

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Rytis Budreika

**Automatizuoti veiklos procesai internete:
konferencijų sistema**

Magistro darbas

Darbo vadovas

doc. dr. Lina Nemuraitė

Kaunas, 2006

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Rytis Budreika

**Automatizuoti veiklos procesai internete:
konferencijų sistema**

Magistro darbas

Kalbos konsultantė

2006-05

Lietuvių k. katedros doc.
dr. J. Mikelionienė

Vadovas

doc. dr. Lina Nemuraitė
2006-05

Recenzentas

2006-05

doc. dr. E. Karčiauskas

Atliko

2006-05-15

IFM 0/2 gr. stud.
Rytis Budreika

Kaunas, 2006

TURINYS

1.	PRATARMĖ.....	5
2.	ĮVADAS	6
3.	INTERNETE VYKDOMŲ VEIKLOS PROCESŲ AUTOMATIZAVIMO PROBLEMA IR JOS SPRENDIMO PAGRINDIMAS	10
3.1.	Konferencijų organizavimo problema	10
3.2.	Panašių sistemų analizė	12
3.3.	Interneto sistemų kūrimo technologijų analizė	15
3.4.	Duomenų valdymo technologijų analizė	15
3.5.	Veiklos logikos ir vaizdo valdymo atskyrimas – Tapestry karkasas	17
3.6.	Veiklos logikos ir duomenų nepriklausomumas – Torque technologija	19
3.7.	Analizės išvados	20
4.	KONFERENCIJŲ SISTEMOS MODELIS.....	21
4.1.	Konferencijų sistemos reikalavimai.....	21
4.2.	Konferencijų veiklos procesų modelis.....	31
4.3.	Veiklos klasių modelis.....	36
4.4.	Automatizuoto etapų valdymo agento modelis	41
5.	AUTOMATIZUOTŲ VEIKLOS PROCESŲ REALIZACIJA TAPESTRY KARKASU	42
5.1.	Loginė sistemos architektūra	42
5.2.	Sistemos klasių modelis.....	43
5.4.	Sistemos architektūros elementų sąveika	46
5.5.	Automatizuotų veiklos procesų šablonai	48
5.6.	Automatizuotų komponentų integravimas į Tapestry karkasu sukurtą sistemą	49
5.7.	Agento klasių modelis	50
5.8.	Sistemos realizacijos klasių modelis.....	51
5.9.	Duomenų modelis	54
5.10.	Diegimo architektūra	55
6.	KONFERENCIJŲ SISTEMOS KOKYBĖS IR EFEKTYVUMO ANALIZĖ.....	56
6.1.	Sistemos veikimo analizė.....	56
6.2.	Automatizuotų procesų efektyvumo tyrimas	63
6.3.	Sistemos plečiamumo tyrimas	64
6.4.	Tapestry karkaso ir Torque technologijos įtaka sistemos kūrimui	66
7.	IŠVADOS	70
8.	LITERATŪRA	71
	SANTRUMPŲ IR TERMINŲ ŽODYNAS	73
	I PRIEDAS. STRAIPSNIS	75

Paveikslų turinys

1 pav. Konferencijos rengimo etapai	11
2 pav. Tapestry karkaso komponentinis išskaidymas	17
3 pav. Tapestry karkasu paremtos sistemos komponentų sąveika	19
4 pav. Konferencijų sistemos panaudojimo atvejų modelis	22
5 pav. Konferencijų veiklos procesai	32
6 pav. Konferencijų sukūrimo/skelbimo procesas	33
7 pav. Konferencijų pagrindinis organizavimo procesas – straipsnių, recenzijų įkėlimas	34
8 pav. Konferencijos užbaigimo procesas	35
9 pav. Automatizuotų agentų veiklos klasių modelis	37
10 pav. Pranešimų valdymo veiklos klasių modelis	38
11 pav. Straipsnių valdymo veiklos klasių modelis	39
12 pav. Recenzavimo valdymo veiklos klasių modelis	40
13 pav. Etapų valdymo modelis	41
14 pav. Konferencijų sistemos paketų struktūra	43
15 pav. Pagrindinės Tapestry karkaso sąsajos, klasės, metodai	44
16 pav. Konferencijų sistemos veiklos klasės	45
17 pav. Tapestry užklausų valdymo modelis	46
18 pav. Valdymo klasių modelis	46
19 pav. Recenzijos įvedimo sekų diagrama	47
20 pav. Etapų valdymo modelis	48
21 pav. Automatizuotų procesų integravimo modelis	49
22 pav. Automatizuoto agento klasių paketas	50
23 pav. Automatizuoto agento klasių detalizacija	50
24 pav. Papildomi sistemos paketai	51
25 pav. Tapestry karkaso realizacijos paketai	52
26 pav. Torque generuotų klasių paketai	53
27 pav. Duomenų modelis	54
28 pav. Sistemos architektūra	55
29 pav. Pradinis sistemos langas	56
30 pav. Prisijungusio dalyvio valdymo meniu	57
31 pav. Autoriaus ir recenzento darbo langas	57
32 pav. Straipsnių sąrašas	58
33 pav. Straipsnio įkėlimo į sistemą langas	58
34 pav. Konferencijų valdymo langas	59
35 pav. Konferencijos redagavimo langas	59
36 pav. Recenzijų vertinimo šablonų kūrimo langas	60
37 pav. Rankinio etapų valdymo darbo langas	61
38 pav. Konferencijos dalyvių sąrašas	61
39 pav. Konferencijos etapų valdymo langas	62

1. PRATARMĖ

Veikla perkeliama į internetą, kad informacija būtų lengvai prieinama kiekvienam suinteresuotam asmeniui. Veiklos procesų kompiuterizavimui keliami dideli reikalavimai – tokios sistemos turi būti lengvai naudojamos vartotojų, daugiakalbės, plečiamos, lanksčios pokyčiams. Be to, jos gali pačios atlikti didelę dalį organizacinių darbų.

Darbe pateikiamas konferencijų informacinės sistemos modelis ir jo įgyvendinimas, naudojant Tapestry karkasą, kuris suteikia sistemai conceptualų aiškumą ir galimybes plėsti ją ateityje. Didelis dėmesys skiriamas veiklos proceso automatizavimui – visą žinučių siuntimą, įvertinimų skaičiavimą reikiamais laiko momentais vykdo programos komponentai, administratorius tik nustato jų parametrus. Sistema leidžia vykdyti iš karto kelias konferencijas ir išsaugo jų istoriją.

2. ĮVADAS

Veiklos perkėlimas į internetą šiandien yra ne naujovė, bet būtinybė. Internetas tapo vienu iš pagrindinių informacijos apsikeitimo būdų, prieinamu dideliame žmonių skaičiui.

Konferencijų organizavimas jau ne pirmi metai vyksta internetu ir yra gana didelė tokių sistemų įvairovė – nuo paprastų informacinių puslapių iki didelių, sudėtingų sistemų. Tačiau Lietuvoje šiuo metu lietuviškų konferencijų sistemų nėra, nors paklausa yra didelė. Rengiant universitetų darbuotojų, magistrantų, doktorantų konferencijas, pasikliaujama tik elektroniniu paštu.

Pasaulinio lygio konferencijos turi gana išvystytas sistemas [1], [2]. Tačiau šios sistemos dažniausiai pritaikytos tam tikram specifiniam procesui, tam tikra kalba, ir nėra tinkamos nedidelėms konferencijoms, vykstančioms lietuvių kalba. Sukurti informacinių konferencijos puslapį nesunku, tačiau to nepakanka. Konferencijos organizatoriams tenka didelis darbo krūvis, kurį gali atlikti ar pagreitinti tinkama programinė įranga.

Vienas pagrindinių konferencijų organizavimo reikalavimų – nuolatinis organizatorių ir konferencijos dalyvių bendravimas, kuris dažniausiai vyksta elektroniniu paštu. Elektroninių laiškų siuntimą tikslinga automatizuoti, kad sistema pati išsiųstų standartinius laiškus ir sumažintų organizatorių apkrovimą. Šiame darbe siekiame maksimaliai automatizuoti laiškų siuntimą, iš anksto nustatant datas, kada, koku periodiškumu ir kokioms dalyvių grupėms reikia siųsti tam tikrų tipų pranešimus.

Kitas šio darbo tikslas – suvienodinti konferencijos valdymą. Tipiniai etapai[9] (paskelbimas, registracija, straipsnių priėmimas, recenzavimas ir kiti) skirtingose konferencijose išsidėsto įvairia tvarka ir įvairiais laiko momentais. Kitaip tariant, konferencijos skiriasi savo valdymo taisyklėmis. Turint galimybes nustatyti atitinkamus etapų parametrus (pradžią, pabaigą), galima juos įvairiais būdais išdėlioti laike, atsisakyti nereikalingų etapų. Reikalui esant, parametrus galima keisti, pavyzdžiui, pratęsti straipsnių priėmimą. Pagal nustatytus parametrus sistema suteikia vartotojams galimybes tam tikrame etape atlikti tam tikrus veiksmus ir padaro juos neprieinamus, etapui pasibaigus. Taip sistema pati save „valdo“, aišku, palikdama administratoriui galimybę tuos nustatymus pakeisti. Automatizuotai atliekamas recenzijų rezultatų skaičiavimas, organizatoriams paliekant tik galutinį įvertinimą. Sistemą galima pritaikyti įvairioms konferencijoms įvairiomis kalbomis, vienu metu organizuoti kelias konferencijas ir išsaugoti jų istoriją.

Sistemą buvo siekiama sukurti taip, kad ją būtų lengva plėsti ateityje. Todėl sukurtam konferencijų sistemos modeliui įgyvendinti pasirinktas Tapestry[3], [4], [13], [14] – interneto

sistemų kūrimo karkasas, kuris leidžia atskirti puslapių valdymą nuo veiklos logikos ir sudaro prielaidas lengvam, aiškiam kūrimui.

Konferencijų sistemos modelį galima pritaikyti ir kitoms sritims, pavyzdžiui, mokymo proceso organizavimo ir žinių vertinimo sistemai, kurioje bendravimas vykta tarp studentų ir dėstytojų.

Konferencijos sistemos funkcionalumas buvo suformuluotas remiantis užsakovo pageidavimais ir egzistuojančių konferencijų sistemų analize[1], [2]. Sistemos funkcionalumas neapima visų savybių, kurias turėtų turėti „ideali“ universali konferencijų sistema, tačiau turi keletą išskirtinių savybių, kurių trūksta esamoms tokio pobūdžio sistemoms: automatizuotas etapų valdymas, galimybė vartotojui peržiūrėti savo pateiktus straipsnius, istorijos saugojimas.

Sistemos architektūra ir realizacija aprašyta, pateikiant fragmentus iš programinės įrangos specifikacijų unifikauta UML kalba[10]: aprašomi panaudojimo atvejai, konferencijų vykdymo procesai, klasių modeliai.

Sistemos efektyvumo bei plečiamumo vertinimui buvo sudaryti įvertinimo kriterijai. Sukurta programinė įranga yra efektyvi – tai įrodo atlikti tyrimai, kuriuose matyti, jog laiko sąnaudos žymiai sumažėja nei darbus atliekant įprastu būdu.

Sistema buvo pradėta projektuoti kartu su Aivaru Sakalu, bet vėliau mūsų darbai išsiskyrė. Nedidelė projekto dalis gali sutapti, tačiau tyrimo kryptys ir realizacijos yra skirtingos.

Sukurtas produktas turi daug privalumų, lyginant su paprastomis konferencijų organizavimo priemonėmis internete. Sistema sukurta naudojant pažangią, sparčiai tobulinamą technologiją – Tapestry. Išskirtinės technologijos savybės – komponentinis programavimas, vaizdo ir veiklos procesų atskyrimas. Tapestry karkaso naudojimas leidžia lengvai keisti ir plėsti sistemos funkcionalumą.

Kita panaudota technologija – Torque[12]. Ji leido naudotis duomenų bazės teikiamomis paslaugomis, nenaudojant tiesioginių SQL užklausų. Torque sugeneruotos klasės leido nesudėtingai keisti klasių funkcionalumą.

Sistema sukurta taip, jog gali būti lengvai pritaikoma ir kitoms sritims. Sistemos pritaikymas kitai sričiai reikalauja minimalių pataisymų. Vienas iš galimų pritaikymų – akademinė mokytojo ir mokinio sistema. Kitas darbo privalumas – jo paliesta sritis.

Sukurta konferencijų organizavimo priemonė pirmiausia yra pagalbininkas konferencijų organizatoriams, kurie lig šiol visus organizavimo darbu atlikdavo rankiniu būdu. Tačiau ir kiti konferencijų dalyviai pajus sistemos privalumus – turėti nuolat prieinamą

naujausią informaciją, gauti iš sistemos priminimus ar pranešimus apie naujienas, matyti savo darbų istoriją.

Pagal suformuluotus reikalavimus sukurta ir įdiegta konferencijų organizavimo sistema. Ji pradeda naudoti konferencijai organizuoti adresu <<http://isd.ktu.lt:7000/KFR>>.

Darbo tyrimo sritis – internete vykdomų automatizuotų veiklos procesų modelių sudarymo ir jų įgyvendinimo programine įranga metodika. Šią metodiką sudaro:

- tipinių elementų (proceso etapų, jų parametrų, valdomų funkcijų) nustatymas analizės metu
- procesų ir agentų modelių sudarymas
- technologijų pasirinkimas, kuri leistų atskirti sąsajos valdymą nuo veiklos valdymo, automatizuotų veiklos logikos sąveiką su duomenų baze.

Analizuojant konferencijos procesą yra išskiriami etapai, išskiriami bendri požymiai, nustatomi parametrai, kuriais galime aprašyti elementus. Suskirstomi etapai, nustatomos valdančios funkcijos. Parametrizuoti tipiniai elementai bus naudojami modelių sudarymui.

Išskyrus elementus, kuriamas procesų ir agentų modelis. Pagal nustatytus elementus ir parametrus sudaromas automatizuotų procesų ir agentų modelis. Agentų modeliais apibūdinamas procesų autonominis valdymas.

Analizuojamos technologijos atskiriančios valdymą nuo veiklos logikos, automatizuojančios veiklos logikos sąveiką su duomenų baze. Tokių technologijų parinkimas realizacijai palengvins programinės įrangos kūrimą.

Visų trijų dalių visuma ir sudaro aprašytą metodiką. Ši metodika pritaikyta tyrimo objektui – konferencijų organizavimo sistemai.

Darbo struktūra.

Skyriuje **internete vykdomų veiklos procesų automatizavimo problema ir jos sprendimo pagrindimas** pateikiamos konferencijų sistemų problemos[11], analogiškų sistemų[1], [2] ir technologijų analizė[3], [4], [5], [6], [7], [8], [15] bei detalus pasirinktų technologijų aprašymas.

Skyriuje **konferencijų sistemos modelis** pateikiami sistemos reikalavimai, konferencijų veiklos procesai[9], [11], veiklos klasių ir automatizuoto etapų valdymo agento modeliai.

Skyriuje **automatizuotų veiklos procesų realizacija Tapestry karkasu** pateikiama sistemos architektūra ir klasių modeliai, architektūros elementų sąveika, automatizuotų

procesų šablonai, automatizuotų komponentų integracija į Tapestry karkasu sukurtą sistemą, agento klasių modelis, realizacijos modelis, duomenų modelis ir diegimo architektūra.

Skyriuje **konferencijų sistemos kokybės ir efektyvumo analizė** analizuojamas sistemos veikimas, palyginamos neautomatizuoto darbo ir darbo su automatizuota sistema laiko sąnaudas, atliekamas sistemos plečiamumo įvertinimas.

Išvadose pateikiami apibendrinti tyrimo rezultatai.

Prieduose pateikiamas straipsnis tyrimo tematika, išspausdintas 2006 metų tarpuniversitetinės magistrantų konferencijos leidinyje[11].

3. INTERNETE VYKDOMŲ VEIKLOS PROCESŲ AUTOMATIZAVIMO PROBLEMA IR JOS SPRENDIMO PAGRINDIMAS

Šio tyrimo sritis yra internete vykdomų automatizuotų veiklos procesų modelių sudarymo ir jų įgyvendinimo programine įranga metodika. Tyrimo objektas yra konferencijų organizavimo sistema, kurioje vyksta procesai, susidedantys iš daugelio etapų ir reikalaujantys didelių organizacinių pastangų: bendravimo su įvairiomis vartotojų grupėmis, nuolatinio keitimosi informacija, prieigos teisių keitimo, ir kita. Šiame skyriuje pateikiamos konferencijų organizavimo internetu veiklos procesų kompiuterizavimo problemos ir jų sprendimo būdai, aptariamos pasirinktos technologijos.

3.1. Konferencijų organizavimo problema

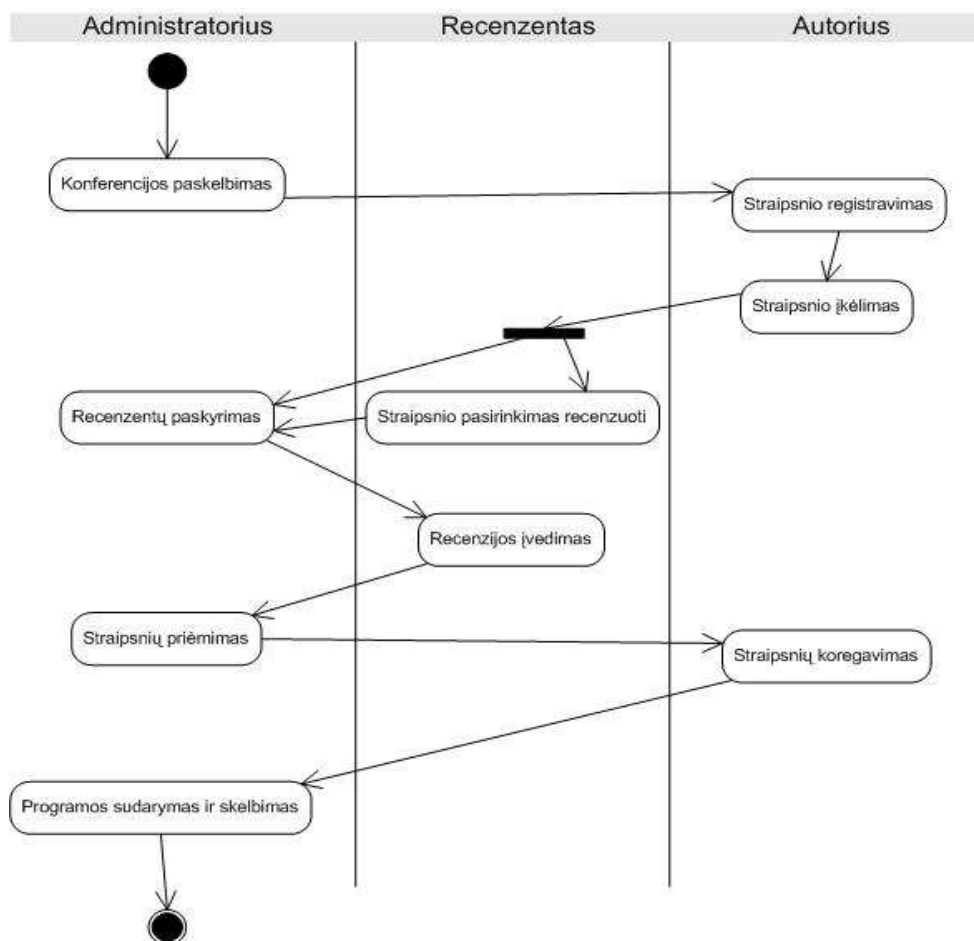
Konferencijų organizavimas yra jau nusistovėjusių procesų rinkinys [9]. Šie procesai kiekvienai konferencijai gali nedaug skirtis, tačiau standartinę konferenciją sudaro šie pagrindiniai etapai:

- Konferencijos skelbimas
- Straipsnio registravimas
- Recenzentų paskyrimas
- Straipsnio įkėlimas
- Recenzijos įvedimas
- Straipsnių priėmimas
- Straipsnio koregavimas
- Programos sudarymas

Šie etapai ir jų vykdomieji asmenys pateikti 1 paveiksle.

