.



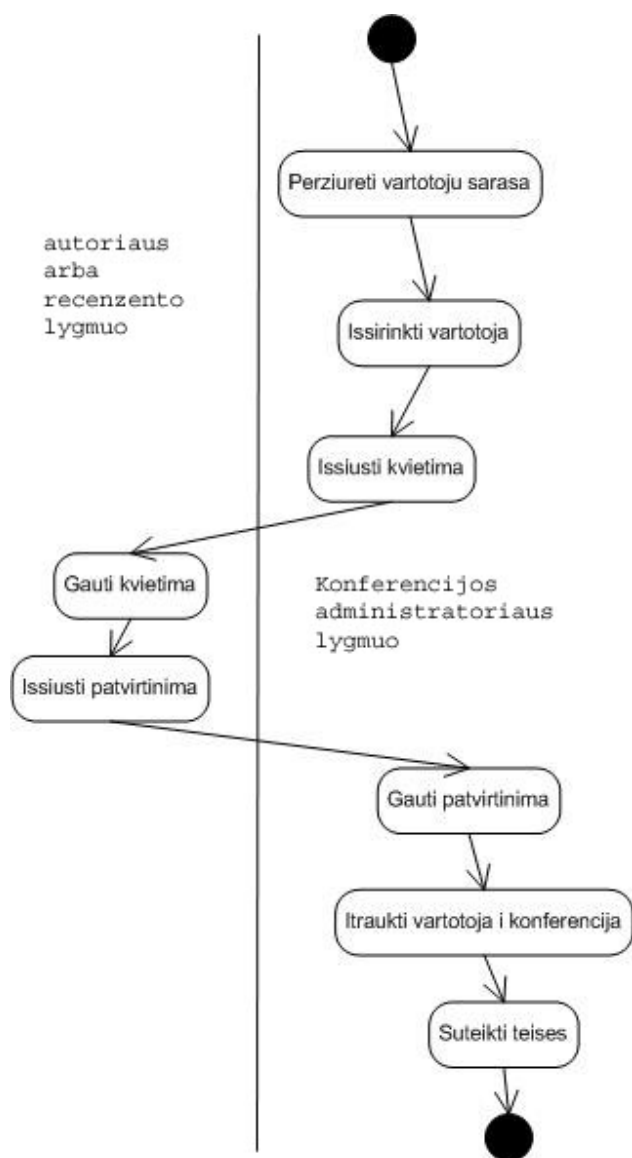
4.9 pav. Straipsnio įkėlimo į konferenciją procesas



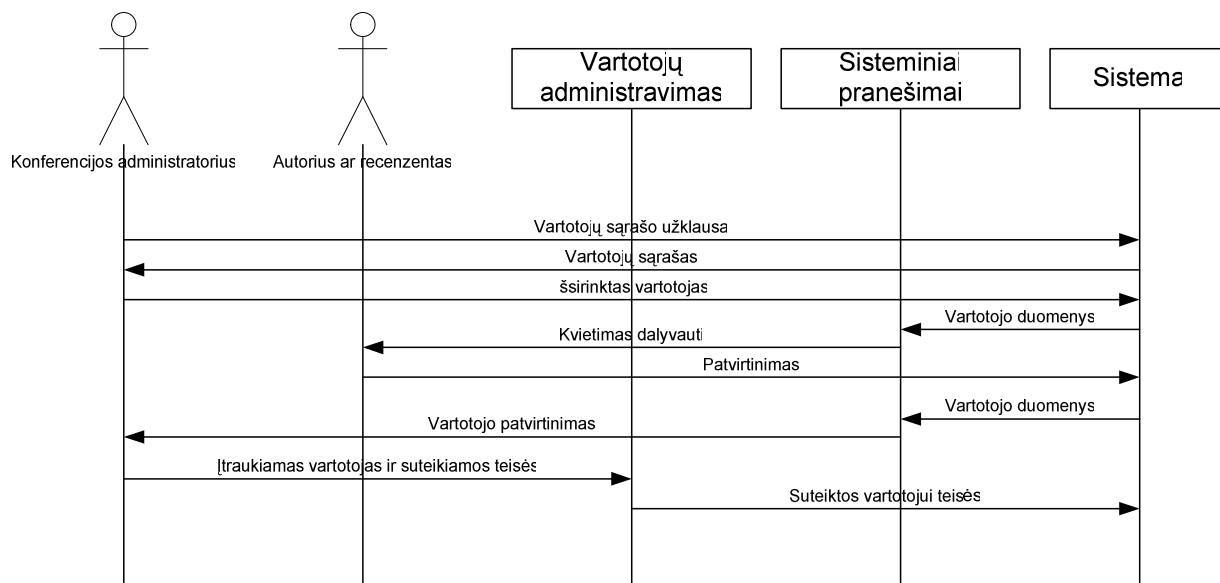
4.10 pav. Straipsnio įkėlimo į konferenciją veiksmų sekos diagrama

4.3.4 Autorių ir recenzentų įtraukimo į konferenciją procesas

Pradedant naują konferenciją, visi vartotojai gauna pakvietimus joje dalyvauti elektroniniu paštu. Konferencijos administratorius mato sąrašą žmonių, kurie patvirtino savo norą dalyvauti, ir iš šio sąrašo išrinkinėja asmenis, kurie jam atrodo tinkami dalyviai šiai konferencijai. Šie dalyviai gauna informaciją elektroniniu paštu, kad jie yra išrinkti dalyvauti konferencijoje, todėl turi dar kartą patvirtinti savo būsimą dalyvavimą.



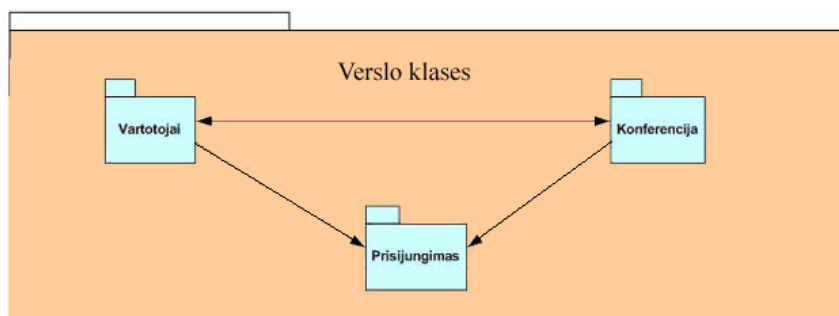
4.11 pav. Dalyvių įtraukimo į konferenciją procesas



4.12 pav. Dalyvių įtraukimo į konferenciją veiksmų sekos diagrama

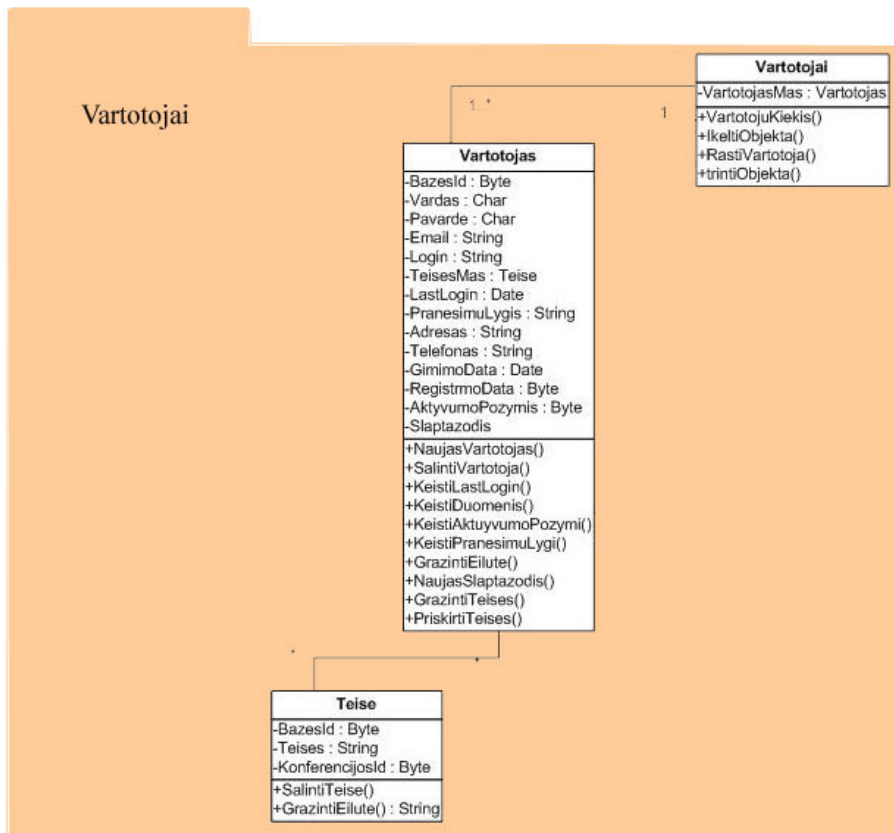
4.4 Klasių detalizavimas

Visos verslo klasės priklauso vienam paketui. Jame inicijuotos klasės duomenų mainams su vartotoju, duomenų baze bei kitomis sąsajomis, paketas skaidomas į žemesnio lygio paketus, kuriuose realizuotos nustatyti funkcionalumai.



4.13 pav. Verslo klasių paketas

4.4.1 Vartotojų paketas



4.14 pav. Paketo „Vartotojas“ struktūra

Šis sistemos komponentas skirtas sujungti klases vartotojų informacijai apdoroti ir kitoms funkcijoms atlikti. Paketo klasės atsakingos už visų veiksmų, susijusių su vartotojų administravimu, pateikimą ir atvaizdavimą. Komponentas sąveikauja su paketais „Konferencija“ ir „Prisijungimas“.

Paketo klasių detalizacija:

➤ Klasė Vartotojas

Paskirtis: naujo vartotojo sukūrimui, jo parametrų inicializavimui ir redagavimui.

Atsakomybės: Klasė turi teisę kurti, keisti, gauti duomenis apie vartotoją. Taip pat turi saugoti savyje vartotojo teises.

Apribojimai: Privalo būti inicijuotas ryšis su duomenų baze.

Sąveikavimas: Klasė sąveikauja su klasėmis Vartotojai.Vartotojai, Vartotojai.Teise, Recenzija, Konferencija.Straipsnis, Prisijungimas.Rsys,

Prisijungimas.CustomException.

Sąsaja/eksportas: Per metodą NaujasVartotojas().

➤ **Klasė Vartotojai**

Paskirtis: Vartotojų sąrašo generavimas pagal užklausos parametrus, kuri ir inicijuoja šio objekto sukūrimą.

Atsakomybės: Klasė yra atsakinga už sistemos vartotojų sąrašo, kuris formuojamas pagal objektą inicijuojančio modulio pateiktus parametrus.

Apribojimai: Turi būti sukurtas objektas, atliekantis duomenų mainus tarp duomenų bazių valdymo sistemos ir programos.

Sąveikavimas: Klasė sąveikauja su klasėmis Vartotojai.Vartotojas bei Konferencija.Konferencija, Prisijungimas.Rsys, Prisijungimas.CustomException.

Sąsaja/eksportas: Atliekamas per klasę Vartotojas.

➤ **Klasė Teisė**

Paskirtis: Klasė, organizuojanti vartotojo teisių tvarkymą bei saugojimą.

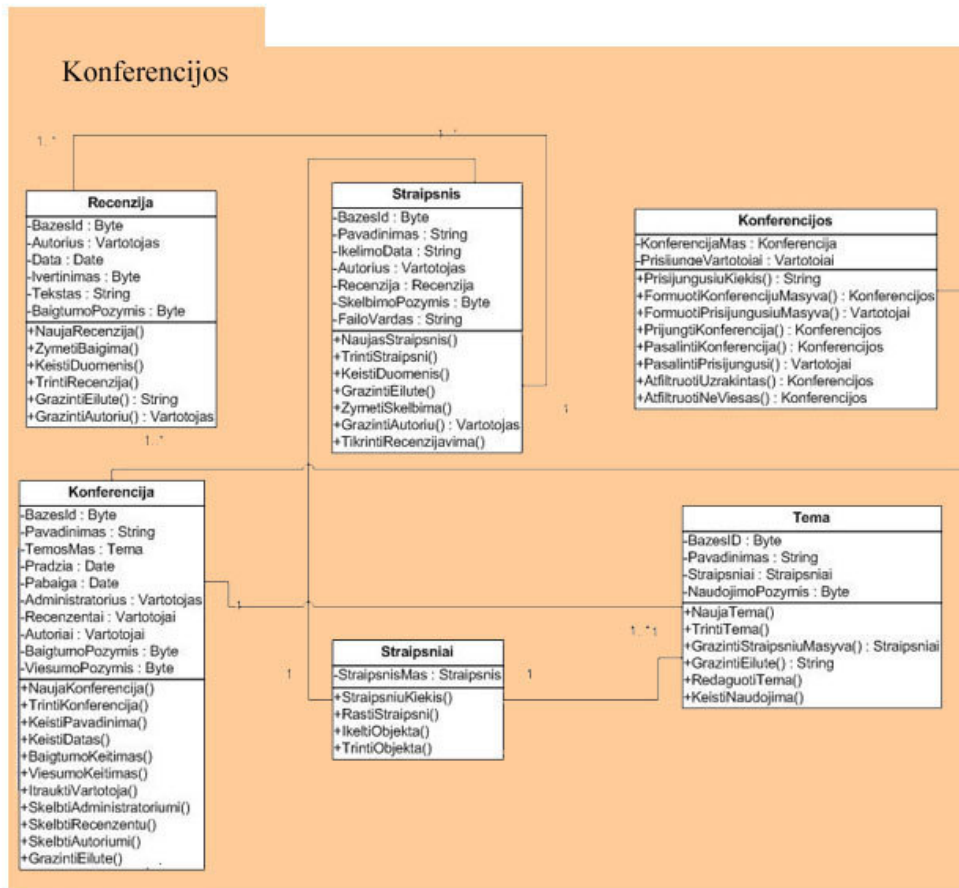
Atsakomybės: Klasė yra atsakinga už teisingą vartotojo teisių pateikimą bei vartotojui neprieinamų konferencijų uždraudimą.

Apribojimai: Turi būti sukurtas objektas, atliekantis duomenų mainus tarp duomenų bazių valdymo sistemos ir programos.

Sąveikavimas: Ši klasė sąveikauja su klase Vartotojai.Vartotojas, Prisijungimas.Rsys, Prisijungimas.CustomException.

Sąsaja/eksportas: Klasės eksportuoja savo duomenis pasinaudodama metodu ReturnRights().

4.4.2 Konferencijų paketas



4.15 pav. Paketo „Konferencijos“ struktūra

Komponentas apjungia klases, skirtas darbui su konferencijomis, straipsniais bei recenzijomis. Paketo klasės atsakingos už veiksmų, susijusių su konferencijų valdymu, pateikimą, sąveikauja su paketu „Vartotojai“ ir „Prisijungimas“ objektais.

➤ Klasė Konferencijos

Paskirtis: Skirta visų konferencijų tvarkymui, valdymui. duomenų mainų organizavimui tarp duomenų bazės ir valdymo sistemos.

Atsakomybės: Visos sistemos teisingas veikimas, tai pagrindinis sistemos objektas, kuriantis ir inicijuojantis kitus sistemos komponentus.

Apribojimai: Apribojimų neturi, nes tai pagrindinis sistemos objektas.

Sąveikavimas: Sąveikauja su klasėmis dauguma objektų bei paketu.

Sąsaja/eksportas: Klasė organizuos duomenų bei darbo rezultatų eksporto į duomenų bazių valdymo sistemą atlikimą bei atliks informacijos persiuntimą į ją Tapestry karkaso klases, kuris užtikrins vartotojui pateikiamų JSP puslapių generaciją.

➤ **Klasė Konferencija**

Paskirtis: Klasė skirta vienos konferencijos tvarkymui ir valdymui.

Atsakomybės: Atsakinga už konferencijos sukūrimą, pilną administravimą, įskaitant dalyvių įtraukimą, būsenų keitimą, straipsnių dėjimą bei kitos funkcijos.

Apribojimai: Redagavimo galimybė suteikta, kol konferencijos baigtumo požymis nėra teigiamas. Konferencijos negali dubliuotis.

Sąveikavimas: Klasė sąveikauja su kitais objektais Konferencija.Konferencijos, Vartotojai.Vartotojai, Konferencija.Tema, Prisijungimas.Rsys, Prisijungimas.CustomException.

Sąsaja/eksportas: Per metodą NaujaKonferencija().

➤ **Klasė Tema**

Paskirtis: Klasė skirta konferencijos temų valdymui.

Atsakomybės: Teisingas temų pateikimas, tvarkymas, valdymas tiek konkrečioje konferencijoje, tiek bendroje sistemoje.

Apribojimai: Turi būti sukurtas objektas, atliekantis duomenų mainus tarp duomenų bazių valdymo sistemos ir programos.

Sąveikavimas: Klasė sąveikauja su klasėmis Konferencija.Straipsniai ir Konferencija.Konferencija. Prisijungimas.Rsys, Prisijungimas.CustomException.

Sąsaja/eksportas: Klasė organizuos duomenų bei darbo rezultatų eksporto į duomenų bazių valdymo sistemą atlikimą bei atliks informacijos persiuntimą į ją Tapestry karkaso klases, kuris užtikrins vartotojui pateikiamų JSP puslapių generaciją.

➤ **Klasė Straipsnis**

Paskirtis: Klasė skirta straipsnio valdymo uždaviniams atlikti.

Atsakomybės: Atsakinga už naujo straipsnio sukūrimą, redagavimą, požymių keitimą, recenzijų suteikimą.

Sąveikavimas: Klasė sąveikauja su klasėmis Konferencija.Straipsniai, Konferencija.Recenzija, Vartotojai.Vartotojas, Prisijungimas.Rsys, Prisijungimas.CustomException.

Sąsaja/eksportas: Per metodą NaujasStraipsnis().

➤ **Klasė Straipsniai**

Paskirtis: Klasė valdanti įkeltų straipsnių masyvą.

Atsakomybės: Atsakinga už straipsnių surinkimą pagal tam tikrą požymį ir užkrovimą į masyvą.

Sąveikavimas: Klasė sąveikauja su klasėmis Konferencija.Straipsnis, Konferencija.Tema, Prisijungimas.Rsys, Prisijungimas.CustomException.

Sąsaja/eksportas: Per klasę Straipsnis().

➤ **Klasė Recenzija**

Paskirtis: Klasė skirta recenzijų valdymui ir tvarkymui.

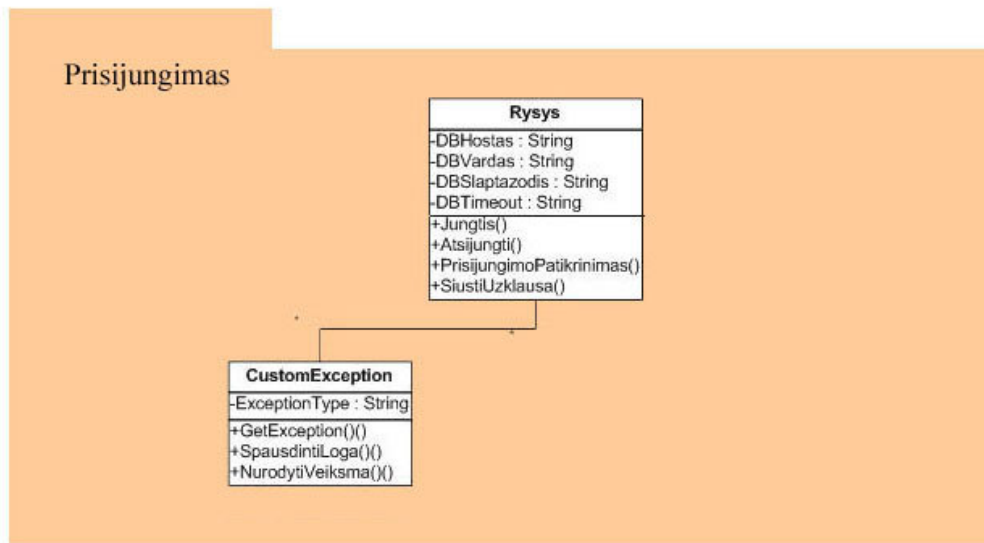
Atsakomybės: Atsakinga už teisingą naujų recenzijų įkėlimą, susiejimą su straipsniais, redagavimą bei kitus funkcionalumus, susijusius su recenzijomis.

Apribojimai: Nauja recenzija gali būti kuriama tik tuo atveju, jei straipsnis, kuriam kuriama recenzija, jos dar neturi.

Sąveikavimas: Klasė sąveikauja su klasėmis Konferencija.Straipsnis, Vartotojai.Vartotojas, Prisijungimas.Rsys, Prisijungimas.CustomException.

Sąsaja/eksportas: Per metodą NaujaRecenzija().

4.4.3 Prisijungimo paketas



4.16 pav. Paketo“Prisijungimas“ struktūra

Šis sistemos komponentas skirtas apjungti klases, atliekančias duomenų mainus tarp duomenų bazių valdymo sistemos ir programos. Paketo klasės yra naudojami visuose kituose paketuose, tačiau pats komponentas išmonių objektų nenaudoja.

➤ **Klasė CustomException**

Paskirtis: Klasė skirta klaidos aptikimui ir sekančio veiksmo parinkimui, įvykus klasėje apibrėžtai klaidos situacijai atliekant duomenų kaitą tarp programos ir duomenų bazės.

Atsakomybės: Klasė atsakinga už teisingą sistemos veikimą atsiradus klaidai vykstant duomenų mainams.

Apribojimai: Turi būti sukurtas objektas Rysys, kuris atlieka duomenų mainus tarp sistemos ir duomenų bazės.

Sąveikavimas: Klasė sąveikauja su klase Prisijungimas.Rysys.

Sąsaja/eksportas: Klasei aptikus klaidą, yra siunčiamas atitinkamas nurodymas į Tapestry framework‘ą numatyto veiksmo parinkimui.

➤ **Klasė Ryšys**

Paskirtis: Klasė skirta duomenų mainams tarp duomenų bazės ir valdymo sistemos.