Pats procesas neatrodo labai sudėtingas. Tačiau paėmus kiekvieną iš etapų ir panagrinėjus jo veikimą, sužinome, jog administratoriui tenka labai daug pasikartojančio ir ne visada lengvai atliekamo darbo.

Kiekviename iš šių etapų galima atlikti tam tikrus veiksmus. Šiuos etapus nustatinėja administratorius ir apie jų pradžią ir pabaigą informuoja atitinkamą konferencijos dalyvį. Tačiau tai yra pasikartojantis ir daug išteklių reikalaujantis darbas. Šių etapų valdymas gali būti automatizuotas, paliekant administratoriui tik konfigūravimo darbus.



1 pav. Konferencijos rengimo etapai

Atidžiau panagrinėjus konferencijos etapus, pastebėta, kad jiems visiems galima pritaikyti bendrą šabloną. Išskirti tokie bendri konferencijos etapų parametrai: pavadinimas, pradžios data, pabaigos data, interesantų grupė. Šie parametrai bus įvertinti tipiniame automatizuotų procesų modelyje.

Kitos su šiuo darbu susijusios problemos yra būdingos bet kuriai, didesnei ir ilgesniam naudojimui kuriamai, sistemai: pasirinkti tokias technologijas, kad būtų nesunku modifikuoti sistemą kūrimo metu ir ateityje, atsiradus poreikiui ją keisti.

Galimi šių problemų sprendimo būdai:

1. Sudaryti ir įgyvendinti automatizuojamų procesų modelius (ši galimybė nagrinėjama 4 skyriuje);
2. Užtikrinti kodo lankstumą, atskiriant programos logiką nuo pateikimo ir duomenų.

3.3 – 3.6 poskyriuose pateikiama kodo lankstumo užtikrinimo galimybių analizė ir pasirinktos technologijos.

3.2. Panašių sistemų analizė

Daugelis konferencijas rengiančių organizacijų tam tikslui turi savas interneto sistemas. Pavyzdžiui, VLDB konferencijų sistema gyvuoja daugelį metų. Mažesnėms konferencijoms (ISD, DBIS ir kitos) organizatoriai, kurie kasmet būna kiti, naudoja savo įrangą. Be to, yra daugelio konferencijų sistemos – IADIS, WORLDCOMP'06.

Pasaulyje yra sukurtų universalių konferencijų organizavimo įrankių, kurie nėra skirti konkrečiai konferencijai. Vieni iš jų labiau profesionalūs ir daug kainuojantys, kiti priskiriami paprastesnei kategorijai.

Pateiksime dviejų sistemų aprašymus. Pirmoji iš žemesnės kategorijos sistemų, o antroji iš profesionaliųjų.

Konferencijų organizavimo sistema ConfMaster

ConfMaster[1] – tai internetinė sistema, skirta konferencijų organizavimui.

Sistemos funkcionalumas:

- Realizuota populiariomis priemonėmis (Apache, PHP, MySQL)
- Pilnas konferencijų ruošimo ciklas
- Didelės apimties – galima valdyti iki 1000 straipsnių ir daugiau nei 2500 registruotų vartotojų
- Pagrįsta rolių-teisių koncepcija
- Redaguojamas meniu, priklausomai nuo reikalavimų
- Keičiama sistemos sąsaja
- Laiškai siunčiami pagal sukurtus šablonus
- Veikia su naršyklėmis Opera, Firefox ir Internet Explorer

Pastebėti sistemos trūkumai:

- Nėra automatizuoto etapų valdymo
- Negalima prieiti prie savo straipsnių, recenzijų
- Negalima organizuoti kelių konferencijų vienu metu

Konferencijų organizavimo sistema Confious

Kita pateikiama konferencijų organizavimo sistema – Confious[2]. Šis įrankis yra labai „galingas“, priskiriamas profesionalaus darbo įrankių kategorijai. Ši sistema naudojama kaip interneto servisas.

Sistemos funkcionalumas:

- Galima organizuoti kelias konferencijas vienu metu
- Pagrįstas rolių-teisių koncepcija
- Dvipakopis recenzavimas
 - Recenzuoja paprastas recenzentas
 - Recenzuoja patyręs recenzentas
- Operacijos atliekamos transakcijomis – užtikrinamas duomenų nuoseklumas
- Automatinis ar rankinis recenzentų priskyrimas
- Konfliktų valdymas – recenzentų, netinkančių recenzuoti priskirtą straipsnį, sąrašas bei jo valdymas
- Recenzavimo šablonų kūrimas
- Pranešimų šablonų apie priėmimą ar atmetimą kūrimas
- Etapų valdymas
- Žinučių siuntinėjimo servisas
- „Žingsnis po žingsnio“ stebėjimo servisas

Sistema yra labai funkcionali, tačiau joje pastebėti keli trūkumai:

- Nėra automatizuoto etapų valdymo
- Negalima prieiti prie savo straipsnių, recenzijų

Šiame darbe sukurtą sistemą galima priskirti prie konferencijų valdymo įrankių. Jos išskirtiniai bruožai – automatizuotas etapų valdymas, galimybė organizuoti kelias konferencijas ir naudoti kelias kalbas. Kadangi sistema skirta ir Lietuvos vartotojui, joje įdiegta lietuvių kalba. Pagrindinis jos tikslas – padėti administratoriui, todėl kitomis savybėmis ji gali nusileisti esamoms sistemoms. Pavyzdžiui, nėra kelių lygių recenzentų. Tačiau administravimo funkcijas buvo siekiama padaryti kuo geresnes.

Baigiant šį skyrelį, pateiksime esamų sistemų bei mūsų kuriamos sistemos lyginamąją lentelę. Be nagrinėtų sistemų, į ją įtrauksime ir „Profesionalų įrankį konferencijoms organizuoti“ – idealią konferencijų organizavimo sistemą, kuri turėtų visas tobulos sistemos funkcijas.

1 lentelė

	ConfMaster	Confious	Profesionalus įrankis konferencijoms organizuoti	Mūsų sukurta sistema
Pilnas konferencijų ruošimo ciklas	●	●	●	●
Pagrįsta rolių-teisių koncepcija	●	●	●	●
Keičiama sistemos sąsaja	●	●	●	●
Laiškai siunčiami sukurtus pagal šablonus	●	●	●	●
Galima organizuoti kelias konferencijas vienu metu	○	●	●	●
Dvipakopis recenzavimas	○	●	●	○
Automatinis recenzentų priskyrimas	○	●	●	○
Rankinis recenzentų priskyrimas	●	●	●	●
Konfliktų valdymas	○	●	●	○
Recenzavimo šablonų kūrimas	○	●	●	●
Pranešimų šablonų apie priėmimą ar atmetimą kūrimas	○	●	●	●
Etapų valdymas	○	●	●	●
Žinučių siuntinėjimo servisas	○	●	●	●
„Žingsnis po žingsnio“ stebėjimo servisas	○	●	●	○
Automatizuotas etapų valdymas	○	○	●	●
Priėjimas prie savo straipsnių, recenzijų	○	○	●	●
Konferencijų istorija	○	○	●	●

3.3. Interneto sistemų kūrimo technologijų analizė

Didžioji dalis interneto svetainių yra sukurtos Java kalba, naudojant karkasus. Karkasų privalumas – pateikimo atskyrimas nuo veiklos logikos. Taigi ir mūsų kuriamą sistemą buvo nuspręsta realizuoti Java kalba, o karkasui pasirinkti buvo nagrinėjami populiarūs karkasai:

- Tapestry
- Struts
- WebWork
- JSF (Java Server Faces)

Pagrindiniai kriterijai, kurie domino renkantis karkasą:

- Veiklos logikos ir vaizdo valdymo atskyrimas
- Lengvas funkcionalumo keitimas
- Lengvas vaizdo keitimas

Šiuos kriterijus atitinka tik viena iš analizuotų technologijų. Rezultatai pateikti 1 lentelėje. Buvo nuspręsta naudoti Tapestry karkasą.

2 lentelė

Kriterijus \ Technologija	Tapestry	Struts	WebWork	JSF
Veiklos logikos ir vaizdo valdymo atskyrimas	●	○	○	○
Lengvas funkcionalumo keitimas	●	●	○	○
Lengvas vaizdo keitimas	●	●	●	●

3.4. Duomenų valdymo technologijų analizė

Antras svarbus sprendimas yra duomenų saugojimo technologijos pasirinkimas ir susiejimas su veiklos logika. Kuriant informacijos teikimo ir saugojimo sistemas, viena iš pagrindinių naudojamų technologijų yra duomenų bazės. Dauguma duomenų bazių valdomos, naudojant vieningą užklausų kalbą – SQL, nors sintaksė gali šiek tiek skirtis.

Sprendžiant, kaip iš programos kodo pasiekti duomenų bazėje esančius duomenis buvo svarstomos šios galimybės:

- Savo sukurtas duomenų bazės užklausų įvalkalas(angl. *wrapper*)
- Torque[3]
- Hibernate[15]

Pirmas pasirinkimas buvo iškart atmestas, nes duomenų bazė didelė(~20 lentelių), taigi reikėjo rinktis mažiau rankinio darbo reikalaujantį variantą.

Renkantis suformuluoti šie kriterijai:

- Paprastas duomenų bazės aprašas
- Paprastas sugeneruotų klasių naudojimas
- Mažai rankinio darbo generuotoms klasėms sukurti
- Lengvas funkcionalumo keitimas

3 lentelė

Kriterijus \ Technologija	Torque	Hibernate
Paprastas duomenų bazės aprašas	●	●
Paprastas sugeneruotų klasių naudojimas	●	●
Mažai rankinio darbo generuotoms klasėms sukurti	●	●
Lengvas funkcionalumo keitimas	●	○
Minimalus kodo kiekis užklausai realizuoti	●	○
Klasių, atitinkančių duomenų bazės lenteles, generavimas	●	○

Abi technologijos atitinka tris pirmuosius kriterijus, tačiau Torque pasirinkta dėl to, jog jos sugeneruotų klasių naudojimas yra paprastesnis. Negalima teigti, jog Hibernate[15] technologijos naudojimas yra sudėtingas, tiesiog pirmąja technologija yra naudojama mažiau kodo užklausai realizuoti.

Dar vienas kriterijus nulėmęs pasirinkimą – Torque sugeneruoja klases naudojimui, o Hibernate naudoja standartines klases, o aprašus ima iš vartotojo sukurtų XML failų. Sugeneruotos klasės leidžia lengviau modifikuoti funkcionalumą.

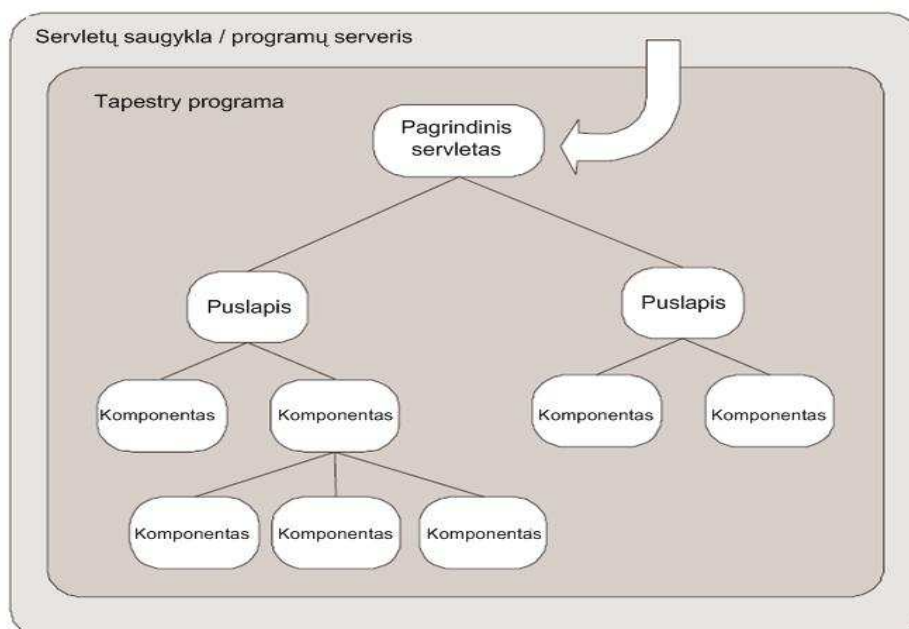
3.5. Veiklos logikos ir vaizdo valdymo atskyrimas – Tapestry karkasas

Tapestry yra atvirojo kodo karkasas, skirtas kurti dinamines, lengvai keičiamas interneto sistemas [3]. Ši technologija realizuojama Java programavimo kalba. Tapestry paremta standartine Java servletų technologija. Ji veikia bet kurioje servletų saugykloje ar programų serveryje.

Interneto sistema, sukurta naudojant Tapestry, skaidoma į apibrėžtą skaičių puslapių, kur kiekvienas puslapis gali būti sudarytas iš komponentų [4]. Komponentai gali būti sudaryti iš kitų komponentų. Pavyzdys pateiktas 2 paveiksle.

Šiuo karkasu paremtos sistemos puslapių šablonai kuriami, naudojant paprastą HTML technologiją, įterpiant nedidelius gabaliukus Java kodo. Java kodo kiekis HTML šablonuose gali būti dar labiau sumažintas, naudojant XML failus, kurie veikia kaip tarpininkai tarp HTML ir Java kodo [5].

Viena iš Tapestry karkaso privalumų yra „grynas Java kodas“ – funkcionalumas rašomas Java kodu visiškai atskirai nuo HTML.



2 pav. Tapestry karkaso komponentinis išskaidymas

Technologija pritaikyta lengvai kurti naujus komponentus, kurie gali būti pakartotinai naudojami – tai labai sumažina klaidų kiekį programoje ir pagreitina kūrimo procesą. Komponentinis programavimas leis ateityje lengvai keisti bei plėsti jau esamą sistemą.

Tapestry gali būti naudojamas tiek mažose sistemose, tiek didelės apimties projektuose.

Lengvai integruojamas su J2EE, HiveMind, JavaSpring technologijomis.

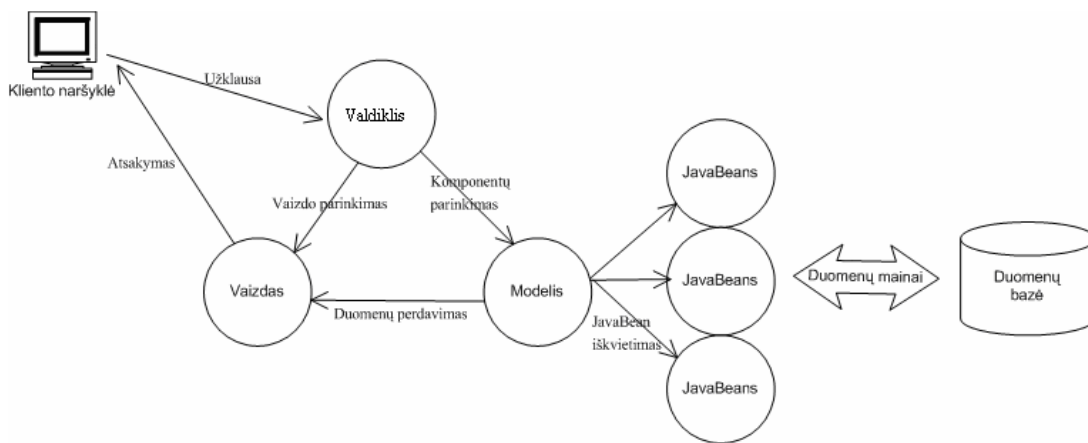
Kaip ir dauguma kitų karkasinių sistemų, taip ir Tapestry naudoja modelio-vaizdo valdiklio šabloną (MVC – *model-view-controler*) [5].

3 paveiksle pateikiame schemą, kaip veikia Tapestry karkasu paremta sistema – kaip ji bendrauja su vartotoju, veiklos logika, duomenų baze bei komponentų sąveika tarpusavyje. Sistemos veikimas remiasi MVC šablonu, kuris valdo visus procesus. Vartotojui pateikus užklausa, ją apdoroja valdiklis, iškviesdamas atitinkamą komponentą. Komponentas pateikiamas vaizdo valdikliui – čia suformuojamas dinaminis HTML puslapis iš HTML šablono bei Java kodo. Suformuotas puslapis siunčiamas atgal į vartotojo naršyklę.

Tapestry karkasą pasirinkome dėl jo pranašumų prieš kitus interneto puslapių kūrimo karkasus. Vienas iš rimtesnių kandidatų dar buvo Struts karkasas. Trumpai apžvelgsime šį karkasą bei pateiksime abiejų karkasų palyginimą.

Struts galima laikyti Java interneto projektų standartu – šiuo metu jis labiausiai paplitęs pasaulyje. Tai yra pakankamai lengvai perprantama technologija. Internete galima rasti daug literatūros apie Struts karkasą [6], [7].

Patogiai naudojamos HTML kalbos žymės (angl. *tags*). Tačiau, norint visiškai išnaudoti jų privalumus, tenka naudoti Struts žymes, o tai tampa trūkumu, neleidžiančiu atskirti kodo nuo dizaino. Sudėtingai naudojamas „ActionForms“ valdiklis, įnešantis daug neaiškumo. Tai galima paaiškinti daugybe dinaminių dalių. Sugaištama nemažai laiko, kol išsiaiškinamas veikimo principas. Kaip dar vieną technologijos trūkumą reiktų paminėti sunkų testavimą bei klaidų aptikimą. HTML puslapiuose yra nemažai kodo eilučių [8]. Kadangi Struts nėra komponentinis karkasas, tai apsunkina sistemos komponentų plečiamumą.



3 pav. Tapestry karkasu paremtos sistemos komponentų sąveika

3.6. Veiklos logikos ir duomenų nepriklausomumas – Torque technologija

Torque technologija leidžia manipuluoti duomenimis iš duomenų bazės, naudojantis Java kalba aprašytais objektais. Kitaip nei kitos technologijos, Torque nenaudoja jau sukurtų sąsajų, kad pasiektų vartotojo pateiktas klases, bet jas susigeneruoja. Į sugeneruotas klases yra įtraukiami ir duomenų objektai (angl. *Data Objects*). Klasės generuojamos naudojant XML schemas aprašus. Aprašai gali būti sukurti ranka ar sugeneruoti, panaudojant jau esamą duomenų bazę. XML schema gali būti panaudota sukurti vykdomąsias užklausas. Šiomis SQL užklausomis galima sukurti tuščią duomenų bazės schemą su aprašytais lentelėmis.

Naudodamas šią technologiją, programuotojas tiesiogiai neprieina prie užklausų, naudojamų duomenų bazėse. Taigi, Torque leidžia sukurti programą nepriklausomą nuo duomenų bazės. Pagrindinėmis priemonėmis galima sugeneruoti prieigos klases daugumai duomenų bazių, nebent duomenų bazė turi ypatingų savybių. Jei duomenų bazė turi ypatingų savybių, tuomet programuotojui reikia susikurti įrankį Torque klasėms generuoti.

Sugeneruotų klasių naudojimas suteikia galimybę lengvai pritaikyti duomenų bazės pritaikymą savo programoje. Tai atliekama persirašant sugeneruotus metodus ir taip keičiant jų veikimą.