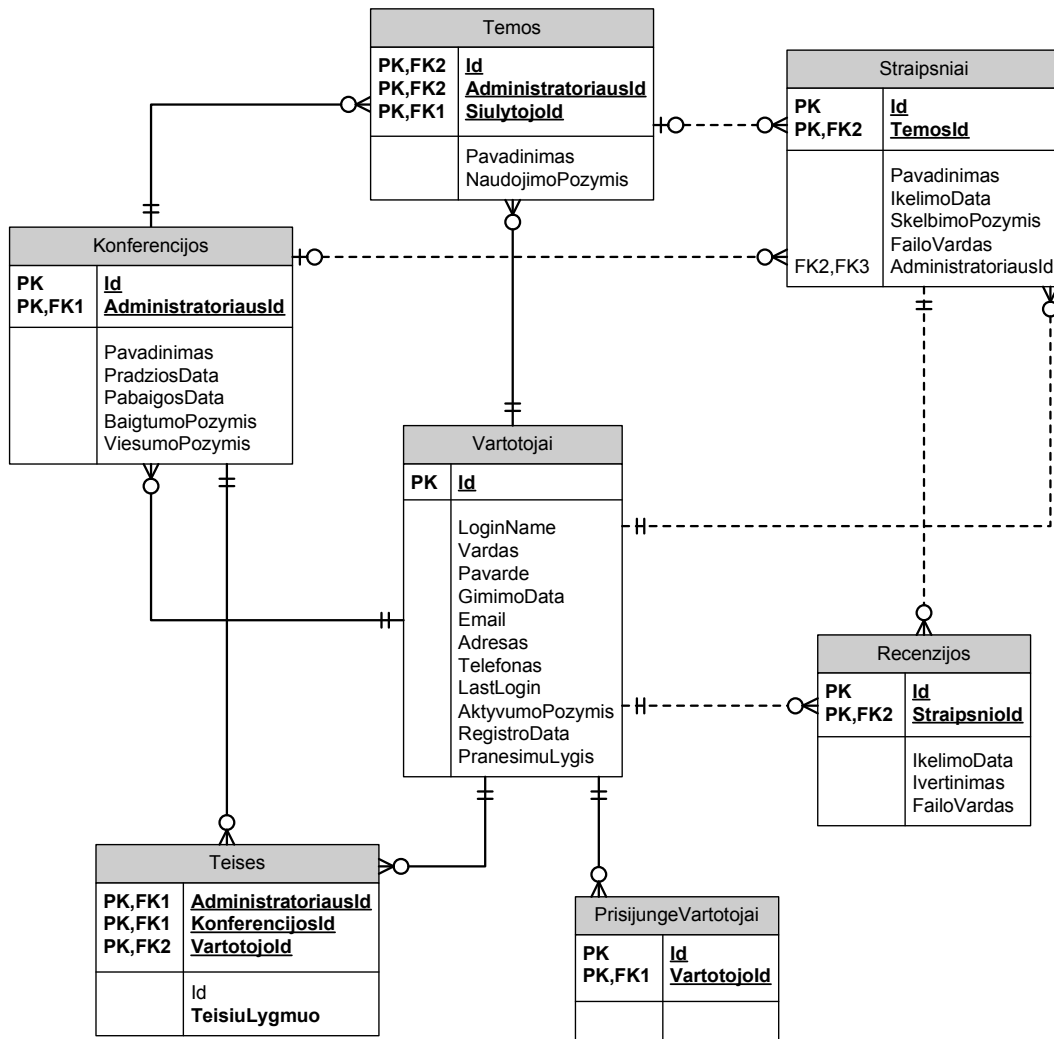
Atsakomybės: Klasė atsakinga už teisingą duomenų kaitą tarp sistemos bei duomenų bazės.

Sąveikavimas: Klasės sąveikauja su visomis pagrindinėmis sistemos klasėmis, nes tai yra duomenų kaitą valdanti klasė.

Sąsaja/eksportas: Duomenys eksportuojami į duomenų bazę.

4.5 Duomenų vaizdas

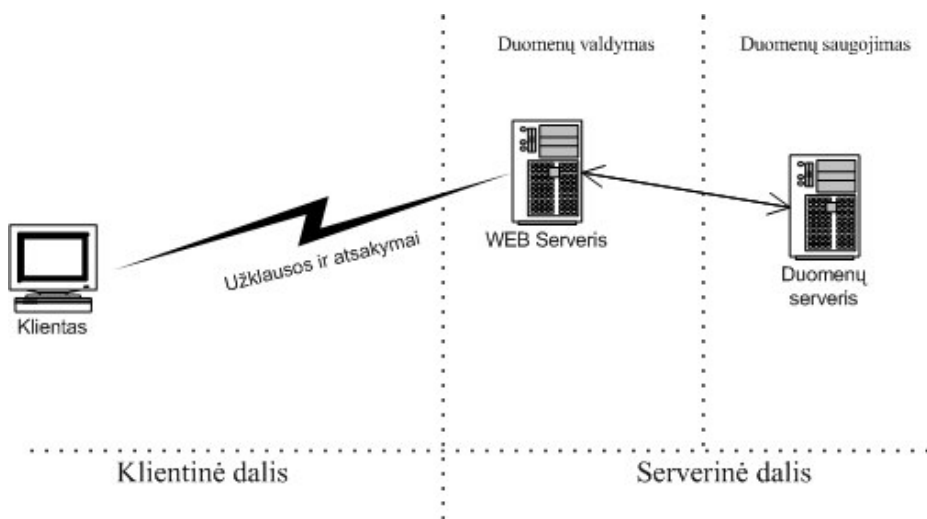
Žemiau esančioje schemoje pateikiamas duomenų vaizdas. Lentelėje *virtotojai*, saugoma vartotojų informacija, apimanti visus vartotojo parametrus apie administratorius, autorius, recenzentus. Informacija apie konferenciją saugoma lentelėje *konferencijos*, aprašant visus su konkrečia konferencija susijusius duomenis. Temos ir straipsniai turi savo atskiras lenteles, kiekviename iš jų turi po identifikacinį lauką, apibrėžiantį, kuriai konferencijai jie priklauso, kuris rodo į lentelę *konferencijos*. Kadangi sistema yra personalizuota, egzistuoja kelių prieigos lygių vartotojai, taip pat vartotojai pagal konferencijas gali įgyti skirtingas teises, todėl vartotojų prieigų galimybės fiksuojamos lentelėje Teisės.



4.17 pav. Duomenų modelis

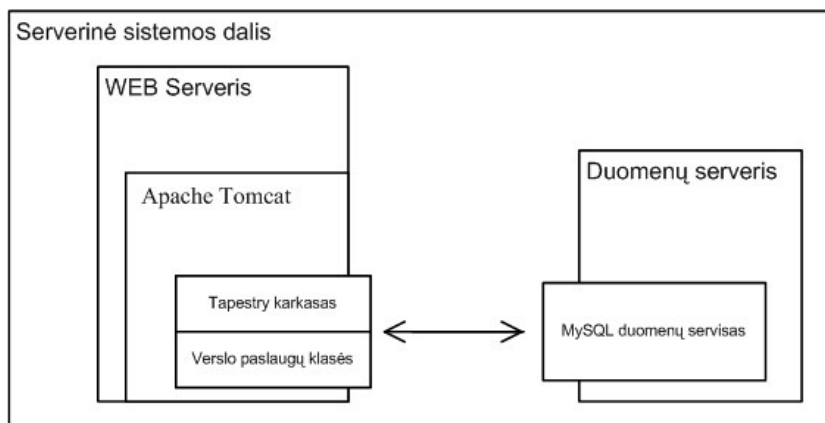
4.6 Komponentų ir diegimo diagramos

Projekto realizacija – tai kliento-serverio principu dirbanti programinė įranga. Klientinė dalis, kurią naudoja vartotojai savo naršyklių languose, nėra kuriama ir diegiam sistemos kūrėjų – tai standartinė programinė įranga, skirta interneto puslapių skaitymui. Kūrėjai programuoja serverio pusės projekto programinę įrangą, kurią mes taip pat galime skirstyti į dar du lygius: tai duomenų saugojimo lygis ir duomenų valdymo lygis. Duomenų valdymo lygyje yra visa mūsų programinė realizacija, visa interneto sistema, kuri kreipiasi į duomenų bazių valdymo sistemą ir atlieka duomenų mainus.



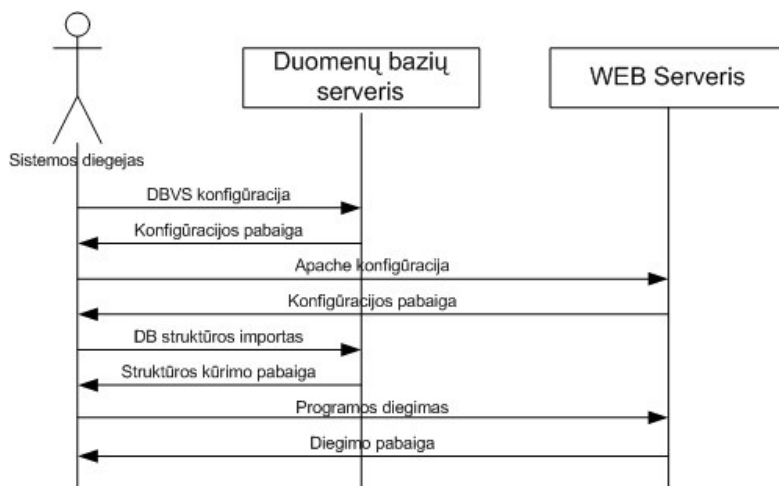
4.188 pav. Sistemos architektūra

Auksčiau sistema apibrėžia bendrus sistemos fizinės architektūros principus, tačiau jų nedetalizuoja jų. Todėl žemiau pateikta diagrama yra detalesnis serverio pusės architektūros vaizdas.



4.199 pav. Serverio pusės sistemos architektūra

Atlikus projekto realizaciją, vykdomas jo diegimas. Diegimo procesas šilo tipo programose nėra sudėtingas, nes daugumą diegimo metu iškylančių problemų „Jakarta Apache“ Tomcat kūrėjas išspėdė, įdiegdami automatizacijos įrankius savo naujesnėse versijose. Šie įrankiai ir atlieka visą konfigūracijų inicializavimą bei organizaciją, todėl vartotojui diegimo projekto paleidimo veiksmų seka nebėra sudėtinga. Esminiai projekto paleidimo sekos punktai (angl. checklist) pažymėti žemiau pateiktoje diagramoje.



4.20 pav. Sistemos diegimo žingsniai

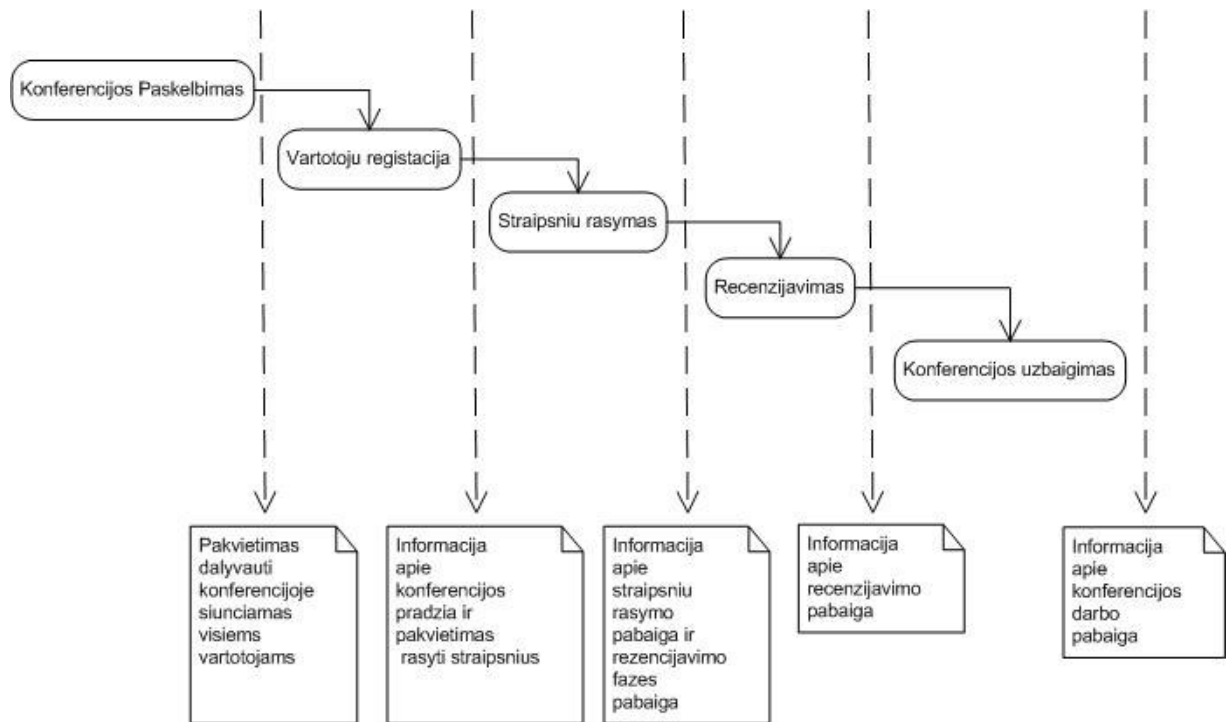
Kaip ir galime susidaryti bendrą vaizdą, diegimo procesas apsiriboja bendra sistemų konfigūracija, nes pačia programinės įrangos instaliacijos paruoša pasirūpina „Jakarta Tomcat“ serveris. Diegėjai patalpina archyvuotą sukurtą sistemą į Apache Tomcat šakninį katalogą su išplėtimu *.war, o sistema, aptikusį šį failą, jį automatiškai išpakuoja ir atlieka jo pasiekiamumo konfigūracijas.

5. SUKURTO PRODUKTO VEIKIMO, KOKYBĖS IR TAIKYMO GALIMYBIŲ ANALIZĖ

5.1 Vartotojų darbas su sistema

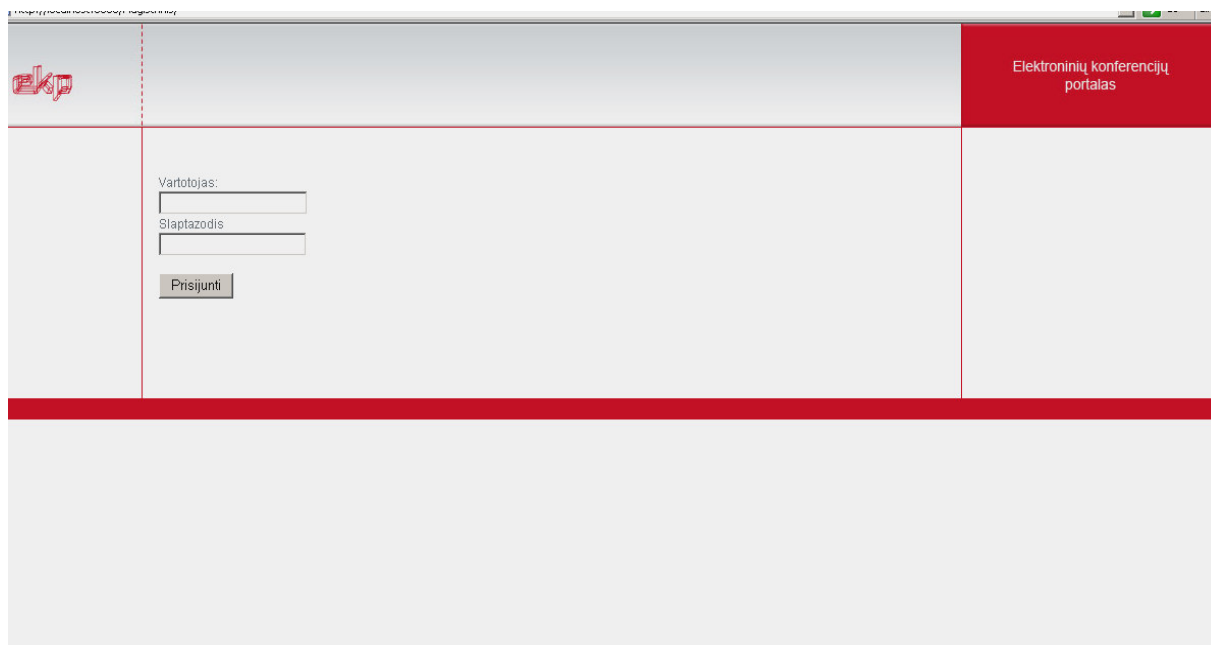
Sukurtame konferencijų organizavimo programinės įrangos prototipe realizuotos specifikacijoje numatytos funkcijos, išspręstas vartotojų teisių valdymas, realizuotas personalizuotas meniu, kurio pagalba vartotojas gali atlikti tik jam leistinus veiksmus.

Visas konferencijos procesas yra skaldomas į fazes (5.1 pav.), taip palengvinant konferencijos organizavimą. Kiekvienos fazės pradžioje vartotojai yra informuojami elektroniniu paštu apie veiksmus, kuriuos jie turi atlikti.



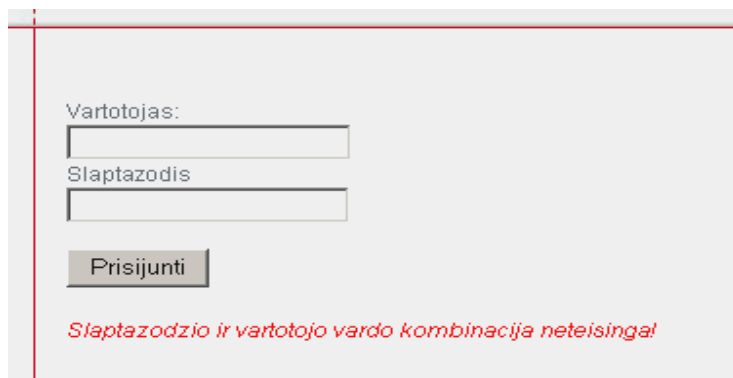
5.1 pav. Konferencijos eigos fazės

Kiekvienas vartotojas, prieš pradėdamas dirbti su sistema, privalo autorizuotis. Atliekant vartotojo registraciją, jam yra sukuriamas vartotojo vardas, o atsitiktinai sugeneruotas slaptažodis išsiunčiamas elektroniniu paštu. Pirmojo prisijungimo metu vartotojas turi pasikeisti slaptažodį ir vėlesnių prisijungimų metu naudoti naująjį, žinomą tik jam vienam.



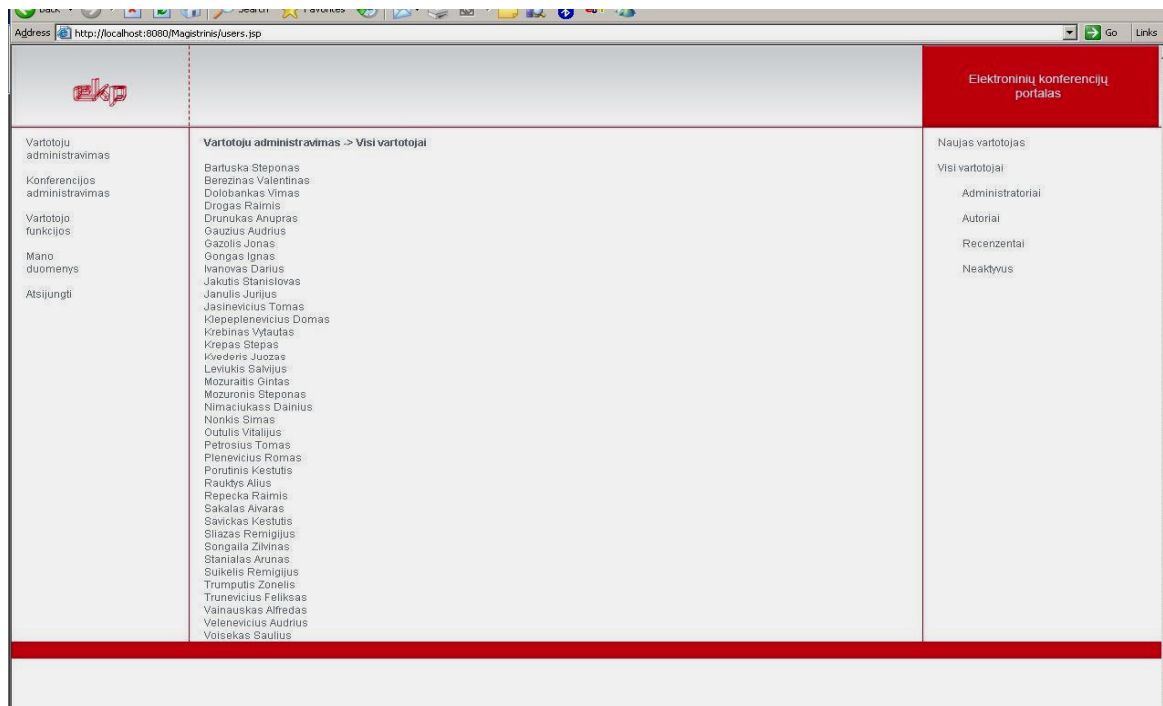
5.2 pav. Prisijungimo prie sistemos langas

Jei vartotojas neteisingai suveda savo prisijungimo informaciją, sistema po keleto klaidingų bandymų pasiūlo atsiusti elektroniniu paštu naujai sugeneruotą vartotojo slaptažodį, su kuriuo jis galėtų patekti į sistemą.