Torque technologija buvo vystoma kaip Turbine karkaso dalis. Tačiau dabar ši technologija yra atskirta yra galima ją naudoti atskirai. O pačiame Turbine karkase ši technologija taip pat yra naudojama, bet jau kaip atskiras sistemos komponentas.

3.7. Analizės išvados

Siekiant sukurti konferencijų organizavimo sistemą, pirmiausia buvo analizuojami užsakovo poreikiai ir esamų konferencijų sistemų savybės. Specifiniai reikalavimai, keliami kuriamai sistemai – galimybė organizuoti konferencijas keliomis kalbomis, tarp jų ir lietuvių kalba, saugoti konferencijų istoriją, automatizuoti administravimo veiklą; lengva priežiūra ir plečiamumas. Automatizuota administravimo veikla – tai konferencijos etapų valdymas, recenzijų vertinimo sistema, automatinis laiškų siuntimas vartotojams, laiškų siuntimas nurodytai dalyvių grupei. Nei viena iš esamų sistemų pilnai šių reikalavimų netenkina.

Atlikus konferencijų organizavimo veiklos kompiuterizavimo ir automatizavimo padidinimo galimybių analizę, išryškėjo dvi problemos: išskirti ir sumodeliuoti veiklos proceso dalis, kurias sistema galėtų vykdyti automatiškai, bei pasirinkti tokią kūrimo technologiją, kuri užtikrintų pateikimo, veiklos logikos ir duomenų saugojimo atskyrimą, lankstų visų dalių keitimą, minimalų ranka rašomo kodo kiekį ir kitas panašias savybes, kurios leistų efektyviai kurti sistemą ir modifikuoti ją ateityje.

Sistemos platforma pasirinkta Java kalba – šiuo metu stipriausias konkurentas lyginant su kitomis platformomis. Šiai platformai pritaikytų karkasų analizės metu išrinktas Tapestry karkasas, kuris geriausiai užtikrina veiklos logikos ir vaizdo valdymo atskyrimą bei nepriklausomą šių kodo dalių keitimą. Be to, tai objektinė, komponentais paremta technologija.

Duomenų ir veiklos logikos atskyrimui išrinkta Torque technologija, kuri visiškai automatizuoja sąveiką su duomenų baze, be to, sugeneruoja duomenų klases ir reikalauja minimalaus ranka rašomo kodo.

Šių technologijų naudojimas pagerins programos suprantamumą ir sumažins rankinį kūrėjo darbą, didelę dalį veiklos perduodant pasirinktiems įrankiams.

Originali darbo dalis bus skirta automatizuojamiems procesams, kuriems bus sudaryti parametrizuoti modeliai.

4. KONFERENCIJŲ SISTEMOS MODELIS

4.1. Konferencijų sistemos reikalavimai

Kuriant konferencijos sistemą buvo keliami funkciniai ir nefunkciniai reikalavimai. Nefunkcinius aprašysime trumpai, o funkcinis panagrinėsime detaliau. Taipogi buvo ir esminių reikalavimų, kurie nulėmė visos sistemos modelio koncepciją:

- Daugiakalbė vartotojo sąsaja
- Daugelio konferencijų sistema

Daugiakalbė vartotojo sąsaja leidžia vartotojui pasirinkti jam suprantamą kalbą, kuria sistema pateikiama.

Daugelio konferencijų sistema suteikia galimybę vienu metu organizuoti kelias konferencijas. Sistemos realizacija, naudojant rolių-teisių koncepciją, leidžia lengvai identifikuoti vartotoją, rengiant kelias konferencijas

Sistema turi leisti dinamiškai kurti straipsnių vertinimo kriterijus, etapus. Etapų valdymas turi būti automatinis, paliekant ir rankinio valdymo galimybę.

Recenzijų vertinimo sistema – skaičiuojanti ir pateikianti recenzijų įvertinimo rezultatus autoriui ir administratoriui.

Pateiksime nefunkcinius reikalavimus:

- Reikalavimai išvaizdai:
 - Lengvai suprantama sąsaja
 - Lengvas valdymo meniu
 - Informatyvus duomenų pateikimas

- Reikalavimai panaudojamumui

Sistemos vartotojai neprivalo būti IT specialistai, kadangi meniu paprastas, nereikalaujantis specialių žinių. Sistemos administratorius privalo turėti žinių, kad gerai atliktų savo darbą, tačiau ir administravimas turi būti paprastas. Darbas atliekamas pelės bei klaviatūros pagalba.

- Reikalavimai saugumui

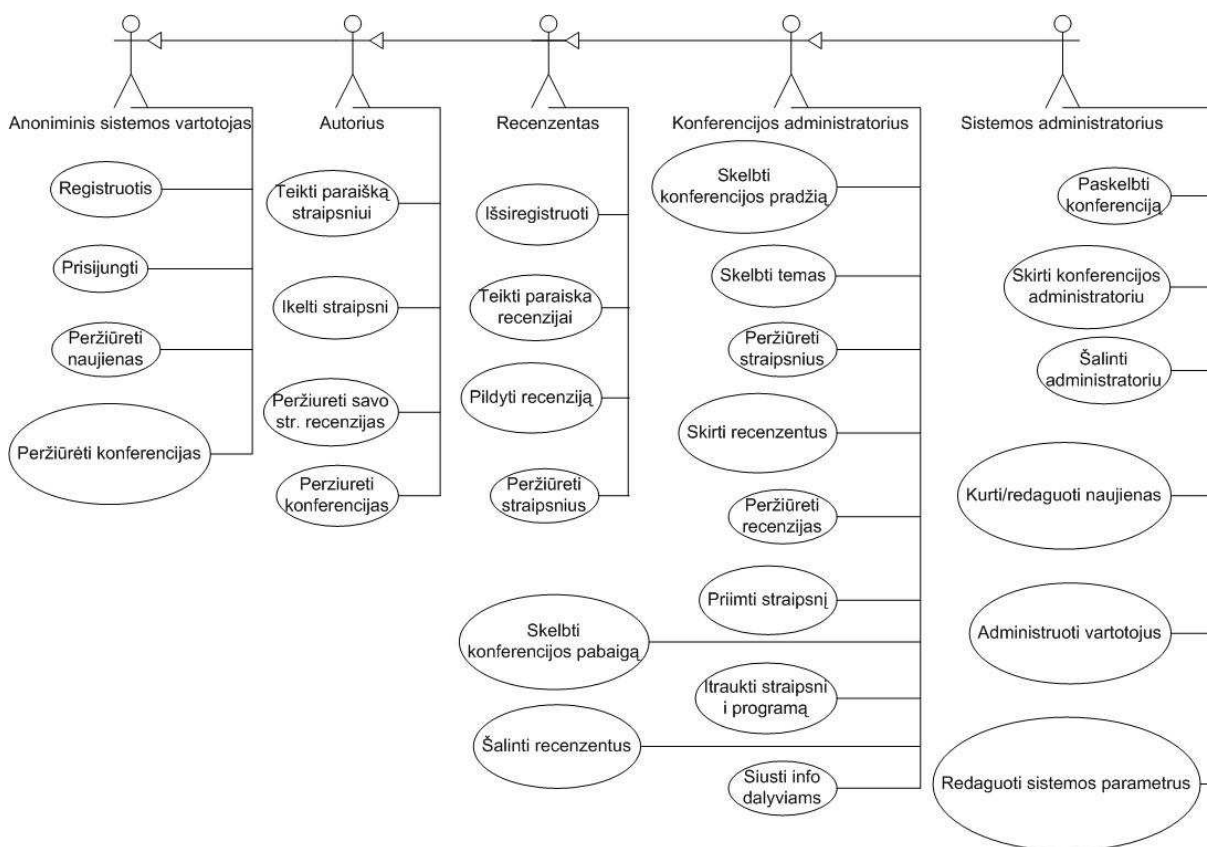
Duomenų saugumas bus užtikrintas – neautorizuoti vartotojai negalės atlikti jokių valdymo veiksmų. O užrakintų duomenų negalės pakeisti net ir

administratorius. Slaptažodžių ir vartotojų vardų apsaugai bus naudojamos standartinės apsaugos priemonės.

Konferencijų sistemos naudotojus pagal rolių-teisių koncepciją suskirstysime į penkis vartotojų tipus:

- Anoniminis vartotojas
- Autorius
- Recenzentas
- Konferencijos administratorius
- Sistemos administratorius

Šiame skyrelyje pateiksime jų atliekamų funkcijų grafinį vaizdavimą (panaudojimo atvejų modelį) bei trumpai aprašysime kiekvieną iš jų. Grafinis vaizdas pateiktas 4 paveiksle.



4 pav. Konferencijų sistemos panaudojimo atvejų modelis

Žemiau pateikiami panaudojimų atvejų aprašai.

1. Panaudojimo atvejis: Peržiūrėti naujienas

Vartotojas/Aktorius:	Anoniminis vartotojas, recenzentas, autorius, konferencijos administratorius, sistemos administratorius
Aprašas:	Peržiūrėjimas naujienų, pateiktų sistemos pirmajame puslapyje.
Prieš sąlyga:	Bent viena naujieną sukurta sistemoje.
Sužadinimo sąlyga:	Atsidaromas sistemos puslapis interneto naršyklėje.
Po-sąlyga:	-

2. Panaudojimo atvejis: Peržiūrėti konferencijas

Vartotojas/Aktorius:	Anoniminis vartotojas, recenzentas, autorius, konferencijos administratorius, sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas turi galimybę peržiūrėti vykstančias bei pasibaigusias konferencijas.
Prieš sąlyga:	Bent viena konferencija sukurta sistemoje.
Sužadinimo sąlyga:	Sužadintas konferencijų iškviatimo meniu punktas.
Po-sąlyga:	-

3. Panaudojimo atvejis: Registruotis

Vartotojas/Aktorius:	Anoniminis vartotojas
Aprašas:	Apima procesą, kurio metu vartotojas užregistruoja savo duomenis sistemoje.
Prieš sąlyga:	Tokio vartotojo dar nėra.
Sužadinimo sąlyga:	Sužadintas registracijos meniu punktas.
Po-sąlyga:	Vartotojas užregistruotas sistemoje. Laukiama administratoriaus patvirtinimo.

4. Panaudojimo atvejis: Prisijungti

Vartotojas/Aktorius:	Anoniminis vartotojas
Aprašas:	Apima procesą, kurio metu vartotojas prisijungia prie sistemos.
Prieš sąlyga:	Vartotojas dar neprisijungęs.
Sužadinimo sąlyga:	Sužadintas prisijungimo meniu punktas.
Po-sąlyga:	Vartotojas prijungtas prie sistemos. Atidaromi meniu punktai pagal nustatytas teises.

5. Panaudojimo atvejis: Teikti paraišką straipsniui

Vartotojas/Aktorius:	Recenzentas, autorius, konferencijos administratorius, sistemos administratorius
Aprašas:	Apima procesą, kurio metu vartotojas pateikia paraišką rašyti straipsnį.
Prieš sąlyga:	Vartotojas yra prisijungęs prie sistemos.
Sužadinimo sąlyga:	Sužadinta paraiškos teikimo nuoroda temos sąrašė.
Po-sąlyga:	Paraiška užregistruota sistemoje. Vartotojui aktyvuojamas punktas „Mano straipsniai“.

6. Panaudojimo atvejis: Įkelti straipsnį

Vartotojas/Aktorius:	Autorius
Aprašas:	Apima procesą, kurio metu vartotojas įkelia straipsnį į sistemą.
Prieš sąlyga:	Vartotojas yra prisijungęs prie sistemos. Buvo pateikęs paraišką straipsniui rašyti. Sistema leidžia įkelti straipsnius.
Sužadinimo sąlyga:	Sužadintas straipsnio įkėlimo meniu punktas.
Po-sąlyga:	Straipsnis įkeltas į sistemą.

7. Panaudojimo atvejis: Peržiūrėti savo straipsnių recenzijas

Vartotojas/Aktorius:	Autorius
Aprašas:	Apima procesą, kurio metu vartotojas peržiūri savo straipsnio recenziją.
Prieš sąlyga:	Vartotojas yra prisijungęs prie sistemos. Baigtas straipsnių recenzavimo etapas.
Sužadinimo sąlyga:	Sužadinta straipsnio recenzijos peržiūros nuoroda.
Po-sąlyga:	-

8. Panaudojimo atvejis: Teikti paraišką recenzijai rašyti

Vartotojas/Aktorius:	Recenzentas
Aprašas:	Apima procesą, kurio metu vartotojas pateikia paraišką recenzijai rašyti.
Prieš sąlyga:	Vartotojas yra prisijungęs prie sistemos. Suteiktos recenzento teisės.
Sužadinimo sąlyga:	Sužadinta paraiškos recenzijai rašyti nuoroda straipsnių sąrašė.
Po-sąlyga:	Paraiška užregistruota sistemoje. Vartotojui aktyvuojamas punktas „Mano recenzijos“.

9. Panaudojimo atvejis: Peržiūrėti straipsnius

Vartotojas/Aktorius:	Recenzentas, konferencijos administratorius, sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas turi galimybę peržiūrėti konferencijų straipsnius.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos.
Sužadinimo sąlyga:	Sužadintas konferencijų temų meniu punktas.
Po-sąlyga:	-

10. Panaudojimo atvejis: Pildyti recenziją

Vartotojas/Aktorius:	Recenzentas
Aprašas:	Apima procesą, kurio metu rašoma recenzija konkrečiam darbui.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Sistema leidžia recenzijų įkėlimą.
Sužadinimo sąlyga:	Sužadintas „Mano recenzijos“ punkte esančios, recenzijos įkėlimo nuoroda.
Po-sąlyga:	Recenzija įkelta į sistemą.

11. Panaudojimo atvejis: Peržiūrėti recenzijas

Vartotojas/Aktorius:	Konferencijos administratorius, sistemos administratorius.
Aprašas:	Procesas, kurio metu vartotojas gali peržiūrėti konferencijų straipsnių recenzijas.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Baigtas recenzijų įkėlimo etapas.
Sužadinimo sąlyga:	Sužadinta konferencijų straipsnių recenzijų nuoroda.
Po-sąlyga:	-

12. Panaudojimo atvejis: Skelbti temas

Vartotojas/Aktorius:	Konferencijos administratorius, sistemos administratorius.
Aprašas:	Procesas, kurio metu vartotojas sukuria temas konferencijai.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Sukurta konferencija.
Sužadinimo sąlyga:	Sužadinta konferencijos temų kūrimo nuoroda.
Po-sąlyga:	Sukurtos temos konferencijai.

13. Panaudojimo atvejis: Įtraukti straipsnį į programą

Vartotojas/Aktorius:	Konferencijos administratorius, sistemos administratorius.
Aprašas:	Procesas, kurio metu vartotojas įtraukia straipsnį į konferencijos programą.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Straipsnis įkeltas į sistemą. Baigti straipsnių įkėlimo ir recenzijų įkėlimo etapai.
Sužadinimo sąlyga:	Sužadinta konferencijos programos sudarymo nuroda.
Po-sąlyga:	Programa sudaryta.

14. Panaudojimo atvejis: Siųsti informaciją dalyviams

Vartotojas/Aktorius:	Konferencijos administratorius, sistemos administratorius.
Aprašas:	Procesas, kurio metu vartotojas sukuria pranešimus siųsti konferencijos dalyviams.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Sukurta konferencija.
Sužadinimo sąlyga:	Sužadintas žinučių kūrimo meniu punktas.
Po-sąlyga:	Sukurta žinutė. Ji nustatytu laiku bus išsiųsta nurodytiems konferencijos dalyviams.

15. Panaudojimo atvejis: Šalinti recenzentus

Vartotojas/Aktorius:	Konferencijos administratorius, sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas panaikina konferencijos dalyviui galimybę recenzuoti straipsnį.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Dalyviui suteiktos recenzento teisės.
Sužadinimo sąlyga:	Atidaromas dalyvių redagavimo langas.
Po-sąlyga:	Dalyvis nebegali recenzuoti nurodyto straipsnio.

16. Panaudojimo atvejis: Skirti recenzentus

Vartotojas/Aktorius:	Konferencijos administratorius, sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas suteikia konferencijos dalyviui galimybę recenzuoti straipsnį.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Dalyvis dar nerecenzuoja nurodyto straipsnio.
Sužadinimo sąlyga:	Atidaromas dalyvių redagavimo langas.
Po-sąlyga:	Dalyviui suteiktos nurodyto straipsnio recenzavimo teisės.

17. Panaudojimo atvejis: Skelbti konferencijos pradžią

Vartotojas/Aktorius:	Konferencijos administratorius, sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas pradeda konferencijos organizavimą.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Konferencijos organizavimas dar nepradėtas.
Sužadinimo sąlyga:	Atidaromas konferencijų valdymo langas.
Po-sąlyga:	Konferencija pradėta. Leidžiamas straipsnių paraiškų registravimas.

18. Panaudojimo atvejis: Skelbti konferencijos pabaigą

Vartotojas/Aktorius:	Konferencijos administratorius, sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas baigia konferencijos organizavimą.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės.
Sužadinimo sąlyga:	Atidaromas konferencijų valdymo langas.
Po-sąlyga:	Konferencija baigta. Ji matoma konferencijų istorijos lange. Visi informacijos įkėlimo procesai negalimi.

19. Panaudojimo atvejis: Redaguoti sistemos parametrus

Vartotojas/Aktorius:	Sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas redaguoja sistemos parametrus.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės.
Sužadinimo sąlyga:	Atidaromas sistemos parametrų valdymo langas.
Po-sąlyga:	Pakeisti sistemos parametrai.

20. Panaudojimo atvejis: Kurti/Redaguoti naujienas

Vartotojas/Aktorius:	Sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas kuria/redaguoja sistemos naujienas.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės.
Sužadinimo sąlyga:	Sužadinamas naujienų redagavimo meniu punktas.
Po-sąlyga:	Sukurta/pakeista naujiena.

21. Panaudojimo atvejis: Administruoti vartotojus

Vartotojas/Aktorius:	Sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas tvarko sistemos vartotojų teises, informaciją.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės.
Sužadinimo sąlyga:	Atidaromas vartotojų valdymo langas.
Po-sąlyga:	Sukurtas naujas vartotojas arba pakeista informacija apie vartotoją, arba suteiktos teisės vartotojui.

22. Panaudojimo atvejis: Paskelbti konferenciją

Vartotojas/Aktorius:	Sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas sukuria konferenciją sistemoje.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Tokia konferencija neegzistuoja.
Sužadinimo sąlyga:	Atidaromas konferencijų valdymo langas.
Po-sąlyga:	Sukurta nauja konferencija.

23. Panaudojimo atvejis: Skirti konferencijos administratorių

Vartotojas/Aktorius:	Sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas priskiria administratorių pasirinktai konferencijai.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Konferencijai nepriskirtas administratorius.
Sužadinimo sąlyga:	Atidaromas pasirinktos konferencijos redagavimo langas.
Po-sąlyga:	Priskirtas administratorius konferencijai.

24. Panaudojimo atvejis: Šalinti konferencijos administratorių

Vartotojas/Aktorius:	Sistemos administratorius
Aprašas:	Procesas, kurio metu vartotojas pašalina administratorių pasirinktai konferencijai.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Administratorius priskirtas konferencijai.
Sužadinimo sąlyga:	Atidaromas pasirinktos konferencijos redagavimo langas.
Po-sąlyga:	Pašalintas pasirinktos konferencijos administratorius.

25. Panaudojimo atvejis: Priimti straipsnį

Vartotojas/Aktorius:	Konferencijos administratorius, sistemos administratorius.
Aprašas:	Procesas, kurio metu vartotojas priima straipsnį.
Prieš sąlyga:	Vartotojas prisijungęs prie sistemos. Suteiktos administratoriaus teisės. Straipsnis įkeltas į sistemą. Straipsnis recenzuotas bei įvertintas. Baigti straipsnių įkėlimo ir recenzijų įkėlimo etapai.
Sužadinimo sąlyga:	Sužadinta konferencijos straipsnių priėmimo nuoroda.
Po-sąlyga:	Straipsnis priimtas.

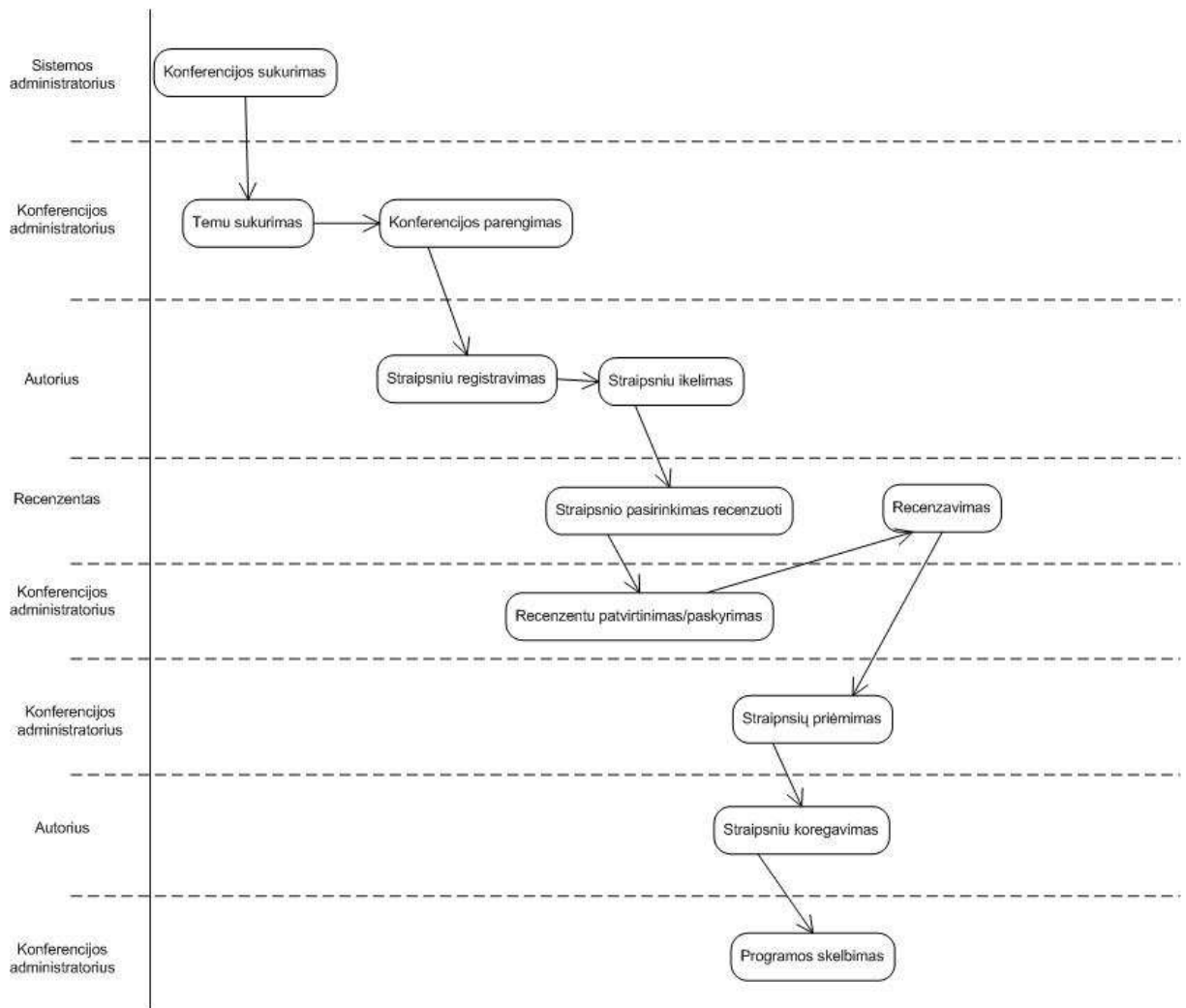
4.2. Konferencijų veiklos procesų modelis

Konferencijos veiklos procese galima išskirti tipinius etapus, kurie skirtingose konferencijose jungiami į bendrą procesą įvairiais būdais, tačiau proceso etapų tvarka panaši. Proceso etapus atlieka tipiniai vartotojai. Tam tikrais atvejais vieni ar kiti procesai gali būti sukeisti vietomis ar iš viso nenaudojami.

5 paveiksle pateikiamas konferencijos procesas, išskaidytas į smulkesnius etapus, nurodomi jo atlikėjai.

Veikėjų grupės yra keturios – tai sistemos administratorius, konferencijos administratorius, autorius ir recenzentas. Sistemos administratorius valdo tik konferencijos sukūrimą bei administratoriaus paskyrimą. Konferencijos administratorius kuria temas, paruošia konferencijos sistemą darbui, skiria ir tvirtina recenzentus, priima straipsnius ir skelbia konferencijos programą. Autorius registruoja, įkelia ir redaguoja straipsnį, na o recenzentas recenzuoja bei vertina straipsnį. Visų veikėjų atliekami darbai sugrupuojami ir pateikiami kitu pjūviu.

Konferencijos veiklos procese pateikti svarbiausi etapai.



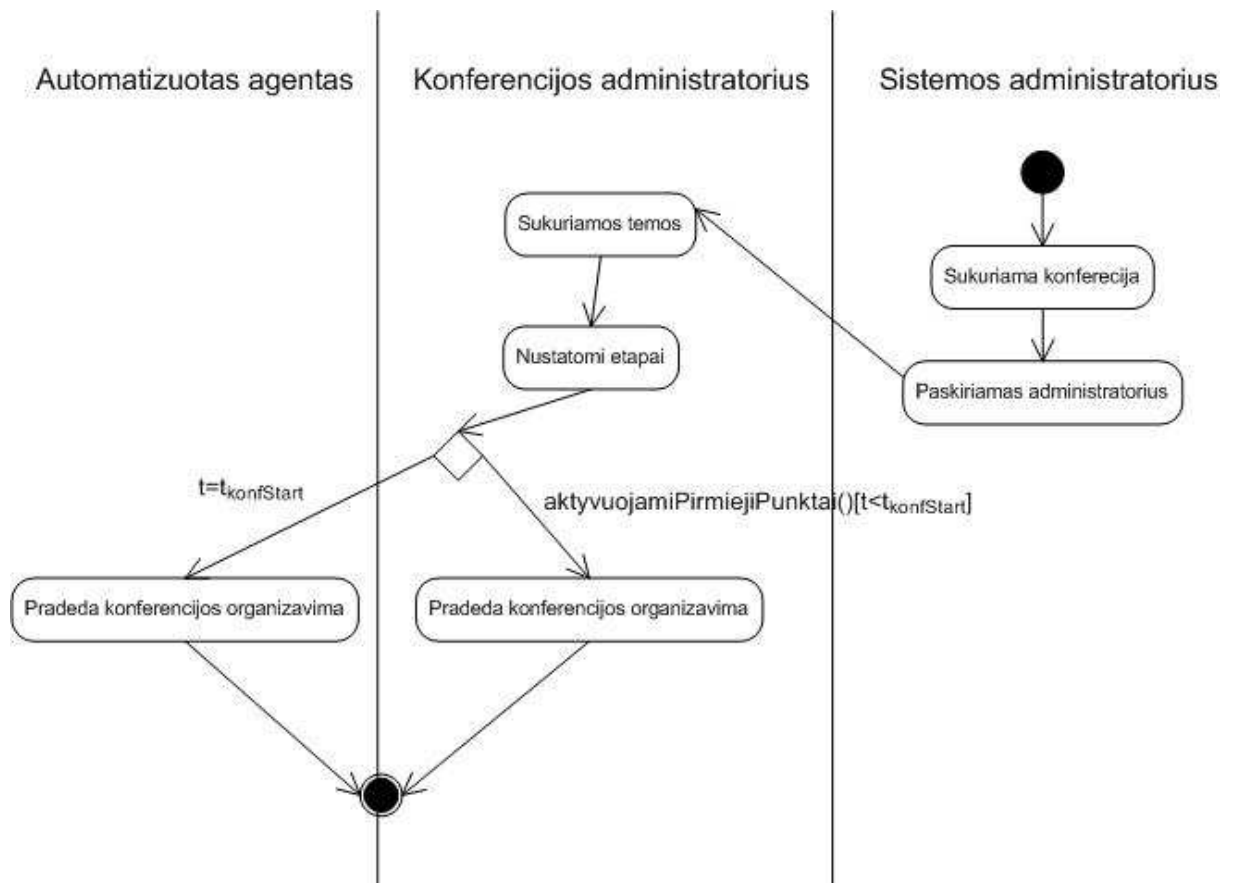
5 pav. Konferencijų veiklos procesai

Bendrą konferencijų procesą galima išskaidyti į tris dalis:

- Konferencijos sukūrimas
- Konferencijos organizavimas
- Konferencijos uždarymas

Detalizuotas konferencijos skelbimo procesas pateiktas 6 paveiksle. Šiame procese dalyvauja trijų tipų veikėjai – sistemos administratorius, konferencijos administratorius ir automatizuotas agentas. Nenaudojant automatizuoto agento, visi darbai būtų atliekami rankiniu būdu.

Konferencijos sukūrimu bei administratoriaus paskyrimu rūpinasi sistemos administratorius, na o etapų nustatymu, recenzijų šablonų kūrimu bei informacijos įrašymu rūpinasi paskirtas konferencijos administratorius.



6 pav. Konferencijų sukūrimo/skelbimo procesas

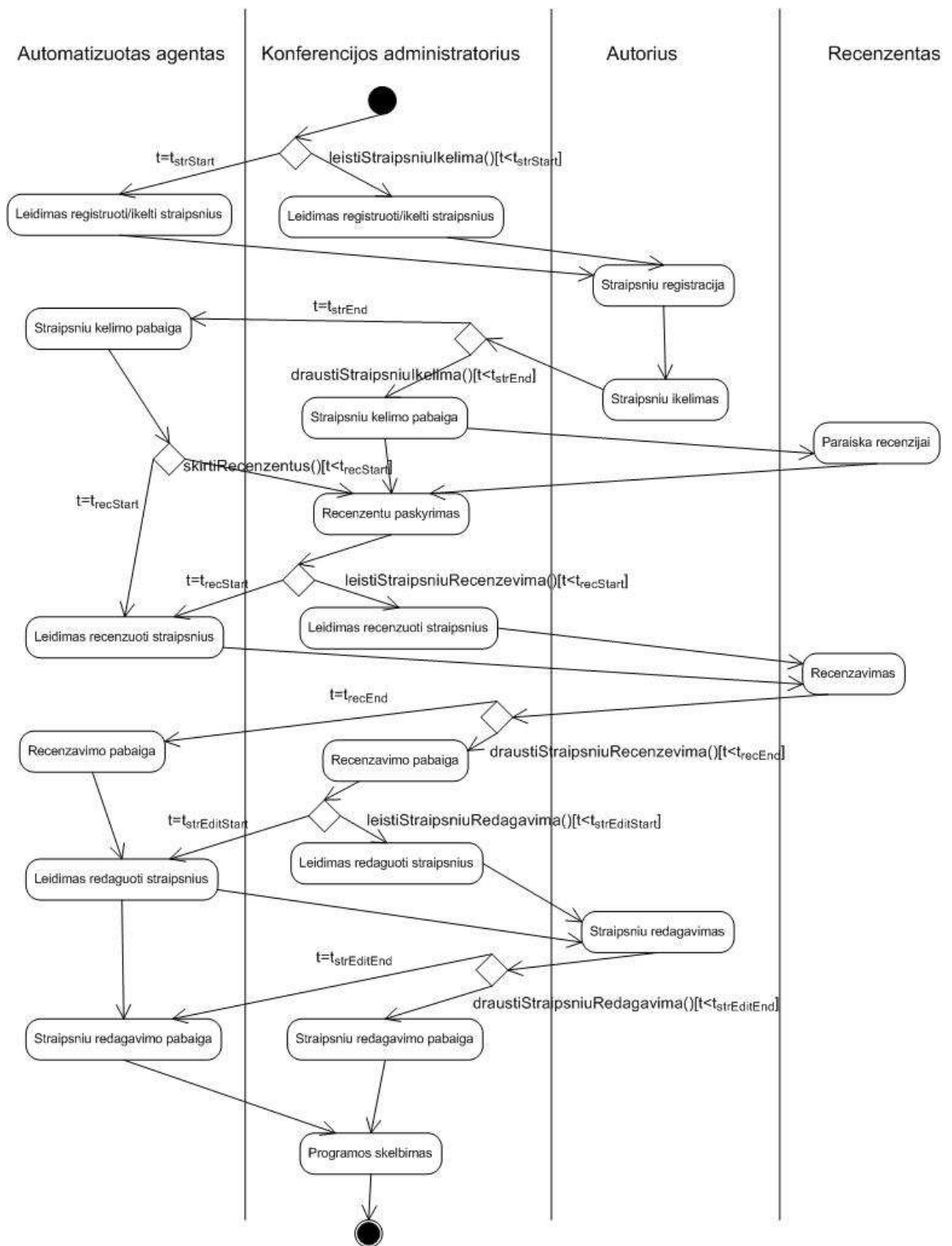
Didžioji konferencijos organizavimo laiko dalis yra sunaudojama straipsnių, recenzijų kėlimui į sistemą. Šio etapo procesas pateiktas 7 paveiksle.

Šiame procese dalyvauja konferencijos administratorius, autoriai, recenzentai. Lygiagrečiai šiam procesui veikia automatizuotas agentas.

Daugelį konferencijos administratoriaus atliekamų veiksmų yra valdomi automatizuoto agento, tačiau paliekama galimybė tuos veiksmus valdyti rankiniu būdu. Kaip ir matosi paveiksle, administratoriaus veikla sudaro didžiausią proceso veiklos dalį. Autoriams ir recenzentams reikia atlikti žymiai mažiau veiksmų..

Kadangi administratorius turi atlikti administravimo darbus tam tikrais laiko momentais, priklausomai nuo nustatytų konferencijos pradžios, straipsnių pateikimo pabaigos, recenzavimo pabaigos ir kitų datų, automatizuotas agentas irgi turi atlikti jam nustatytus veiksmus tiksliai nurodytu laiku..

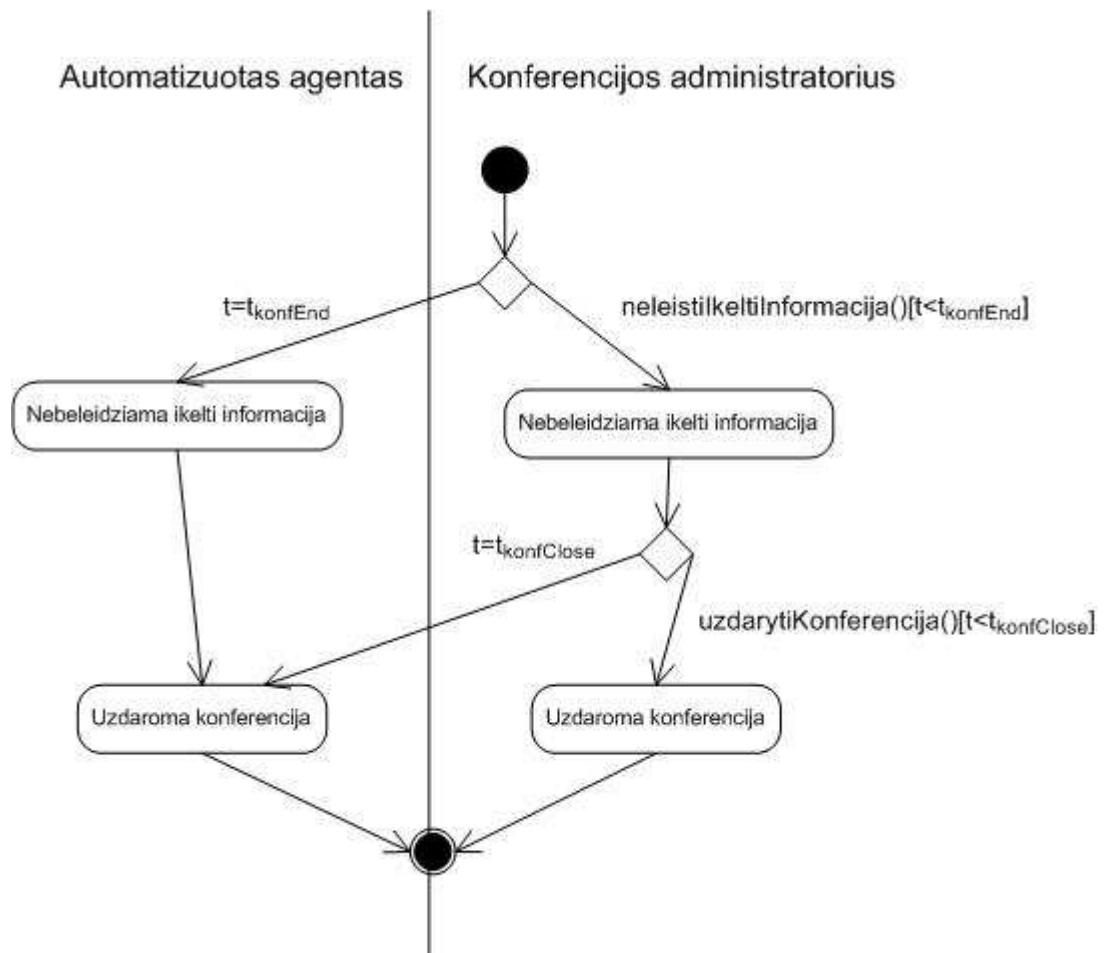
Matome, kad didžioji veiklos dalis gali būti perduota agentui. Jei iki nustatyto laiko nebus atliekamas etapo valdymo žingsnis rankiniu būdu, jį atliks automatizuotas agentas. Kai kurie etapai turi būti atlikti rankiniu būdu. Šie etapai turi būti atlikti savo laiku, kas automatizuotas agentas nepaleistų kito etapo.



7 pav. Konferencijų pagrindinis organizavimo procesas – straipsnių, recenzijų įkėlimas

Konferencijos užbaigimo procesas pateiktas 8 paveiksle. Šis procesas gali būti atliekamas konferencijos administratoriaus arba automatizuoto agento.

Pasibaigus konferencijos organizavimui, tai yra baigiami visi paruošiamieji darbai, kurių metu į sistemą buvo sukelta visa reikalinga informacija. Ši informacija yra paruošiama saugojimui. Reikalinga informacija paimama konferencijai. Uždaromi visi informacijos keitimo, įkėlimo etapai. Na ir galiausiai sistema pažymima kaip baigta ir ji tampa aktyvi peržiūrai, bet ne keitimui.



8 pav. Konferencijos užbaigimo procesas

Administratoriaus sprendimu, konferencijos informacija gali būti prieinama uždaram ratui žmonių arba prieinama visiems.

4.3. Veiklos klasių modelis

Šiame skyrelyje pateikiami sistemos veiklos klasių modeliai. Pateikiama, kaip valdymo klasės naudoja esybių klases.