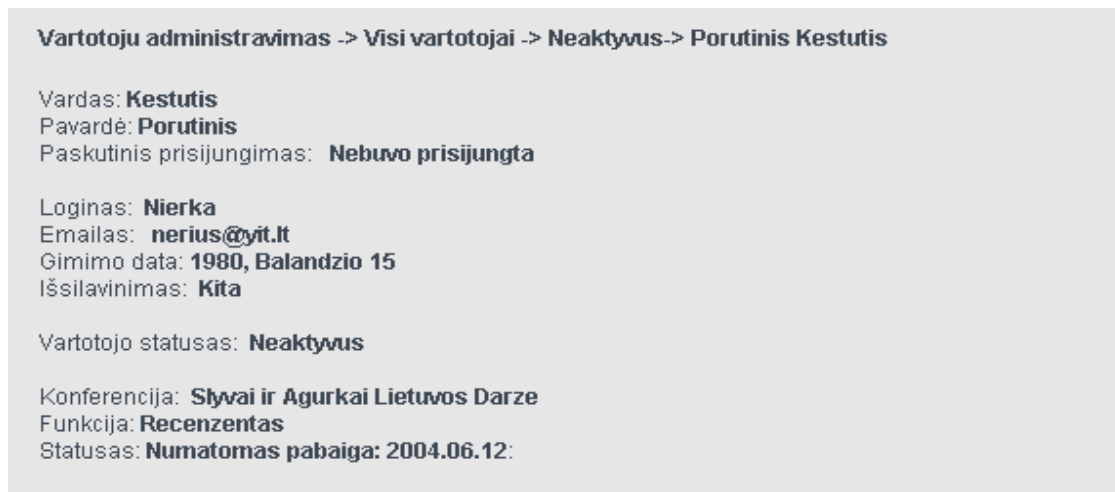
5.3 pav. Informacija apie nepavykusį prisijungimą

Bendra sistemos vartotojo sąsaja organizuota taip, kad vartotojas nebūtų apkrautas didžiuliu kiekiu meniu punktų. Sistemos vartotojo sąsaja kuriama hierarchiniu principu, tai yra, jog vartotojas iš aukštesnio lygio meniu keliauja vis žemesnio lygmens link.



5.4 pav. Sistemos vartotojo sąsaja

Kadangi buvo siekiama neapkrauti vartotojo daugybe meniu punktu, kurių didelėse sistemose yra daug, todėl navigacija tarp puslapių gali būti paini. Problemai šalinti buvo pasirinktas paprastas sprendimas – būsenos eilutė. Joje matome, koku būdu iki esamos padėties vartotojas atkeliavo, kokios sistemos funkcijos buvo naudojamos ir kokios prieinamos dabar.




5.5 pav. Informacija apie sistemos vartotoją



5.6 pav. Sistemos būsenos eilutė

Konferencijose dalyvauja didelis kiekis vartotojų, todėl jų valdymas yra sudėtingas ir tam reikalingi atskiri filtrai. Šioje sistemos realizuoti filtrai pagal vartotojų teisių tipą, bei vienas sudėtinis filtravimas, kuris leidžia atlikti vartotojų paiešką pagal daugumą kriterijų (pagal pavardę, konferenciją, metus, straipsnį, recenziją ir t.t.). Visas sistemos vartotojų sąrašas prieinamas tik sistemos administratoriui, konferencijos administratorius gali dirbti tik su konferencijos dalyvių sąrašu, kitiems vartotojams konferencijos dalyvių sąrašas nėra prieinamas, jei jis nėra viešas.

		Elektroninių konferencijų portalas
Vartotojų administravimas Konferencijos administravimas Vartotojo funkcijos Mano duomenys Atsijungti	Vartotojų administravimas -> Visi vartotojai Bartuska Steponas Berezinas Valentinas Dolobankas Virmas Drogas Raimis Drunukas Anupras Gauzius Audrius Gazolis Jonas Gongas Ignas Ivanovas Darius Jakulis Stanislovas Janulis Jurijus Jasinevicius Tomas Klepeplenevicius Domas Krebinas Vytautas Krepas Stepas Kvederis Juozas Leviukis Salvijus Mozuraitis Gintas Mozuronis Steponas Nimaciukas Dainius Nonkis Simas Outulis Vitalijus Petrosius Tomas Plenevicius Romas Porutinis Kestutis Rauktys Alius Repecka Raimis Sakalas Alvaras Savickas Kestutis Sliazas Remigijus Songaila Zilvinas Stanikala Arunas Suikelis Remigijus Trumpulis Zonelis Trunevicius Feliksas Vainauskas Alfredas Mozuraitis Gintas Mozuronis Steponas Nimaciukas Dainius Nonkis Simas Outulis Vitalijus Petrosius Tomas Plenevicius Romas Porutinis Kestutis ...	Naujas vartotojas Visi vartotojai Administratoriai Autoriai Recenzantai Neaktyvus

5.7 pav. Sistemos vartotojų sąrašas

Nauji dalyviai yra įtraukiami į sistemą tik turint sistemos administratoriaus teises, įvedant jų pradinį duomenis, leidžiančius jiems prisijungti į sistemą ir pradėti savo mokslinę veiklą. Vėliau, kai vartotojas dalyvauja konferencijose, visa jo istorija (straipsniai, recenzijos, dalyvavimas konferencijose, kiti įvykiai) kaupiasi ir nėra trinami.

Vartotoju administravimas -> Naujas vartotojas

Vardas:

Pavardė:

Loginas:

Slaptažodis:

Email:

Gimimo data:

Išsilavinimas:
 ▼

5.8 pav. Naujo vartotojo pradinė inicializacija

Naują konferenciją registruoja tik sistemos administratorius, ir ją skelbdamas parenka ir jos administratorių, kuris yra atsakingas už visą tolesnį konferencijos procesą, nes sistemos administratorius atlieka tik patį naujos konferencijos inicializavimą.

Konferencijos administravimas -> Skelbti konferencija

Konferencijos pavadinimas:


Konferencijos pradžia:

Konferencijos pabaiga:


Konferencijos administratorius:
 ▼

5.9 pav. Naujos konferencijos pradinė inicializacija

Kiekvienas vartotojas gali dalyvauti iš karto keliuose konferencijose su skirtingomis teisėmis. Vartotojui yra rodomas sąrašas konferencijų, kuriose jis dalyvauja, ir pasirinkęs konkrečią konferenciją, vartotojas ten atlieka veiksmus, kurie jam priskirti pagal jo teisių lygmenį.

		Elektroninių konferencijų portalas
Vartotojų administravimas	Konferencijos administravimas -> Vykstancios konferencijos Įmonių tinklu bei duomenų apsaugos sprendimai nuo isorinių isilauzimų. Inžinerinių sistemų diegimo problemos moderniuose ofisuose Java kalbos populiarumas Lietuvos programines irangos kurimo sferoje. Java programavimo kalbos privalumai ir trukumai WEB pagrindu veikiančiose sistemose Lietuvos studentų stazuoclu užsienyje itaka jų studiju kolybei Populiariausios Antivirusines sistemos Lietuvos didziosiuose İmonese.	Skelbti konferencija
Konferencijos administravimas		Vykstancios konferencijos
Vartotojo funkcijos		
Mano duomenys		
Atsijungti		

5.100 pav. Vykstančių konferencijų sąrašas

		Elektroninių konferencijų portalas
Vartotojo funkcijos	<u>Mano konferencijos</u> <u>Įmonių tinklu bei duomenų apsaugos sprendimai nuo isorinių isilauzimų.</u> <u>Inžinerinių sistemų diegimo problemos moderniuose ofisuose</u> <u>Java kalbos populiarumas Lietuvos programines irangos kurimo sferoje.</u>	
Mano duomenys		
Atsijungti		

5.114 pav. Vartotojui prieinamų konferencijų sąrašas

Kiekviena konferencija yra skaidoma į tam tikras temas, kurioms autoriai rašo straipsnius. Temų sąrašas gali padidėti ir konferencijos eigoje, jei jos administratorius įkelia naujų temų ir pakviečia dalyvius rašyti straipsnius.

Konferencijos administravimas -> Vykstancios konferencijos -> Įmonių tinklu bei duomenų apsaugos sprendimai nuo isorinių isilauzimų. Konferencija: Įmonių tinklu bei duomenų apsaugos sprendimai nuo isorinių isilauzimų. Pradžia: 2005.1.22 21.53 Pabaiga: 2005.6.27 21.53 Konferencijos administratorius: Juozas Kvederis
Konferencijos temos: Duomenų archyvavimo įrenginiai. Privalumai ir trukumai ForteGate interneto vartu sargai. Ar jie patikimi? Įmonės duomenų pavogimas. Kaip tai daroma?

5.125 pav. Konferencijos temų sąrašas

Konferencijos administravimas -> Vykstancios konferencijos -> Imoniu tinklu bei duomenu apsaugos sprendimai nuo isoriniu isilauzimu. -> Prideti tema

Nauja tema konferencijai:

Sukurtos konferencijos temos:

Duomenu archyvavimo irenginiai. Privalumai ir trukumai

ForteGate interneto vartu sargai. A r jie patikimi?

Imones duomenu pavogimas. Kaip tai daroma?

5.136 pav. Naujos temos įkėlimas

Skelbiant naują konferenciją, visiems sistemos vartotojams elektroniniu paštu išsiunčiamas pakvietimas dalyvauti konferencijoje. Pakvietime yra nuoroda, kurią paspaudus, atliekamas patvirtinimas, jog vartotojas sutinka dalyvauti konferencijoje ir taip patenka į norinčių dalyvauti sąrašą, iš kurio administratorius atrinkinėja konferencijos dalyvius. Tie dalyviai, kurie nėra pakviečiami dalyvauti konferencijoje, taip pat lieka sistemoje ir vėl gaus pakvietimą registruotis, kai bus vykdomos naujos konferencijos.

Vartotoju administravimas -> Visi vartotojai -> Konferencija apie C++ kalba -> Pageidaujantys dalyvauti vartotojai

[Bartuska Steponas](#)

[Berezinas Valentinus](#)

[Dolobankas Vimas](#)

[Drogas Raimis](#)

[Drunukas Anupras](#)

[Gauzius Audrius](#)

[Gazolis Jonas](#)

[Gongas Ignas](#)

[Ivanovas Darius](#)

[Jakutis Stanislovas](#)

[Janulis Jurijus](#)

[Jasinevicius Tomas](#)

[Klepeplenevicius Dornas](#)

[Krebinas Vytautas](#)

[Krepas Stepas](#)

[Kvederis Juozas](#)

[Laviukis Salvijus](#)

[Mozuraitis Gintas](#)

[Mozuronis Steponas](#)

[Nimaciukass Dainius](#)

[Nonkis Simas](#)

[Outulis Vitalijus](#)

[Petrosius Tomas](#)

[Plenevicius Romas](#)

[Porutinis Kestutis](#)

[Raukty Alius](#)

[Repecka Raimis](#)

[Sakalas Alvaras](#)

[Savickas Kestutis](#)

[Sijazas Remigijus](#)

[Songaila Zilvinas](#)

[Stanialas Arunas](#)

[Suikelis Remigijus](#)

[Trumputis Zonelis](#)

[Trunevicius Feliksas](#)

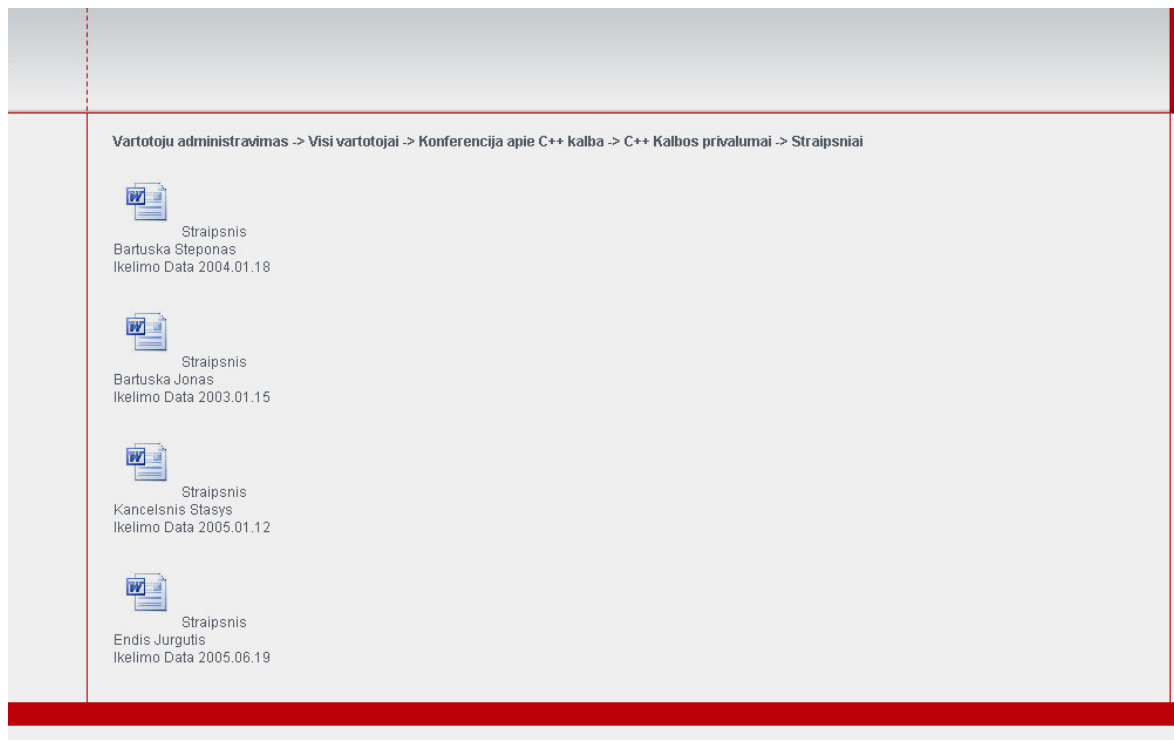
[Vainauskas Alfredas](#)

[Velenevicius Audrius](#)

[Voisekas Saulius](#)

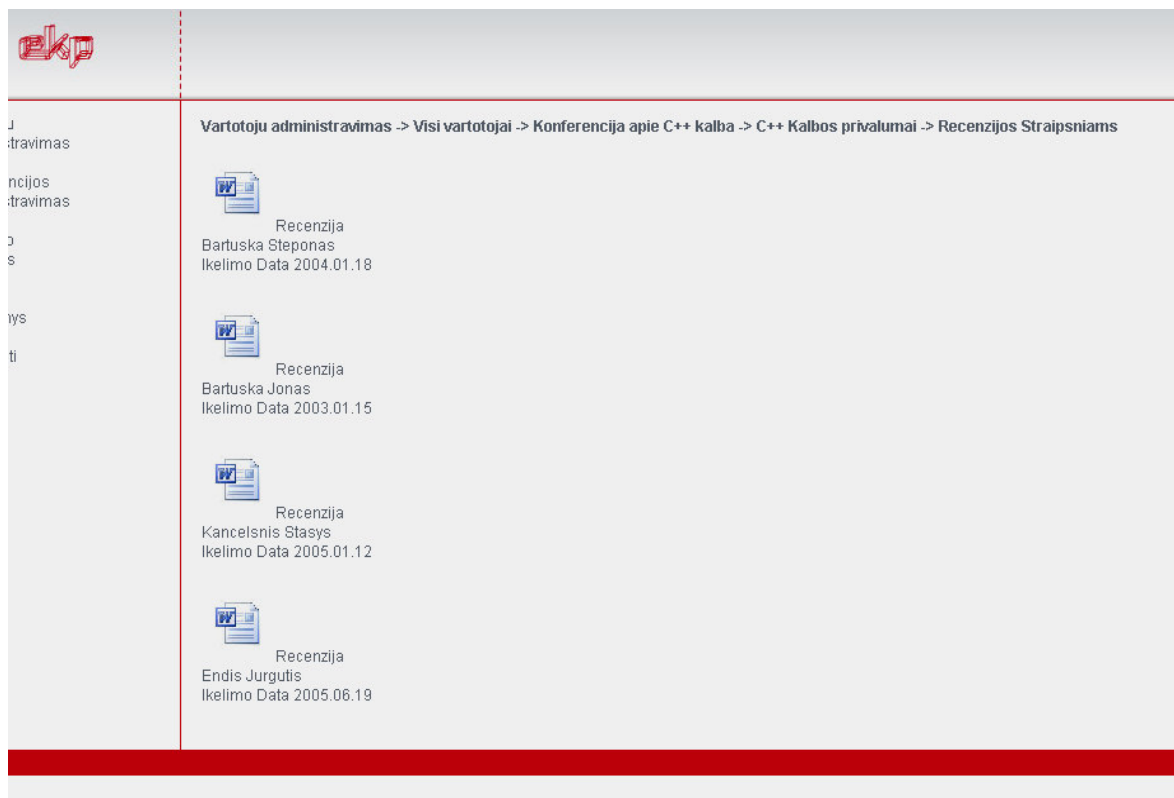
5.147 pav. Galimų konferencijos dalyvių sąrašas

Vartotojai kelia straipsnius pagal tam tikras temas, todėl pasirinkęs temą, konferencijos administratorius, mato, kokie straipsniai yra įkelti. Pasirinkus bet kurį įkeltą straipsnį iš sąrašo, administratorius gali matyti visą jo detalią informaciją. Autoriai turi teisę įkelti straipsnius tol, kol negauna specialaus pranešimo elektroniniu paštu, jog straipsnių kelti nebegalima. Tuo pačiu metu recenzentai yra informuojami, jog jau yra prasideda recenzijavimas ir yra kviečiami rašyti savo recenzijas.



5.158 pav. Konferencijos temos straipsnių sąrašas

Kai baigiasi straipsnių kėlimo fazė, prasideda straipsnių recenzijavimas. Visi vartotojai, kurie yra užregistruoti konferencijai kaip recenzentai, gauna priminimą elektroniniu paštu, jog prasidėjo recenzijavimo fazė. Prisijungę prie sistemos, jie mato, kokie straipsniai yra jiems priskirti, kuriuos reikia recenzuoti. Savo recenzijas jie įkeltinėja tol, kol baigiasi recenzijavimo fazė, o sistemos administratorius mato bendrą vis didėjanti sąrašą įkeltų recenzijų.



5.169 pav. Recenzijų sąrašas

Ši programinė įranga leidžia ne tik vykdyti mokslines konferencijas, bet ir nedalyvaujant jų veikloje susipažinti su visomis vykstančiomis ar vykusiomis, procesais, vykstančias mokslo srityje, todėl asmenims, atsakingiems už švietimo sistemas Lietuvoje, taip pat atsiranda paprasta galimybė iš arčiau pažvelgti į mokslo įstaigų vykdomą mokslinę veiklą.

5.2 Kokybės analizė

Sistemos tikslas buvo konferencijos darbo automatizavimas ir administratorių laiko taupymas, darbo automatizavimas ir greitesnis pačios konferencijos vykdymas, nes jo tarpinės fazės, kuriose atliekami administratorių rūšiavimo, publikavimo darbai labai sutrumpėja.

Atlikdami skaičiavimus, apklausėme sistemos prototipą bandžiusius Kauno Technologijos Universiteto doktorantus. Pasirinkome keturis funkcinius pjūvius, skaičiavome laiką, reikalingą funkcijai atlikti. Kiekvienas doktorantas užpildė lentelę, kurioje nurodė laiką sekundėmis, kiek jo testavimo metu užtrunkama, atliekant vieną ar kitą funkciją naudojantis automatizuota konferencijų valdymo sistema.

5.2.1 Kokybės tikrinimas pagal funkcionalumą

5.1 lentelė

Užtruktas laikas (sekundėmis) funkcijoms atlikti automatizuotoje sistemoje

Funkcija Testuotojas	Straipsnio priskyrimas recenzentui	Straipsnio publikavimas internete	Autoriaus straipsnio priėmimas	Vartotojo dalyvavimo patvirtinimas
Aidas Oželis	57	15	15	5
Lina Čeponinė	62	19	16	8
Gediminas Vedrickas	69	35	29	9
Milda Balandytė	50	15	16	5
Elita Miliauskaitė	72	17	23	8

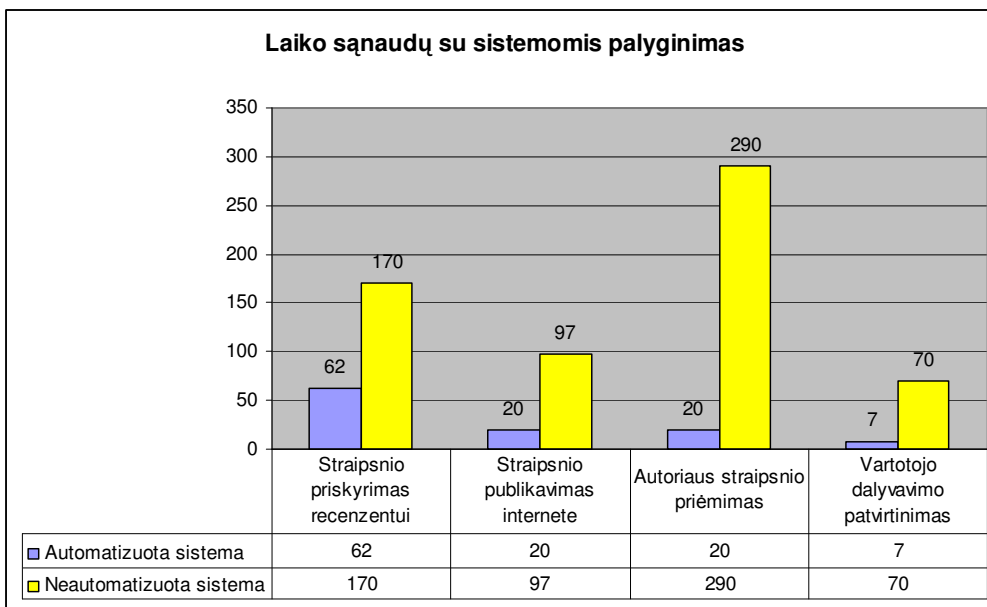
Įvertinimui ir lyginimui, kaip kokybiškai automatizuota sistema atlieka darbą ir taupo administratorių laiką, doktorantų pateikti duomenys buvo palyginti su laikais, kurie užtrunkami atlikti analogiškas funkcijas konferencijose, kur informacijos mainai vyksta elektroniniu paštu. Tokio pobūdžio konferencijos funkcijoms sugaištus laikus pateikė KTU Informacinių sistemų katedros docentė Lina Nemuraitė.