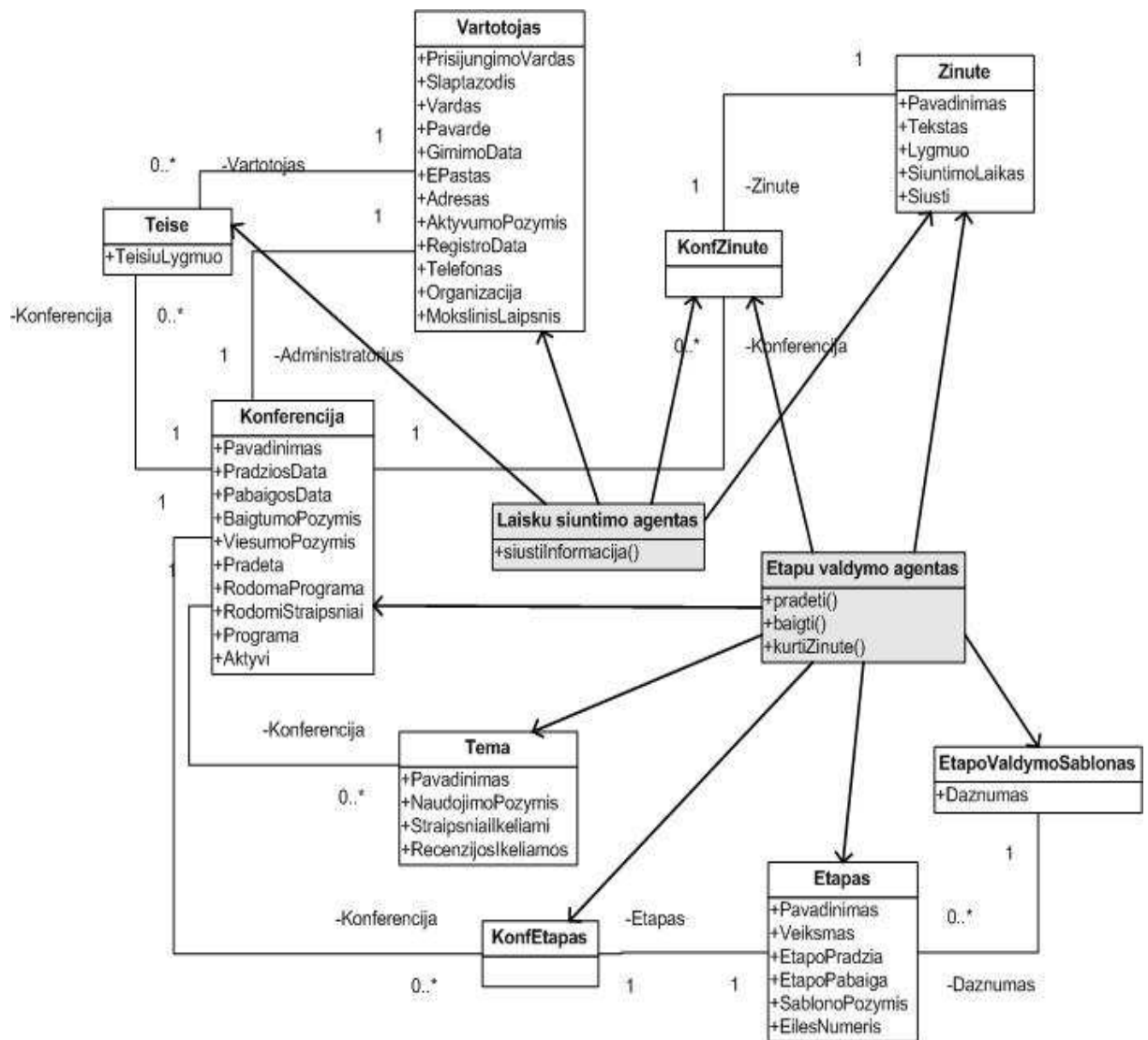
Viena iš svarbiausių sistemos valdymo klasių – tai automatizuotų agentų valdymo klasės. Automatizuoti agentai yra dviejų tipų – laiškų siuntimo agentas ir etapų valdymo agentas.

Laiškų siuntimo agentas rūpinasi informacijos siuntimu vartotojams. Ji siunčiama priklausomai nuo nustatyto dažnumo bei žinučių šablonų, atsižvelgiant į vartotojų teises konkrečioje konferencijoje. Antrasis agentas valdo konferencijos organizavimo etapus, ruošia informaciją siuntimui pirmajam agentui tinkama forma. Etapai valdomi pagal nustatytus parametrizuotus šablonus. Jų veiklos rezultate keičiasi informacija duomenų bazėje.

. Kadangi abu agentai glaudžiai susiję, tai ir jų realizacijas buvo prasminga sujungti. Realizacijos lygmenyje veikia vienas agentas, kuris apima dviejų loginių agentų veikimą

Automatizuotų agentų veiklos modelis pateiktas 9 paveiksle.

Šių agentų veikimas pagrįstas laikmačio principu. Kas nustatytą laiko intervalą yra tikrinama informacija. Ji apdorojama ir, jei reikia, sukuriama žinutė, pakeičiama informacija duomenų bazėje. Taip pat kas nustatytą laiko intervalą tikrinama siuntimo informacija. Jei informacija paruošta siuntimui – surenkami duomenys apie vartotojus, kuriems ji bus siunčiama, suformuotas laiškas išsiunčiamas vartotojams.



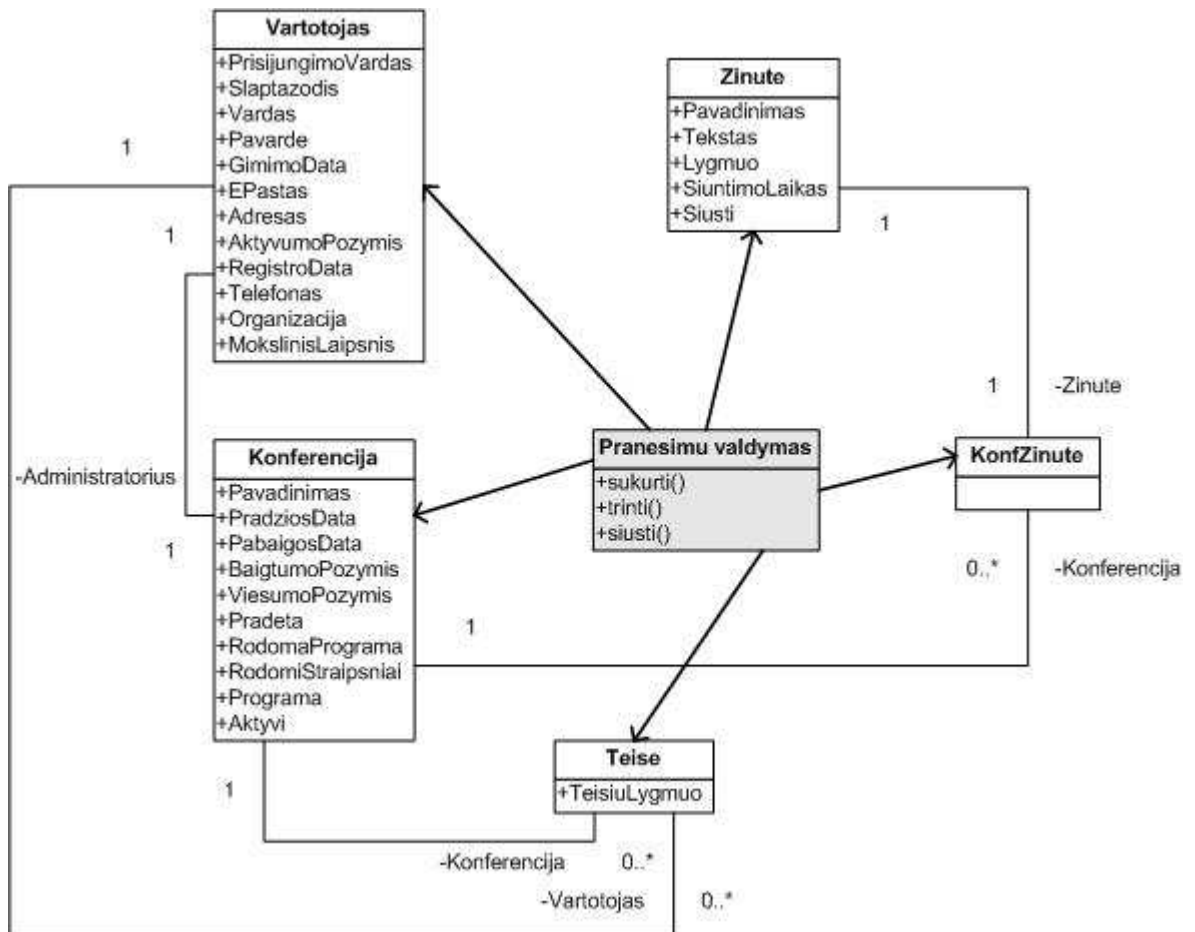
9 pav. Automatizuotų agentų veiklos klasių modelis

Žemiau pateikti veiklos modeliai nėra automatizuoti laiko atžvilgiu, juos turi atlikti administratorius rankiniu būdu.

Pranešimų valdymo modeliu pateikta informacijos kūrimo koncepcija. Administratorius sukuria žinutės tekstą ir pasirenka iš sąrašų interesantų grupes. Informacija yra išsaugoma laiškų agentui suprantamu formatu ir nustatytoje lentoje.

Automatizuotas laiškų siuntimo agentas pasirūpina žinutės išsiuntimu vartotojams.

Etapų valdymo agento sukurtas žinutes gali matyti ir administratorius bei jas valdyti. Jis gali jas laikinai stabdyti, trinti ar pagedaguoti siunčiamą informaciją. Tai yra suteikiamas lankstumas šabloniškam veikimui. Kadangi ne visi etapai yra universalūs ir siunčiama informacija kartais būna nestandartinė, taigi šis bendradarbiavimas tarp automatizuoto agento ir administratoriaus suteikia lankstumo. 10 paveikslėlyje pateiktas pranešimų valdymo veiklos modelis.



10 pav. Pranešimų valdymo veiklos klasių modelis

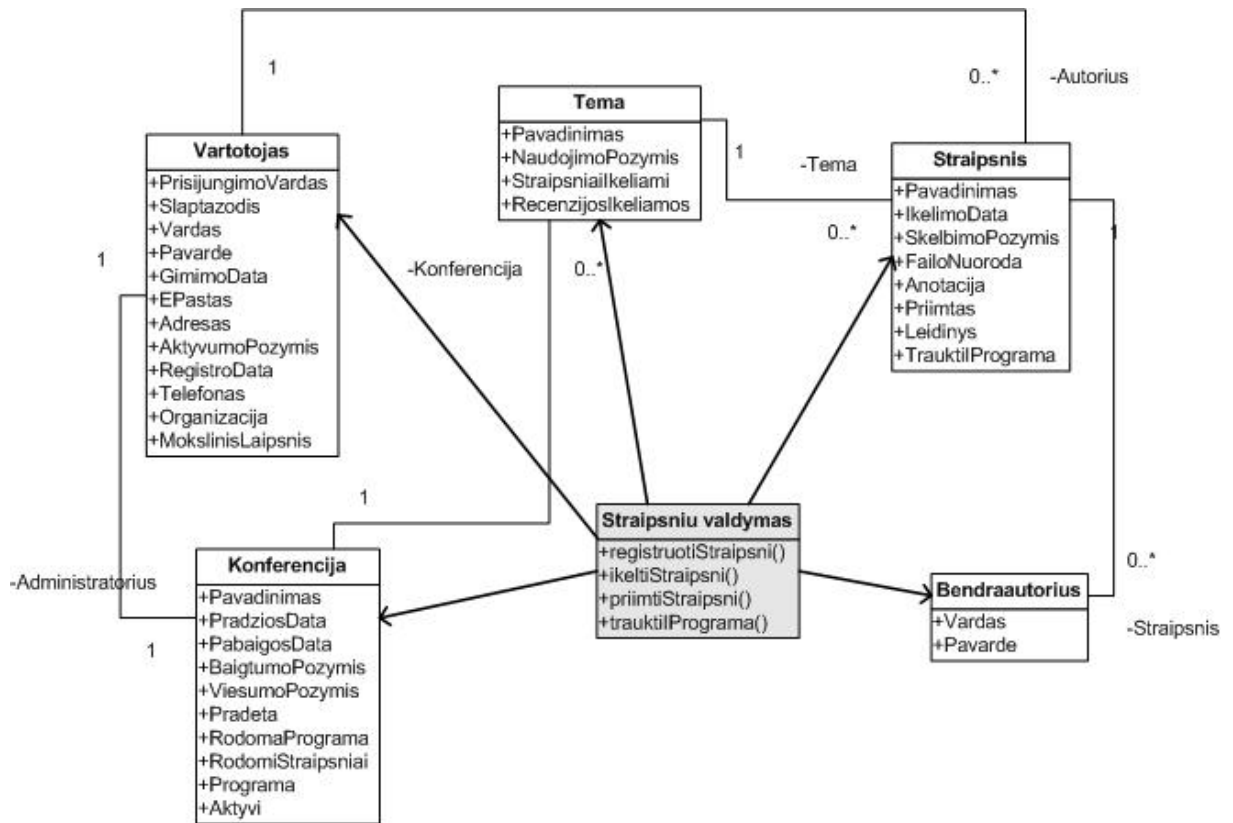
Prieš tai aptarti du veiklos modeliai yra tarpusavyje susiję. Parodyta galimybė suderinti automatizuotą darbą su žmogaus protine veikla.

Žemiau pateikiami du modeliai, skirti apibrėžti esminius konferencijos rengimo etapus, tai straipsnių ir recenzijų valdymas.

Pirmiausia, pateikiamas straipsnių valdymo modelis. Šis modelis pateiktas 11 paveiksle. Straipsnių valdymo modelyje yra apibrėžti dviejų atlikėjų veikla – tai autorius ir konferencijos administratorius.

Autorius registruoja straipsnį konkrečioje konferencijoje bei pasirinktoje temoje. Vėliau straipsnis talpinamas į sistemą. Po šių veiklos etapų yra įtraukiamas recenzavimo etapų valdymas. Tačiau kol kas priimama, jog recenzavimas yra aiškus, bet žemiau pateiksime jo detalizavimą su modeliu. Po recenzavimo straipsnis yra pakartotinai įkeliamas, taip atliekamas jo redagavimas.

Pasibaigus autoriaus veiklos etapams, prasideda administratoriaus veikla. Straipsnio priėmimo etapas įsiterpia po recenzavimo pabaigos, straipsnio traukimu į programą, baigiamas straipsnių valdymas.



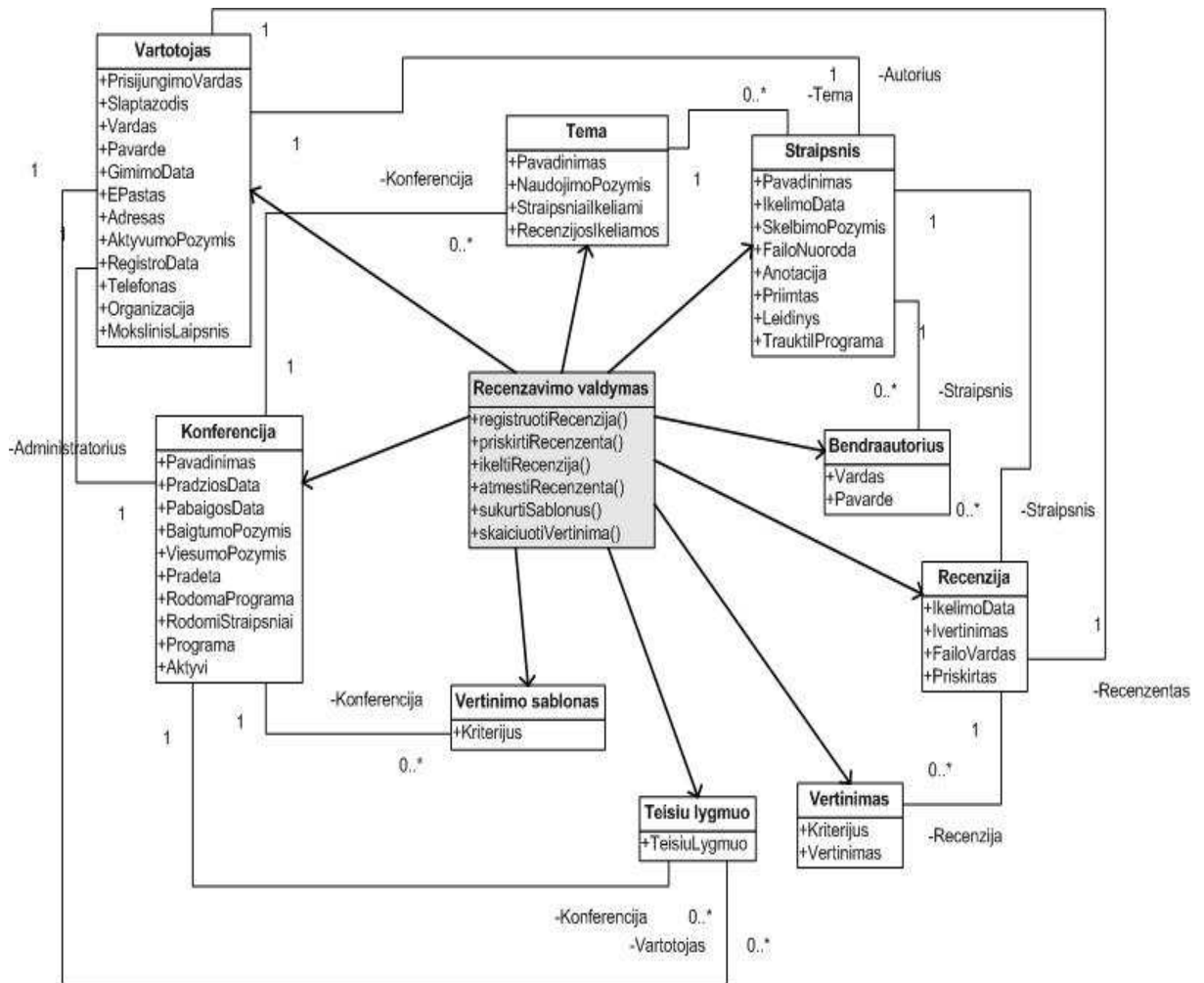
11 pav. Straipsnių valdymo veiklos klasių modelis

Jau minėtas recenzavimo valdymo etapo modelis pateiktas 12 paveiksle. Šio modelio veiklą atlieka taip pat du atlikėjai – recenzentas ir konferencijos administratorius.

Veikla dar pradeda konferencijos organizavimo pradžioje, kai administratorius nustato recenzijų vertinimo šablonus. Sukurtais šablonais remiantis, recenzentas turės vertinti recenzuojamus straipsnius.

Recenzantai gali registruoti paraiškas, konkreitiems darbams recenzuoti. Visos paraiškas įvertinęs administratorius priima jas, tai yra priskirdamas recenzentą straipsniui arba jį atmesdamas.

Pasibaigus straipsnių kėlimo į sistemą etapui, recenzantai recenzuoja straipsnius bei kelia recenzijas į sistemą. Taip pat pateikia straipsnio vertinimą. Suskaičiuotus vertinimus, peržiūri administratorius bei nusprendžia priimti ar atmesti straipsnį. Šis etapas yra pateiktas prieš tai aprašytame straipsnių valdymo modelyje.