5.2 lentelė

Užtruktas laikas (sekundėmis) funkcijoms atlikti konferencijoje elektroniniu paštu

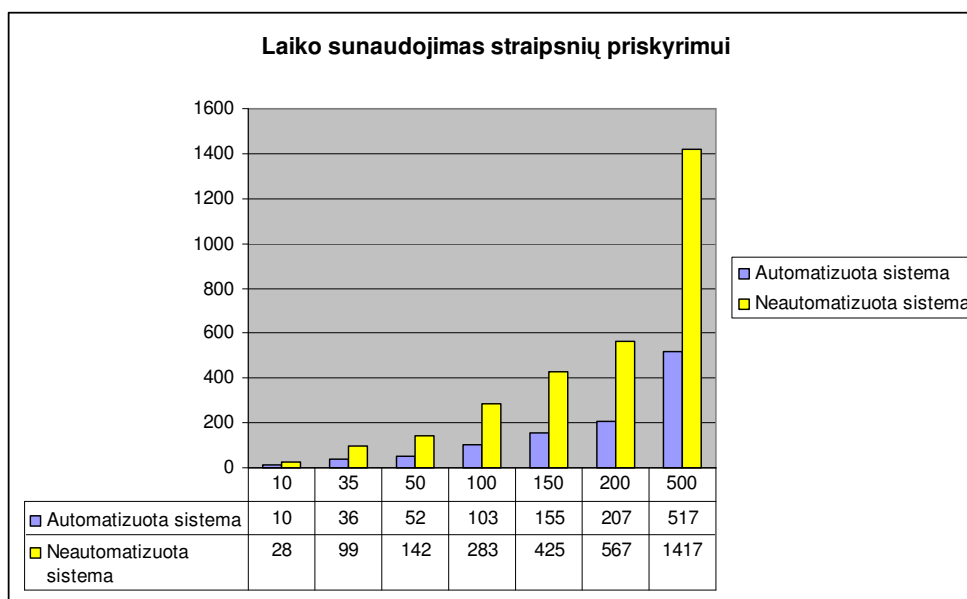
Funkcija Testuotojas	Straipsnio priskyrimas recenzentui	Straipsnio publikavimas internete	Autoriaus straipsnio priėmimas	Vartotojo dalyvavimo patvirtinimas
Lina Nemuraitė	170	97	290	70

Norėdami apskaičiuoti sistemos teikiamą naudas ir resursų taupymą, t.y. atlikti įvertinimą, ar sistemos realizacija atliko planuotus laiko ir resursų taupymo uždavinius, buvo naudojamas doktorantų įvertinimų vidurkis, kuris buvo lyginamas su laikais, sugaištamais neautomatizuotoje konferencijoje.

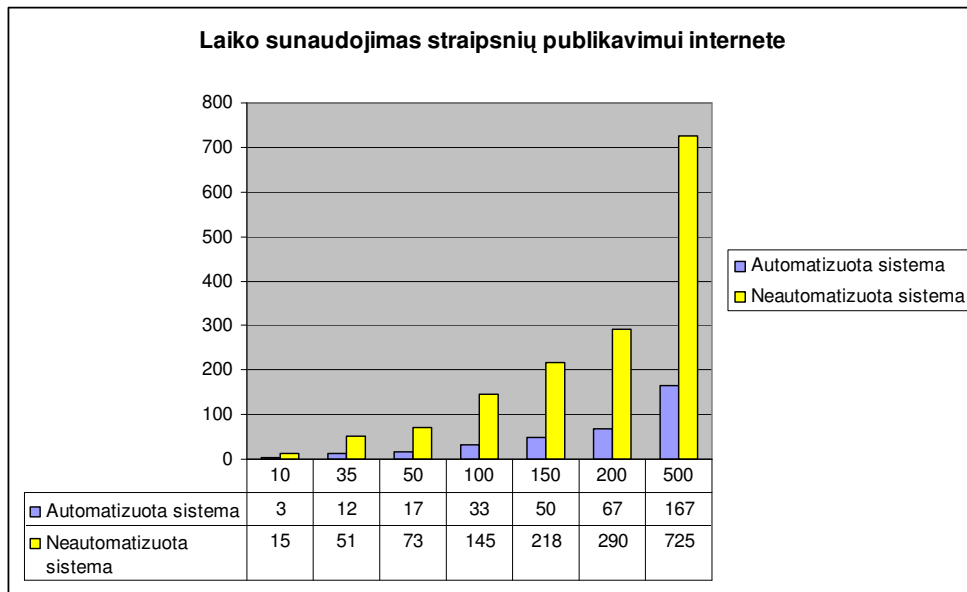


5.20 pav. Sugaištas laikas (sekundėmis) funkcijoms atlikti skirtingose sistemoje

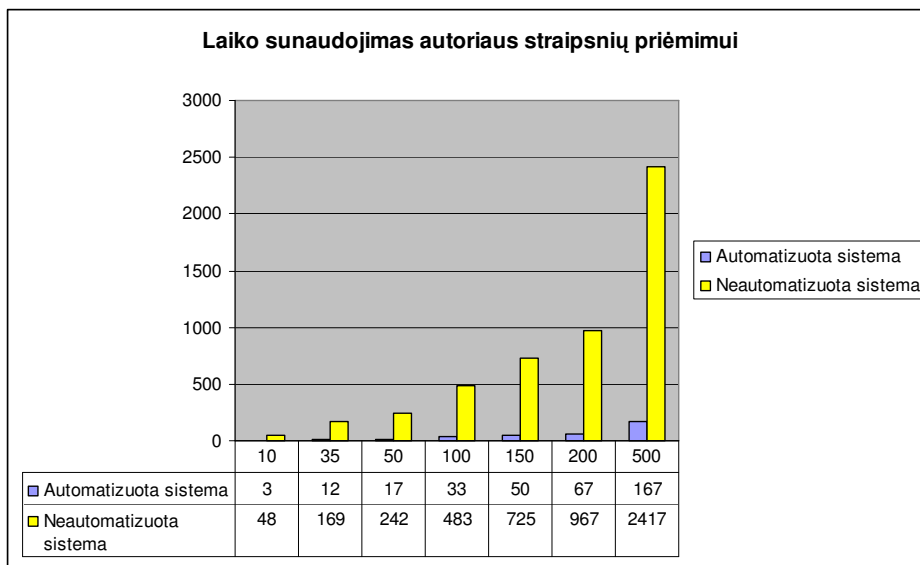
Tolesnėje analizėje atlikti skaičiavimai, kaip būtų taupomi resursai naudojantis automatine sistema didesnėse konferencijose. Skaičiavimo paprastumui laikas apvalinamas ir pateikiamas minučių tikslumu, o geresniam skaičiavimų vizualizacijos ir palyginimo rezultatui gauti naudojamas septynių skirtingų dydžių konferencijų darbo funkcijos modelis. Skaičiavimai atlikti visoms keturioms anksčiau nagrinėtoms funkcijoms, jų resursų naudojimo grafikus pateikiame žemiau.



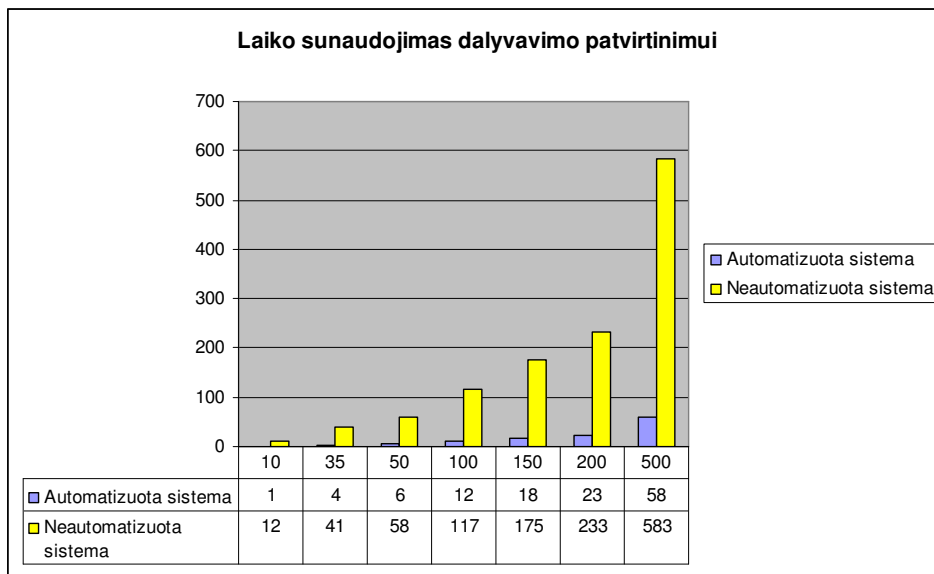
5.21 pav. Laiko sunaudojimas straipsnių skirstymui



5.22 pav. Užtruktas laikas (minutėmis) straipsnių publikavimui internete



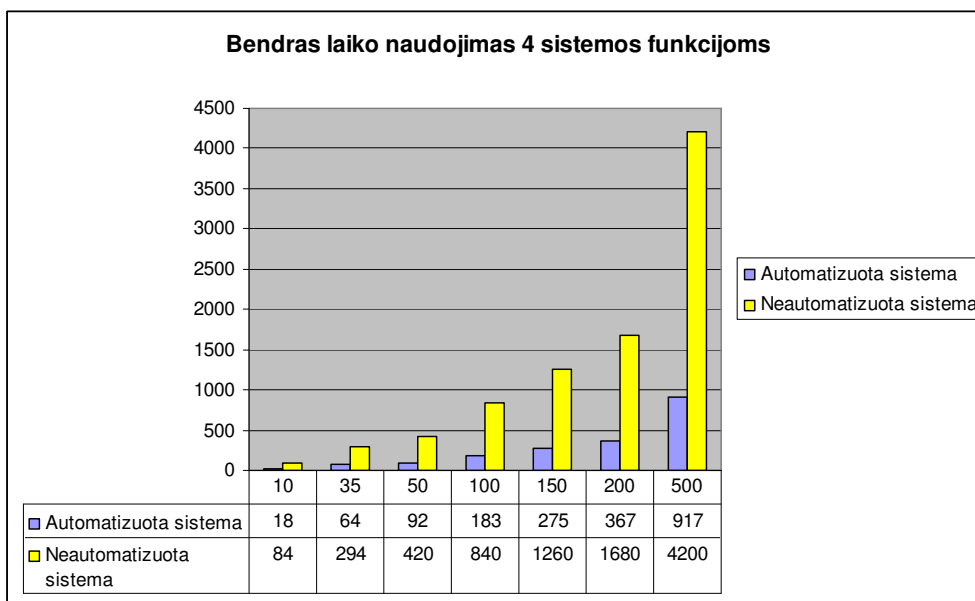
5.23 pav. Užtruktas laikas (minutėmis) autorių straipsnių priėmimui



5.24 pav. Užtruktas laikas (minutėmis) konferencijos dalyvių registracijos patvirtinimui

5.2.2 Kokybės analizės rezultatai

Kaip galime įsitikinti iš anksčiau pateiktų grafikų, kiekviename iš nagrinėtų funkcionalumų sistema veikia kokybiškai – realizuoja pagrindinį savo kūrimo ir modeliavimo tikslą – resursų taupymą. Apibendrinimui galime įvertinti bendrą nagrinėtą keturių sistemos funkcijų resursų sutaupymą toms pačioms septynioms skirtingo dydžio konferencijoms.



5.25 pav. Užtruktas laikas (minutėmis) darbui su 4 sistemos funkcijomis

5.3 Taikymo galimybių analizė

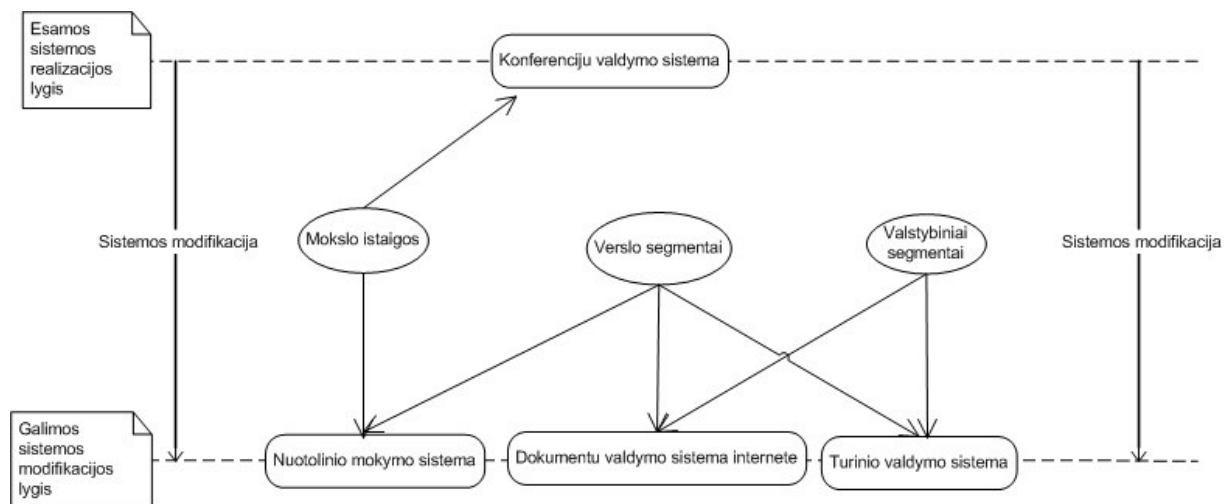
Konferencijų darbo automatizavimo programinė įranga skirta informacijos valdymui, tvarkymui, publikavimui ir apsaugos užtikrinimui. Vykstantis konferencijos procesas reikalauja daug resursų jos valdymui, pateikiamos informacijos tvarkymui, rūšiavimui ir publikavimui. Sistemos kūrėjų pagrindinis tikslas – sukurti universalios sistemos prototipą, pasinaudojant technologija, leidžiančia atlikti projekto modifikacijas ir adaptacijas naujiems poreikiams lengviausiu keliu, atliekant modifikacijas ir jas jungiant prie esamo sistemos karkaso. Sukurta programinė įranga bus naudojama tolesniame projekte, todėl išanalizuotos ir pasiūlytos plėtimo galimybės: papildyti sistemą pranešimų siuntimu ir veiksmų leidimo/uždraudimo valdymo funkcijomis.

Tačiau sistemos pritaikymas tuo neapsiriboja. Lengvas adaptavimas leidžia ją nesunkiai platinti įvairiose rinkos segmentuose, nes kiekvieno kliento poreikiai sistemai yra kažkiek individualūs, todėl visada būtina adaptacija. Sukūrus lengvai modifikuojama sistemą, nesunku ją platinti, nes modifikacija nereikalauja daug resursų ir laiko.

Mūsų sistemos sritis – informacijos valdymas, tvarkymas ir prieiga per internetą yra itin aktuali šiomis dienomis, nes tai lengviausias kelias padaryti informaciją bei darbą su ja lengvai pasiekiamą. Ši sritis turi plačias panaudojimo sritis, tiek mokslo, tiek verslo, tiek valstybinėse įstaigose, visur, kur vyksta informacijos, failų apsikeitimas, automatizuoti procesai.

Nors sistema kurta kaip konferencijų darbo automatizavimo sistema, jos funkcionalumas labai nesunku plėsti, todėl jos galimas panaudojimas įvairus. Labai nesunku šią sistemą reorganizuoti į dokumentų valdymo sistemą *online*, kas šiomis dienomis yra gana plataus panaudojimo sritis, nes kompanijos, ypač privataus kapitalo, diegiasi sistemas, leidžiančias efektyviai valdyti dokumentaciją, todėl modelis, leidžiantis saugiai pasiekti visą turimą įmonės dokumentaciją per internetą, yra tikrai patrauklus.

Pačios konferencijų valdymo sistemos galimybės rinkoje taip pat yra didžiulės, nes iki šiol daugumoje mokslo įstaigų konferencijų valdymas bei informacijos mainai vyko elektroniniu paštu, todėl anksčiau ar vėliau visos jos pereis prie automatizuotų sistemų. Tai leidžia ne tik sutaupyti dalyvių laiko, nes jis daug aiškiau gali suprasti savo, kaip dalyvio funkcijas, tačiau ir organizatorių resursus, kuriuos labai nesunku išversti ir į piniginę išraišką. Naudojamų resursų sumažėjimas reiškia mažesnę etatų poreikį valdyme, todėl įsigijus automatizuotas sistemas, galima suskaičiuoti ir kaštų sutaupymo procentą.



5.26 pav. Galimos sistemos modifikacijos

Tokių sistemų įdiegimas universitetuose ar kituose mokslo įstaigose atliktų ne tik pagrindinį tikslą – automatizavimą, tačiau ir leistų efektyviai keistis informacija tarp atskirų mokslo vienetų. Vien sistemos naudojimas leidžia organizuoti konferencijas, kurios gali vykti kur kas platesniu mastu, nei universitetas ar kita mokslo įstaiga, nes visas procesas vyksta internetu. Tai įgalina kelti bendrą žinių lygį, nes atsiranda galimybė paprasčiausiu keliu ją dalintis ir keistis, o tai yra svarbiausia.

6. IŠVADOS

- Sukurtas konferencijų darbo automatizavimo sistemos prototipas, skirtas informacijos bei darbo eigos valdymui, leidžiantis vykdyti konferencijas internetu, atsisakant konferencijų vykdymo proceso elektroniniu paštu.
- Sistemos sukūrimui naudotas „Java Tapestry“ karkasas, leidžiantis realizuoti paprastai tobulinamą ir nepriklausomą nuo platformos sistemą.
- Java MVC (Model – View – Controller) šablono naudojimas leido išspęsti veiklos ir atvaizdavimo atskyrimo problemą. Sprendimo panaudojimo rezultatas – paprastesnis ir lengvesnis sistemos funkcijų plėtimas, atnaujinimas ir tobulinimas.
- Analizės dalyje nagrinėti, aiškinti ir detalizuoti sprendimai yra praktiškai realizuoti internetu veikiančiame konferencijų informacinės sistemos portalo prototipe.
- Projekto kokybės tikslai pilnai įgyvendinti, sistema veikia efektyviai ir leidžia taupyti konferencijų administravimo resursus bei kaštus, nes funkcijos automatizuotos, to rezultatas – daug mažesnis konferencijos administratoriaus laiko sąnaudų poreikis.
- Atlikti skaičiavimai rodo, jog programinės įrangos atliekamas resursų taupymo koeficientas gana didelis. Buvo atsitiktinai pasirinktos keturios sistemos funkcijos, matuojami joms sugaišti laikai ir pagal juos atliekama analizė, skaičiuojant resursus įvairių dydžių konferencijoms valdyti.
- Sukurta sistemos prototipas lengvai pritaikomas kitokios pakraipos, tačiau analogiškais principais dirbančioms programinės įrangos sistemoms verslo, mokslo, valstybiniame segmente: turinio valdymo, nuotolinio mokymo, dokumentų valdymo sistemoms.
- Tyrimai įrodo, jog konferencijų automatizavimo sistemų poreikis rinkoje akivaizdus, dauguma šios dienos mokslo konferencijų vyksta elektroniniu paštu, todėl ši niša greitai laiku bus užpildyta, įsigyjant sistemas, analogiškas sukurtajam prototipui.

7. LITERATŪRA

Produkto kūrimo eigoje buvo naudojamosi įvairios literatūros šaltiniai, kurių sąrašas pateiktas žemiau.

- 1 . Olegas Vasilecas, Albertas Caplinskas, Wita Wojtkowski, W. Gregory Wojtkowski Jose Zupancic, Stanislav Wrycza. „Proceedings of the Thirteenth International Conference on Informatics Systems Development. Advances in Theory, Practice and education ” . pp. 430-437, 2004.
2. Howard M. Lewis Ship. „Tapestry in Action“. Pp.37-60, 2004.
3. Neal Ford. „Art of Web development“. pp.106-130, 2003.
4. CMSWorks. CMSWatch™. [interaktyvus]. [žiūrėta 2005.03.20]. Priega per internetą: <http://www.cmswatch.com>
5. Rutkauskaitė R., Nemuraitė L. Šablonų naudojimas kuriant duomenų apdorojimo sistemas internete. Iš: Informacinės technologijos 2004: konferencijos pranešimų medžiaga, Kaunas 2004 sausio 28 - 29 d.. Kaunas, 2004, p. 453-459. Informacinės technologijos 2004.
6. Howard M. Lewis Ship. „Tapestry User’s Guide“ pp.110-117, 2003.
7. Lu J. Reengineering of Database Applications to EJB Based Architecture. CAISE 2002, Lecture Notes in Computer Science, Vol. 2348, pp. 361-376, 2002.
8. Comparison of Java Web Frameworks. [interaktyvus] [žiūrėta 2004.03.27]. Priega per internetą: *The Sixth Framework Programme (2002-2006)*.
9. Matt Raible. „Comparing Web frameworks: Struts, Spring MVC, WebWork, Tapestry & JSF“. [interaktyvus]. [žiūrėta 2005.04.20]. Priega per internetą: <http://www.raibledesigns.com>
10. Kochmer C. An Introduction to Struts. [interaktyvus]. [žiūrėta 2004.05.02]. Priega per internetą: <http://www.jspinsider.com/content/jsp/struts/strutsintro.jsp>
11. Howard M. Lewis Ship. „Tapestry Developer’s Guide“. Pp. 64-71, 2003
12. Schuster E., Wilhelm S. Web-Content-Managementsysteme. Informatik-Spektrum. vol. 23, Number 6, Publisher: Springer-Verlag Heidelberg, 2002.
13. ServerSideScripting techniques analysis. [interaktyvus]. [žiūrėta 2005.005.02]. Priega per internetą: <http://www.b2bsim.de/documents/wewior/main.html>
14. Struts 1.1 Controller UML diagrams. [interaktyvus]. [žiūrėta 2005.01.12]. Priega per internetą: <http://rollerjm.free.fr/pro/Struts11.html>
15. SUN. Comparing JavaServer Pages and Microsoft Active Server Pages Technologies. [interaktyvus]. [žiūrėta 2005.03.12]. Priega per internetą: <http://java.sun.com/products/jsp/jsp-asp.html>