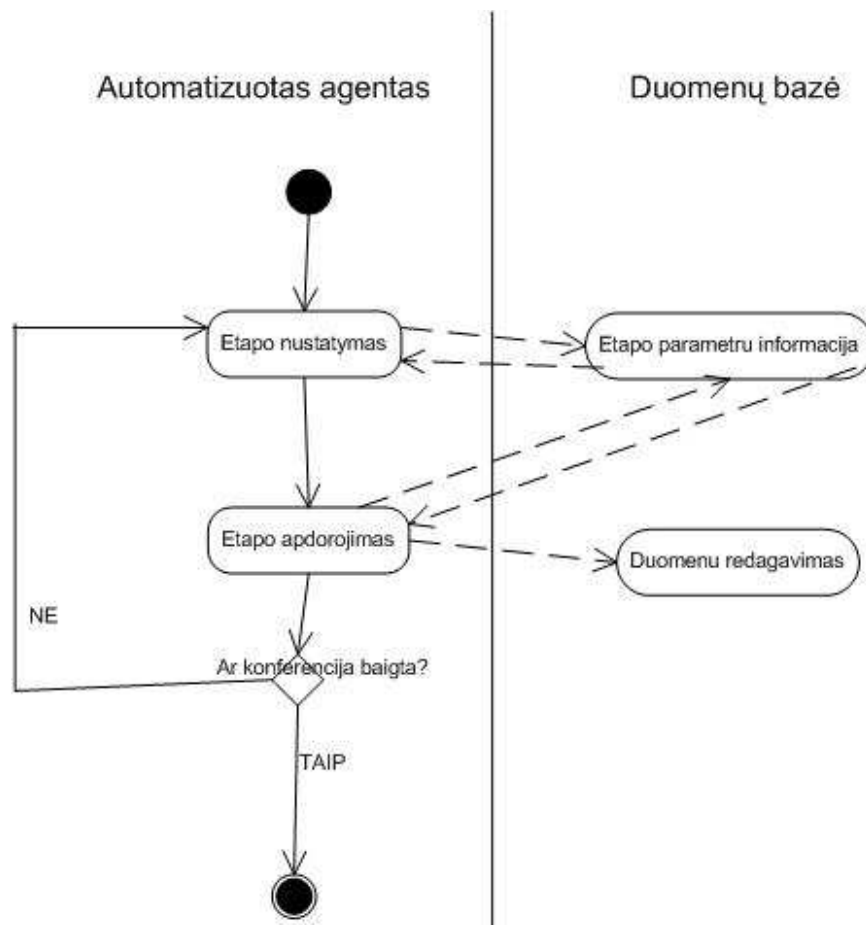
12 pav. Recenzavimo valdymo veiklos klasių modelis

Pateikti veiklos modeliai yra skirtingi, tačiau tuo pačiu ir glaudžiai susiję. Automatizuotas modelis suskaidytas į du loginius agentus, kurie glaudžiai susiję. Taip pat vertinant pranešimų valdymą, matome, jog jis susijęs su automatizuoto agento modeliu. Kiti tarpusavyje susiję modeliai – tai straipsnių valdymo ir recenzijų valdymo modeliai.

4.4. Automatizuoto etapų valdymo agento modelis

Daug buvo rašoma apie automatizuotą valdymą. Taigi, šiame skyrelyje pateiksime automatizuoto agento veiklos modelį. Šis modelis aprašo etapų valdymą laike.

Automatizuotas agentas – tai nuolat veikianti programa, kuri tikrina duomenų bazėje esančius parametrus, juos analizuoja ir apdoroja. Etapų valdymo modelis pateiktas 13 paveiksle.



13 pav. Etapų valdymo modelis

Trumpai buvo aprašytas etapų valdymo modelis 4.3 skyrelyje. Dabar detalizuotai paaiškinsime, kaip šis agentas valdo etapus.

Dirbdamas amžiname cikle, agentas tikrina informaciją apie parametrizuotus etapus, informaciją ją imdamas iš duomenų bazės. Apdorota informacija yra įrašoma į duomenų bazę – tai gali būti etapo valdymo informacija arba sukurtas pranešimas vartotojams apie etapų eigą.

5. AUTOMATIZUOTŲ VEIKLOS PROCESŲ REALIZACIJA TAPESTRY KARKASU

5.1. Loginė sistemos architektūra

Projektuojant sistemą, pagrindinis dėmesys buvo skiriamas veiklos klasių projektavimui. Nereikėjo rūpintis komponentų integravimu į HTML kodą, tai atlieka Tapestry karkasas. Didžiąją dalį vartotojo įvesties tikrinimo irgi aptarnauja Tapestry karkaso varikliukas. Ryšį tarp duomenų bazės ir programos kodo užtikrina Torque generuotos klasės.

Visą sistemą suskirstėme į keturias stambias dalis:

- JSP (*Java Server Page*) puslapiai
- Tapestry karkasas
- Veiklos klasės
- Torque generuotos klasės

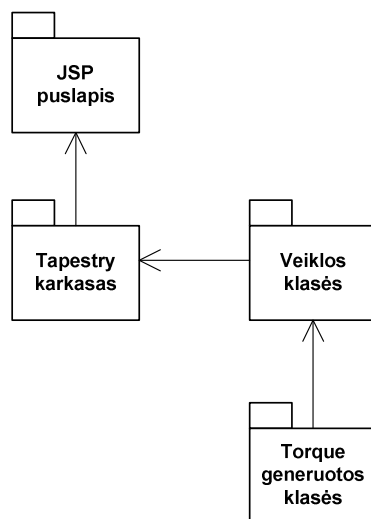
14 paveiksle pavaizduotos šios dalys bei jų loginiai ryšiai.

JSP puslapis naudojamas sąsajai su vartotoju. Jį galima pavadinti sąsaja tarp programos kodo ir vartotojo. JSP puslapio talpykla pasirinktas Tomcat serveris.

Tapestry karkaso dalis rūpinasi veiklos klasių susiejimu su HTML šablonais. Panaudojus šiuos du komponentus, yra sugeneruojamas HTML puslapis, kuris pateikiamas vartotojui JSP pavidale.

Veiklos klasėse aprašytas sistemos funkcionalumas. Žemiau pateiksime detalesnę veiklos klasių schemą. Ši dalis yra svarbiausia sistemos struktūroje.

Torque sugeneruotos klasės naudojamos kaip sąsaja su duomenų baze. Torque technologija leidžia iš sistemos kodo kreiptis į duomenų bazę tiesiogiai nenaudojant SQL užklausų.



14 pav. Konferencijų sistemos paketų struktūra

5.2. Sistemos klasių modelis

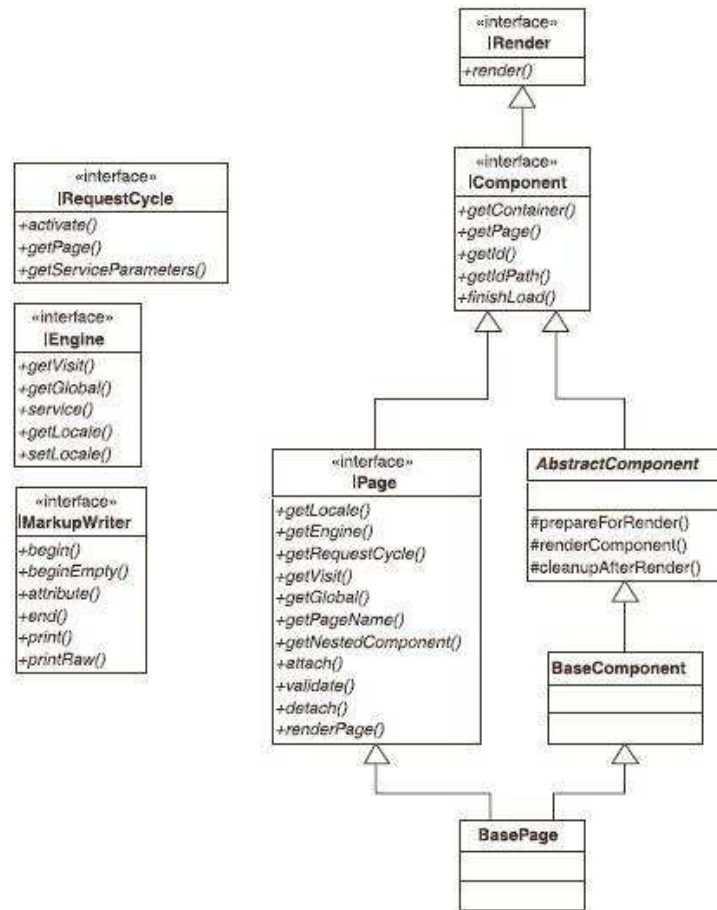
Kaip jau minėjome veiklos (ar verslo) klasės yra svarbiausia architektūros dalis. Tai yra visos sistemos veiklos centras. Visos vartotojo užklauskos yra apdorojamos bei pateikiamas rezultatas, naudojant veiklos klases. Į veiklos klasių struktūrą įtraukiame ir Torque generuotas klases, kad galėtume pateikti išsamesnę visos veiklos struktūrą.

Veiklos klases padaliname į keturias logines dalis:

- Tapestry karkaso realizacijos paketai
- Torque generuoti paketai
- Papildomi paketai
- Automatizuotas agentas

Pagrindinėje dalyje – Tapestry karkaso realizacijos pakete – aprašomi komponentai, jų tarpusavio sąveikos, puslapių apdorojimo mechanizmai. Taipogi aprašomos struktūros, kurios bus naudojamos vartotojo informacijai saugoti. Šio paketo informacija yra įdiegiama į Tapestry karkasą. Diegimo instrukcijos aprašomos nustatymo rinkmenose XML formatu.

Tapestry karkaso klasės pateikiamos 15 paveiksle[13].



15 pav. Pagrindinės Tapestry karkaso sąsajos, klasės, metodai

Torque generuotų paketų nedetalizuosime, kadangi šios technologijos principai buvo aprašyti anksčiau pateiktuose skyriuose.

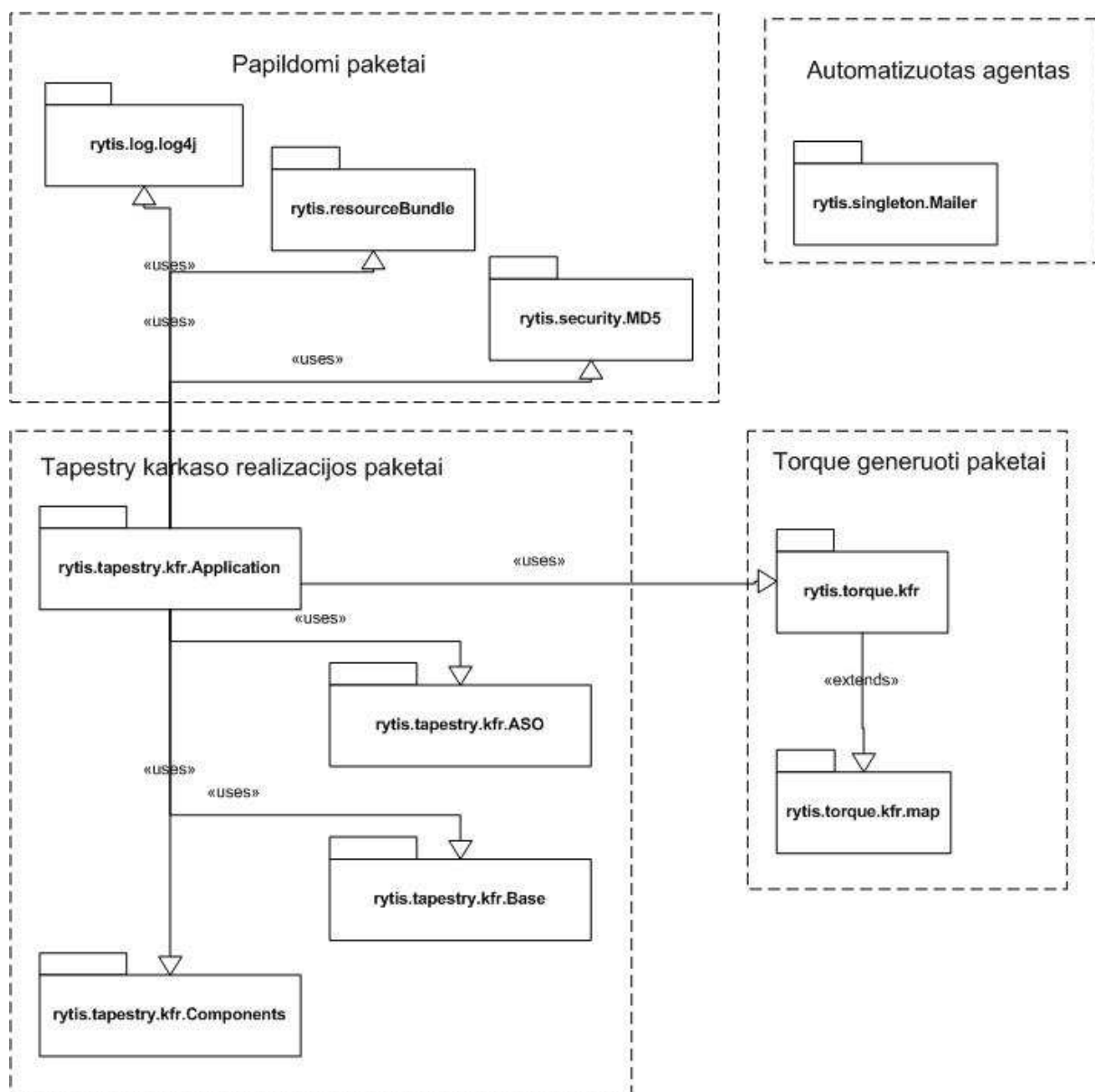
Automatizuoto agento veikla yra atskirta nuo pagrindinių dalių, kadangi jis dirba nepriklausomai nuo vartotojų užklausų. Visus reikalingus duomenis jis randa duomenų bazėje.

Papildomuose paketuose pateikiami papildomi įrankiai, reikalingi normaliam sistemos darbui. Čia pateikiami klaidų išvedimo į rinkmenas įrankiai, daugiakalbystės palaikymo įrankiai bei slaptažodžių šifravimo įrankis.

Aprašytų veiklos klasių struktūrą bei tarpusavio sąsajas pateikiame 16 paveiksle. Paveiksluose paketai yra klasių grupavimo vienetai. Pateiksime klasių paketuose funkcionalumo paaiškinimus:

- rytis.singleton.Mailer – klasių, aprašančių automatizuoto agento veikimą, rinkinys
- rytis.log.log4j – istorijos apie klaidas išsaugojimo klasių rinkinys
- rytis.security.MD5 – MD5 šifravimo algoritmo realizacija

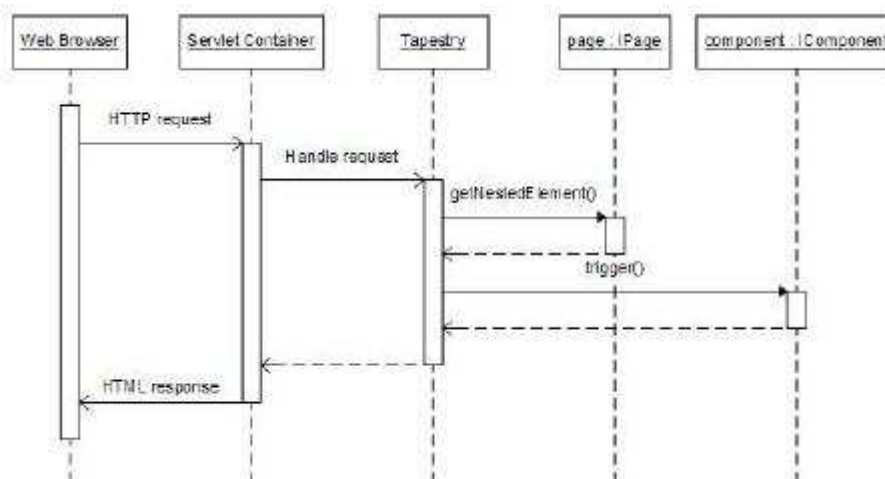
- rytis.torque.kfr – duomenų bazės valdymo klasių rinkinys
- rytis.torque.kfr.map – „žemo lygio“ duomenų bazių valdymo klasių rinkinys
- rytis.tapestry.kfr.Base – smulkiusių struktūros vienetų aprašymo rinkinys
- rytis.tapestry.kfr.ASO – vartotojo duomenų išsaugojimui, naršant tinklapyje, klasių rinkinys
- rytis.tapestry.kfr.Components – komponentų klasių rinkinys
- rytis.tapestry.kfr.Application – puslapių klasių rinkinys



16 pav. Konferencijų sistemos veiklos klasės

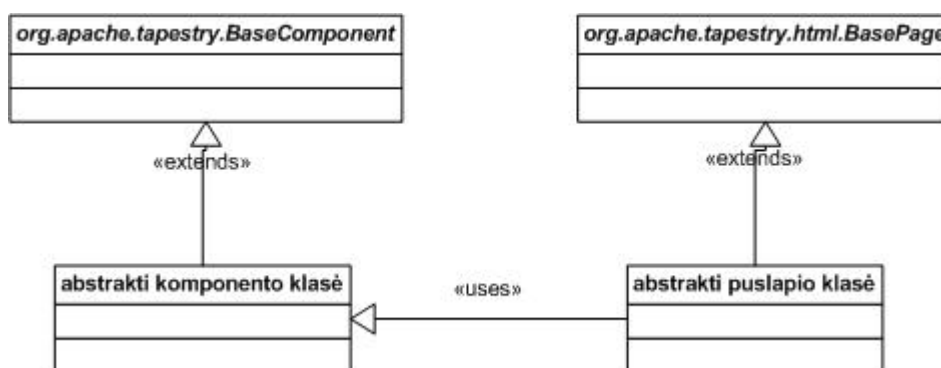
5.4. Sistemos architektūros elementų sąveika

Visos sistemos valdymą atlieka Tapestry valdymo klasės. Pateikiamas supaprastintas Tapestry užklausų vykdymo modelis.



17 pav. Tapestry užklausų valdymo modelis

Tapestry karkasui pateikiamos abstrakčios klasės, kurios yra apdorojamos pagal 17 paveiksle pateiktą schemą[13]. 18 paveiksle yra pateikiamos principinės puslapių ir komponentų klasių realizacijos. Šios realizacijos aprašomos XML valdymo faile, o Tapestry valdiklis jas panaudoja galutiniam puslapiui sugeneruoti.

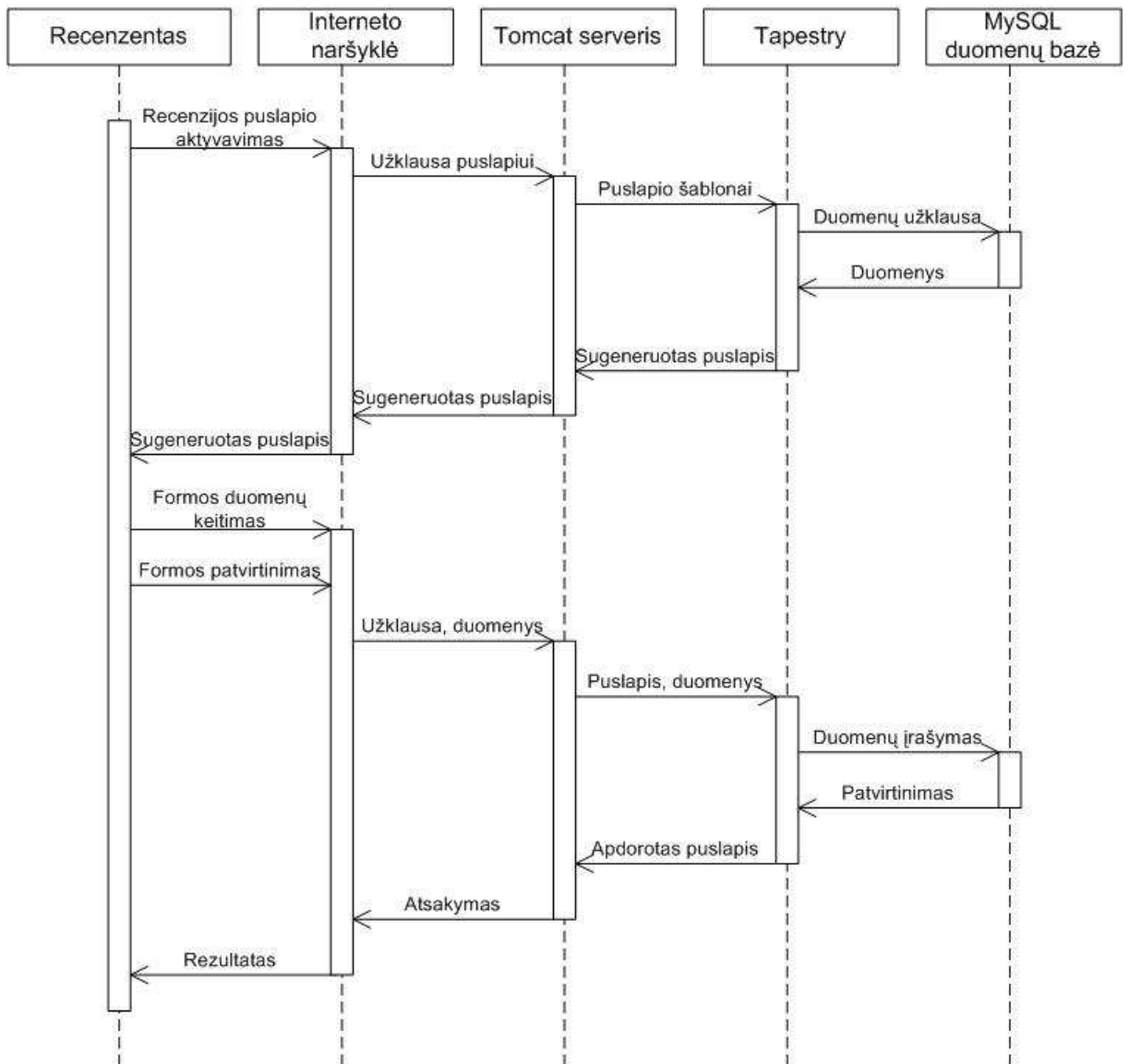


18 pav. Valdymo klasių modelis

Realizuojant abstrakčias klases, kuriamas sistemos veikimas. Visas komponentų bendravimas tarpusavyje paslepiamas nuo programų kūrėjų.

Dauguma vykstančių procesų yra panašūs vienas į kitą. Pateikiamas konkretaus atvejo – recenzijos įkėlimo proceso – sekų diagrama. Šią sekų diagramą galima laikyti kaip modelio

sekų diagramą, kadangi visi procesai prasideda užklausa puslapiui ir duomenų išsaugojimu arba ne. Recenzijos įkėlimo sekų diagrama pateikta 19 paveiksle.



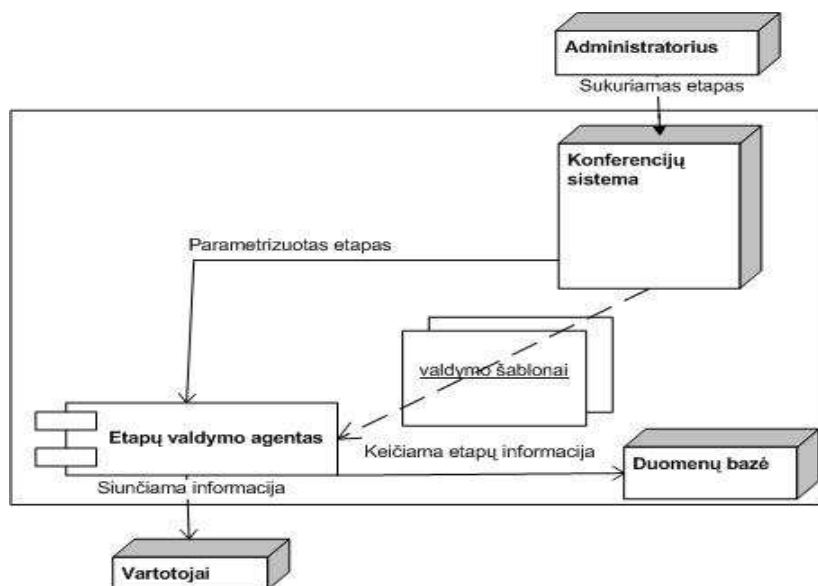
19 pav. Recenzijos įvedimo sekų diagrama

5.5. Automatizuotų veiklos procesų šablonai

Kuriant automatinę sistemą, reikia nustatyti, kokius procesus norime automatizuoti. Pirmiausia tai turėtų būti etapų valdymas laike (etapus pateikėme šio straipsnio 3 skyriuje). Be to, reikia automatizuoti informacijos siuntimą vartotojams, atsižvelgiant į vykstantį etapą, administratoriui paliekant tik nustatymo darbus. Galiausiai, automatizuojama straipsnių recenzijų apibendrinimo funkcija, kuri pateikia suminius straipsnių vertinimų rezultatus. Toliau pateikiami šių procesų automatizavimo modeliai.

Kiekvieną iš etapų galima laikyti automatizuoto modelio kintamuoju, kur modelis vaizduoja savarankišką sistemą – informacijos apdorojimo ir valdymo agentą. 3 skyriuje pateikėme etapo suskaidymą į parametrus, dabar pateiksime modelį tiems etapams valdyti.

20 paveiksle pateiktame modelyje vyksta tokie procesai: siunčiamas valdymo signalas naujo etapo sukūrimui ar jo redagavimui. Tuomet informacija apie etapą bei valdymo šablonai perduodami į etapų valdymo agentą. Valdymo šablonais nurodomi sąlyginiai laiko vienetai bei informacija, kuri turi būti siunčiama vartotojams nurodyto etapo metu. Agentas renka informaciją apie kiekvieną iš etapų bei pagal ją atlieka tam tikrus veiksmus – keičia informaciją duomenų bazėje ar siunčia ją vartotojams.



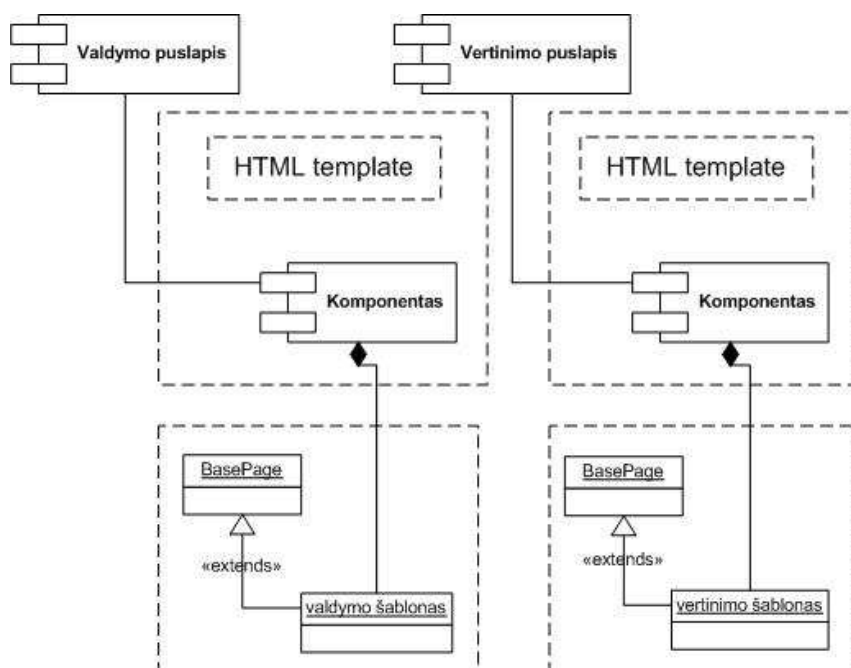
20 pav. Etapų valdymo modelis

Automatizuotoje vertinimų sistemoje panašiai kaip ir etapų valdymo modelyje yra vertinimo šablonai. Šie šablonai sukuriami konferencijos pradžioje ir naudojami recenzavimo

etape vertinant straipsnius. Surinkta informacija vėliau naudojama skaičiuojant straipsnių recenzijų vertinimus ir pateikiamos suminės reikšmės administratoriui.

5.6. Automatizuotų komponentų integravimas į Tapestry karkasu sukurtą sistemą

Automatizuoto modelio šablonai realizuojami Tapestry komponentais ir integruojami į visą sistemą. Kaip jau minėta darbo pradžioje, vienas iš Tapestry karkaso privalumų yra komponentų naudojimas. Komponentus nesunku integruoti į sistemą ar pakeisti vieną komponentą kitu. Integravimo modelis pateiktas 21 paveiksle.



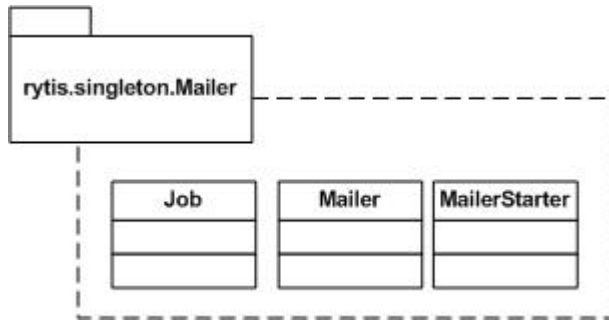
21 pav. Automatizuotų procesų integravimo modelis

Sukurtos šablonų klasės veldi informaciją iš standartinių Tapestry klasių, kad vėliau jos galėtų būti įtraukiamos į Tapestry karkasą. Šabloninės klasės realizuoja etapų valdymo ar recenzijų vertinimo funkcionalumą. Sukurti komponentai yra įtraukiami į kitus komponentus, kuriuose išnaudojami sukurti komponentai.

Tapestry varikliukas ima du komponentus – HTML šabloną ir mūsų sukurtą komponentą. Jie sugeneruojami pagal nustatytas taisykles ir gaunamas vartotojui pateikiamas puslapis. Tai etapų valdymo ar straipsnių vertinimo puslapiai.

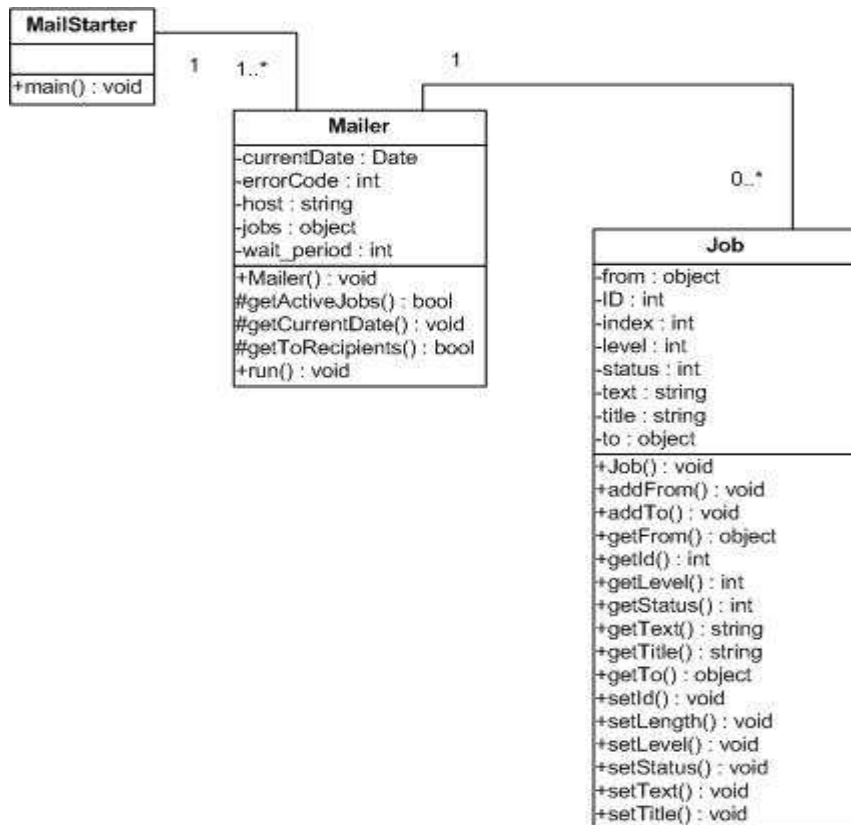
5.7. Agento klasių modelis

Aukščiau aprašyto, automatizuoto agento modelius atvaizduosime agento realizacijos modeliu. 22 ir 23 paveiksluose pateikiamas automatizuoto agento klasių modelis. Pateikiama bendras paketo vaizdas bei detalizacija iki klasių lygio, pateikiant ryšius tarp klasių.



22 pav. Automatizuoto agento klasių paketas

Detalizuotas paketo vaizdas pateiktas žemiau, 23 paveikslėlyje. Pateiktos klasės su vidiniais kintamaisiais bei klasėmis.

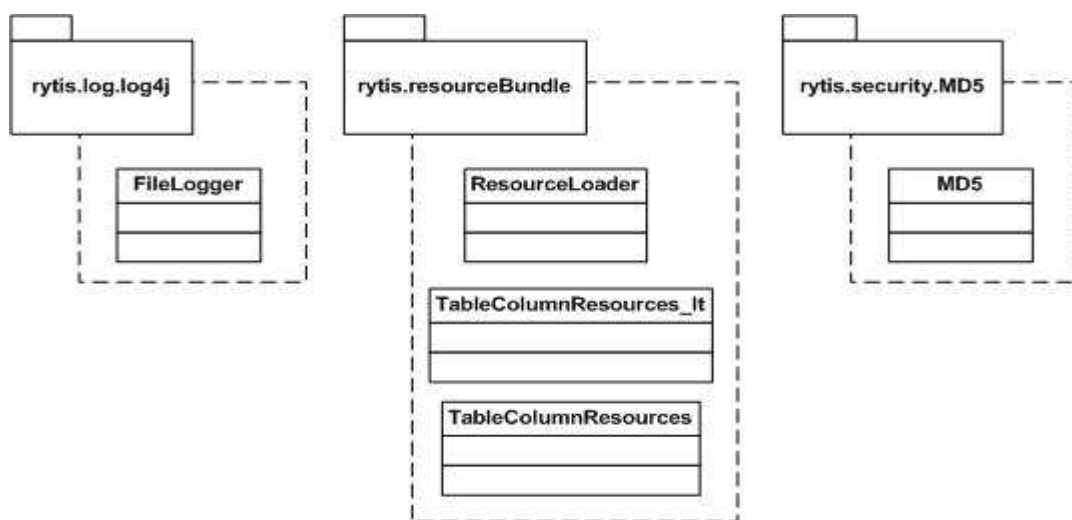


23 pav. Automatizuoto agento klasių detalizacija

MailStarter klasė skirta agento paleidimui. Gali būti paleidžiami keli lygiagrečiai dirbantys agentai. Automatizuoto agento veiklos realizacija pateikta Mailer klasėje. Ši klasė naudoja Job klasę darbams aprašyti bei juos apdoroti.

5.8. Sistemos realizacijos klasių modelis

Sistemos saugumo funkcijas, daugiakalbystės realizaciją bei sistemos veiklos istorijos sekimą realizuojančias klases sugrupuotas, pateikiame kaip papildomus sistemos paketus. Šie paketai nėra patys svarbiausi sistemos funkcionalume, bet pakankamai svarbūs. Jie pateikiami 24 paveiksle.

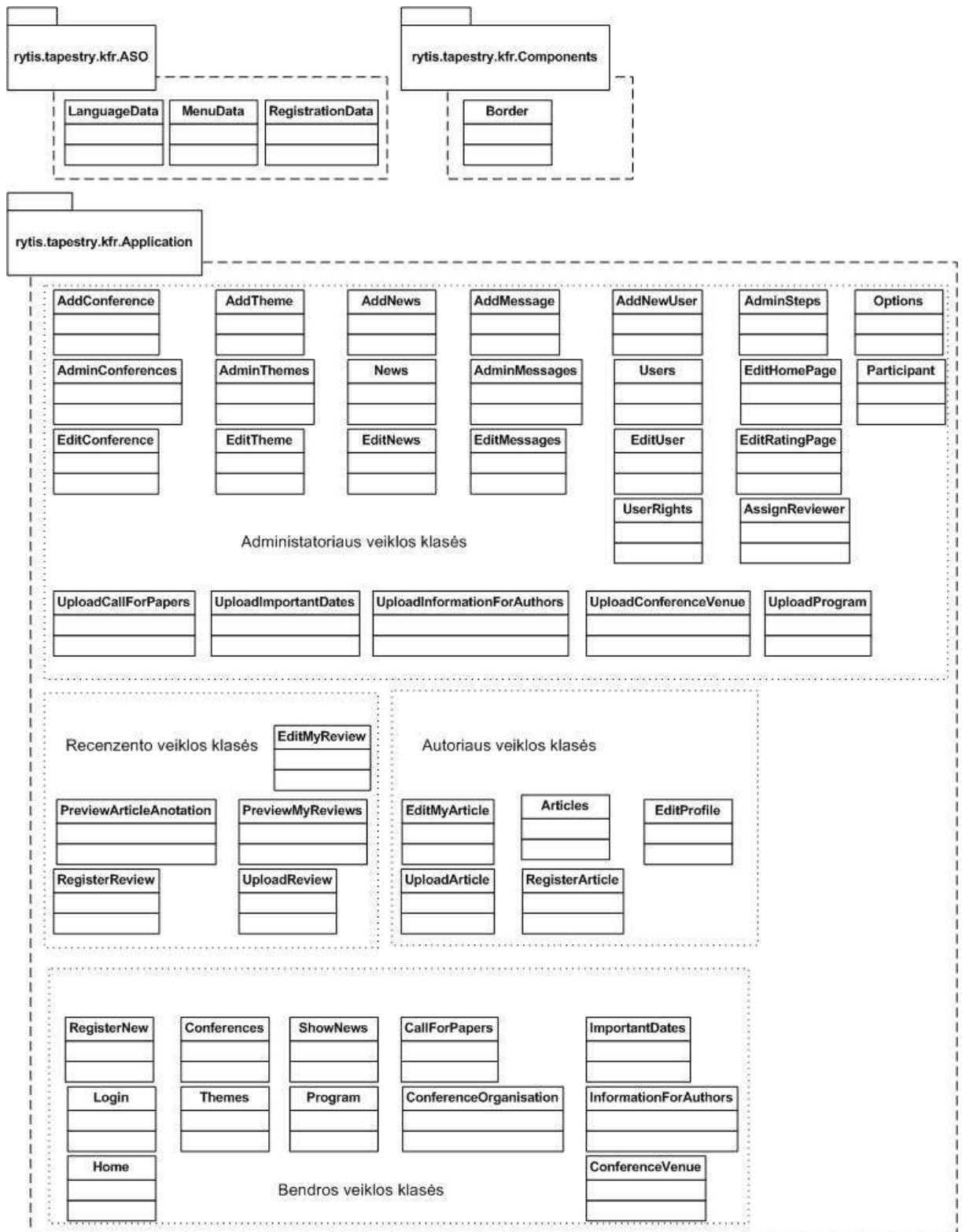


24 pav. Papildomi sistemos paketai

25 paveiksle pateikti Tapestry karkaso realizacijai reikalingų klasių rinkiniai. Dauguma šių klasių yra abstrakčios ir naudojamos, generuojant vartotojui pateikiamus puslapius ir apdorojant užklausas.

Application paketo klasės tarpusavyje tiesiogiai nesusiję, bet naudoja ASO, Components, Base paketų klases funkcionalumui realizuoti.

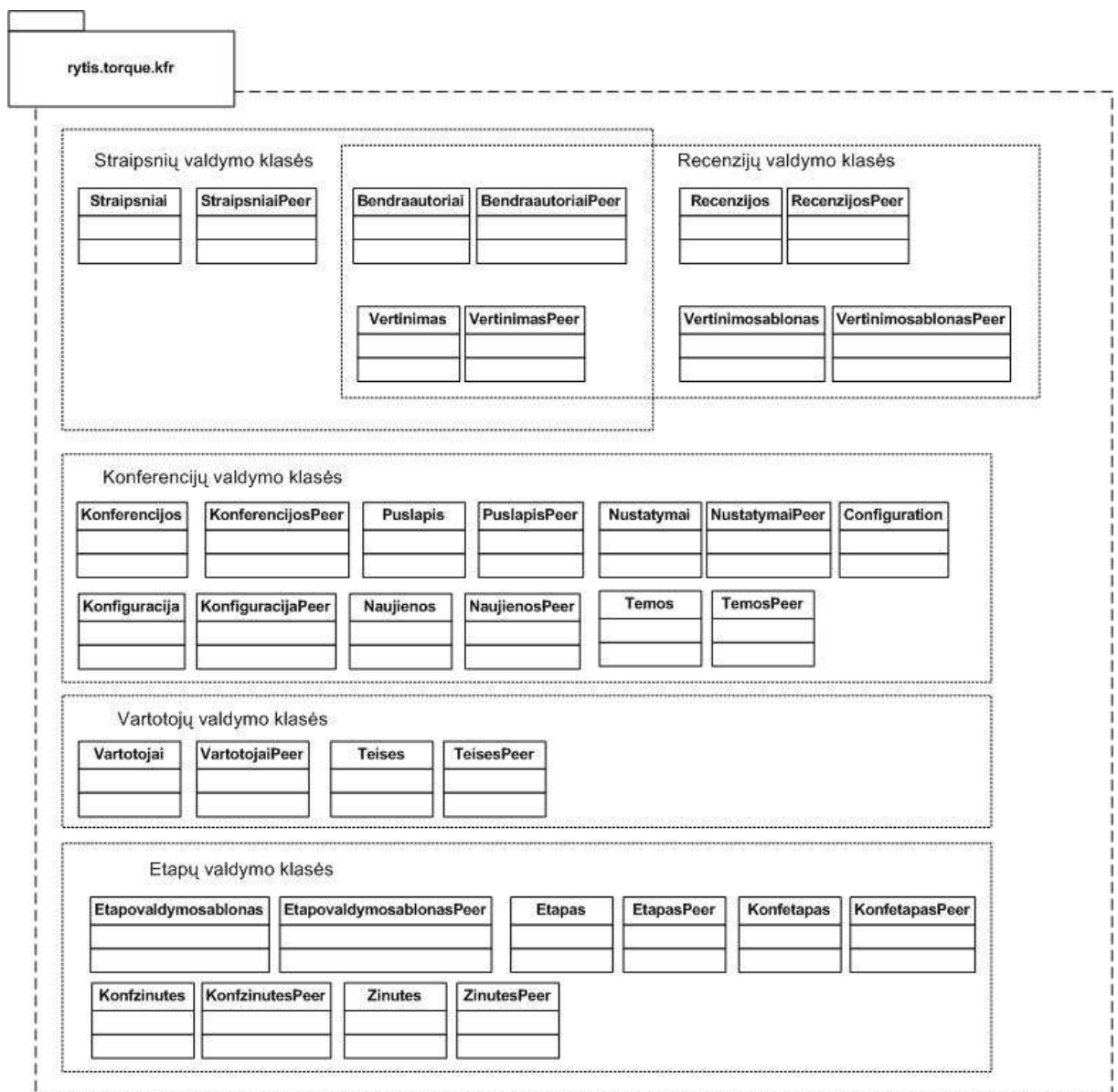
Klasių nepriklausomumas tarpusavyje suteikia sistemai plečiamumą. Šią savybę įvertinsime darbo eksperimentinėje dalyje.



25 pav. Tapestry karkaso realizacijos paketai

Toliau 26 paveiksle pateikiamos Torque generuotos klasės paketai. Map paketo klasės yra nekeičiamos, jos yra kaip tarpininkas, tarp kfr paketo klasių ir duomenų bazės. Jų realizacija yra pakankamai „žemo lygio“, tai yra realizuotos SQL užklausos, naudojant standartines Torque klases.

Kfr pakete pateiktos klasės skirtos naudojimui ir galimybei perrašyti, kad realizuoti specifinį ar tobulesnį klasių naudojimą, kreiptis į duomenų bazę.



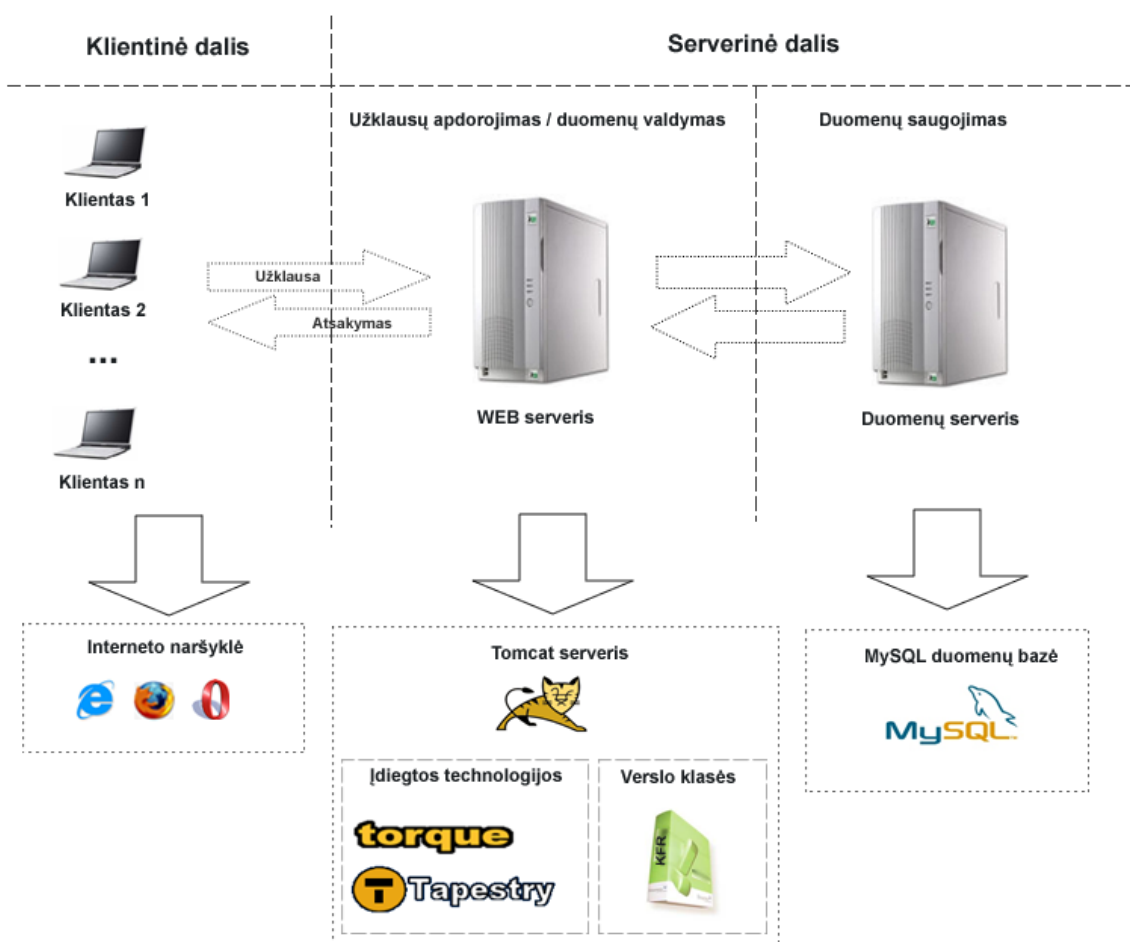
26 pav. Torque generuotų klasių paketai

26 paveiksle pateiktas klasės sugeneruotos Torque generatoriaus įrankiu, panaudojant duomenų bazės aprašą, kuris aprašytas 5.9 skyrelyje.

5.10. Diegimo architektūra

Konferencijų sistemos programinė įranga realizuota kliento-serverio principu. Klientinė dalis – tai vartotojai, naudojantys interneto naršykles kompiuteriuose su prieiga prie interneto. Tai standartinė programinė įranga, skirta interneto puslapių peržiūrai. Interneto naršyklė gali būti Internet Explorer, Firefox, Opera ar kitos.

Kuriama programinė įranga diegiama į serverį. Ši galima skaidyti į užklausų apdorojimo, duomenų valdymo ir duomenų saugojimo lygius. Duomenų apdorojimui pasirinktas Tomcat serveris, na o duomenų saugojimui MySQL. Detalesnę sistemos architektūrą galima peržiūrėti 28 paveiksle.



28 pav. Sistemos architektūra

6. KONFERENCIJŲ SISTEMOS KOKYBĖS IR EFEKTYVUMO ANALIZĖ

6.1. Sistemos veikimo analizė

Konferencijų sistema skirta konferencijoms organizuoti. Šiame skyrelyje pateiksime pagrindinių organizavimo etapų realizacijos langus. Pateikiami langai iš šiuo metu organizuojamos pirmos konferencijos anglų kalba.

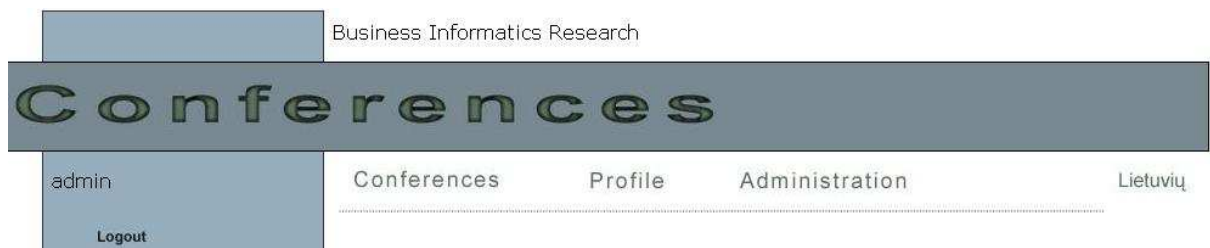
Prisijungus prie sistemos yra pateikiama visa reikalinga informacija, norintiems dalyvauti vykstančioje konferencijoje ar peržiūrėti kokios jau įvyko.



29 pav. Pradinis sistemos langas

Iš pagrindinio meniu pasiekiami informacija apie konferencijos temas, svarbias datas, konferencijos komitetą, vietą, kvietimą į konferenciją, informaciją autoriams. Taip pat yra galimybė registruotis į sistemą arba jungtis, jei anksčiau buvo prisiregistruota.

Priklausomai nuo prisijungusio vartotojo lygmens, yra atidengiami specialūs meniu punktai.



30 pav. Prisijungusio dalyvio valdymo meniu

Pateiktame 30 paveikslėlyje pateikiamas administratoriui atidengtas valdymo meniu.

Prisijungus autoriui ar recenzentui, atidaromi specialūs punktai „My articles“ ir „My reviews“, kuriuose jis ras visus savo rašytus straipsnius ir jų recenzijas, o recenzentas - savo rašytas recenzijas bei recenzuojamus straipsnius. Pateikiama informacija ne tik apie konferenciją, kurioje dalyvauja šiuo metu, bet ir apie seniau įvykusias konferencijas, kuriose jis dalyvavo. 31 paveiksle pateiktas autoriaus, kuris yra ir recenzentas, darbo langas.



31 pav. Autoriaus ir recenzento darbo langas

Autoriaus straipsnių sąrašas pateiktas 32 paveiksle.

Business Informatics Research

Conferences

rytisbud

Logout

Call for papers

Conference topics

Important dates

Conference organisation

Conference pictures

Conference program

Conference venue

Information for authors

Conference history

⇒ My articles

Conferences Profile Lietuvių

Conference	Theme	Article	Date	Review
Business Informatics Research	Change Management	RTS_TEST	2006-05-01	Preview
Business Informatics Research	Business and information systems development	RTS_TEST2	2006-05-01	Preview

32 pav. Straipsnių sąrašas

Pasirinkus aktyviają straipsnio nuoroda, atidaromas langas straipsnio koregavimui bei įkėlimui. Vaizdas pateiktas 33 paveiksle.

Business Informatics Research

Conferences

rytisbud

Logout

Call for papers

Conference topics

Important dates

Conference organisation

Conference pictures

Conference program

Conference venue

Information for authors

Conference history

⇒ My articles

Conferences Profile Lietuvių

←

Article: RTS_TEST
Delete

Upload article: Browse...
Upload

Article title: RTS_TEST

Annotation:

Update

33 pav. Straipsnio įkėlimo į sistemą langas

Recenzento sukurtų recenzijų sąrašas bei įkėlimo langai panašūs. Vienintelis didesnis skirtumas yra tas, jog recenzentai dar privalo įvertinti straipsnį.

Administratoriaus pagrindinis darbas vyksta pasirinkus „Administration“ meniu punktą. Pasirinkus šį punktą yra pasiekiami konferencijos valdymo, žinučių kūrimo, vartotojų

administravimo, nustatymų valdymo bei etapų valdymo meniu. 34 paveiksle pateiktas konferencijų sąrašas iš valdymo meniu.

Administratorius gali aktyvuoti arba uždaryti konferencijos organizavimą vienu mygtuko paspaudimu – tai atliekama pasirinkus aktyvias nuorodas „Started“ ar „Closed“. Pasirinkus konferencijos aktyvavimą, automatiškai įjungiamas visoms temoms straipsnių registravimas ir įkėlimas.



The screenshot shows the 'Conferences' management interface. At the top, it says 'Business Informatics Research'. Below that is a large 'Conferences' header. On the left is a sidebar with 'admin' and 'Logout' options. The main content area has tabs for 'Conferences', 'Profile', and 'Administration' (which is selected). There is a '+ Add' button and a table of conferences.

Title	Participants	Themes	Start date	End date	Started	Closed	Public
Business Informatics Research	2006-05-30	2006-10-27	Taip	Ne	Taip

34 pav. Konferencijų valdymo langas

Pasirinkus redaguoti konferenciją yra atidaromas atskiras langas, jis pateiktas 35 paveiksle. Administratoriui suteikiama galimybė taisyti konferencijos pavadinimą bei pradžios ir pabaigos datas. Taipogi administratorius nustato ar konferencija bus prieinama tik registruotiems sistemos vartotojams, ar bus rodomas pirmas sistemos puslapis. Yra galimybė nustatyti ar bus rodoma programa ir straipsniai, pasibaigus konferencijai.



The screenshot shows the 'Conferences' editing interface. At the top, it says 'Business Informatics Research'. Below that is a large 'Conferences' header. On the left is a sidebar with 'admin' and 'Logout' options. The main content area has tabs for 'Conferences', 'Profile', and 'Administration' (which is selected). There is a '< Atnaujinti' button and a form for editing conference details.

Administrator:

Title:

Start:

End:

Public

Show first page

Show program

Show articles

[Edit home page](#)

[Edit rating templates](#)

[Edit steps](#)

35 pav. Konferencijos redagavimo langas

Iš aukščiau pateikto darbo lango galima patekti į pirmo puslapio redagavimo, šablonų redagavimo ir etapų valdymo langą.

Pirmo puslapio redagavimo lange nurodomas pavadinimas, paveikslas ir kita aktuali informacija. Kuri vėliau bus suformatuota bei išvesta vartotojui.

Etapų redagavimo lange nurodomi kokie etapai bus vykdomi bei kokios jų pradžios ir pabaigos datos. Taip pat nurodomi dažnumo intervalai – informacijos siuntimui.

Konferencijos redagavimo lange galima keisti visą su pačia konferencija tiesiogiai susijusią informaciją. Taipogi iš šio lango yra pasiekiamas konferencijos straipsnių recenzavimo vertinimų šablonų kūrimas. Šis langas pateiktas 36 paveiksle.



The screenshot shows a web interface for 'Business Informatics Research' with a 'Conferences' section. The user is logged in as 'admin'. The main navigation includes 'Conferences', 'Profile', and 'Administration' (which is active). A sidebar on the left contains links for 'Logout', 'Conferences', 'Messages', 'Users', 'Options', and 'Steps'. The main content area shows a 'Criterion' form with a table of criteria:

Criterion	
1. Originality	Remove
2. Presentation	Remove
3. Relevance to conferer	Remove
4. Significance of topic	Remove
5. Technical quality	Remove
6. Overall	Remove

Below the table is a 'Submit' button. At the top of the main content area, there are links for 'Conferences', 'Profile', and 'Administration', along with a 'Lietuvių' language selector and a '← Atnaujinti +' refresh button.

36 pav. Recenzijų vertinimo šablonų kūrimo langas

Čia sukurtus kriterijus, recenzantai vertins, įkeldami recenzijas į sistemą.

Iš 36 paveiksle pateikto darbo lango galima patekti į etapų rankinio valdymo langą, kuriame galima nustatyti atskirai kiekvienai temai šiuo metu galimus veiksmus. Ar bus įkeliami/redaguojami straipsniai, ar bus įkeliamos recenzijos. Darbo langas pateiktas 37 paveiksle.

Business Informatics Research

Conferences

admin Conferences Profile **Administration** Lietuvių

← +

Conference title: Business Informatics Research
Conference administrator: Admin Admin

Title	Uploading articles	Uploading reviews	Used
Business and information systems development	Taip	Ne	Taip
Business intelligence	Taip	Ne	Taip
Business, people and system interoperability	Taip	Ne	Taip
Change Management	Taip	Ne	Taip
Knowledge Management and Semantic Web	Taip	Ne	Taip
Open source and open information society	Taip	Ne	Taip

37 pav. Rankinio etapų valdymo darbo langas

Žemiau pateiktame 38 paveiksle matome konferencijos dalyvių sąrašą. Šiame lange galima blokuoti straipsnius, jiems priskirti recenzentus.

Business Informatics Research

Conferences

admin Conferences Profile **Administration** Lietuvių

←

Conference: Business Informatics Research

Author	Theme	Article	Reviewer	
Lina Nemuraite	Information systems	blogas straipsnis	Priskirtas	Blokuoti
Lina Nemuraite	Information systems	Conference system	Priskirtas	Blokuoti
Rytis Budreika	Information systems	Design of information systems	Priskirtas	Blokuoti
Lina Nemuraite	Information systems	fff	Priskirtas	Blokuoti
Rimantas Butleris	Information systems	Geras straipsnis	Priskirtas	Blokuoti

38 pav. Konferencijos dalyvių sąrašas

Konferencijos organizavimo etapus valdo automatizuotas agentas, tačiau palikta galimybė administratoriui rankiniu būdu valdyti etapus. 39 paveiksle pateiktas etapų rankinio valdymo langas.

Conferences

admin

Conferences

Profile

Administration

Lietuvių

Logout

⇒ Conferences

Messages

Users

Options

Steps



Conference title: Business Informatics Research

Conference administrator: Admin Admin

Title	Uploading articles	Uploading reviews	Used
Business and information systems development	Taip	Ne	Taip
Business intelligence	Taip	Ne	Taip
Business, people and system interoperability	Taip	Ne	Taip
Change Management	Taip	Ne	Taip
Knowledge Management and Semantic Web	Taip	Ne	Taip
Open source and open information society	Taip	Ne	Taip
zzz1	Ne	Ne	Ne
zzz2	Ne	Ne	Ne

39 pav. Konferencijos etapų valdymo langas

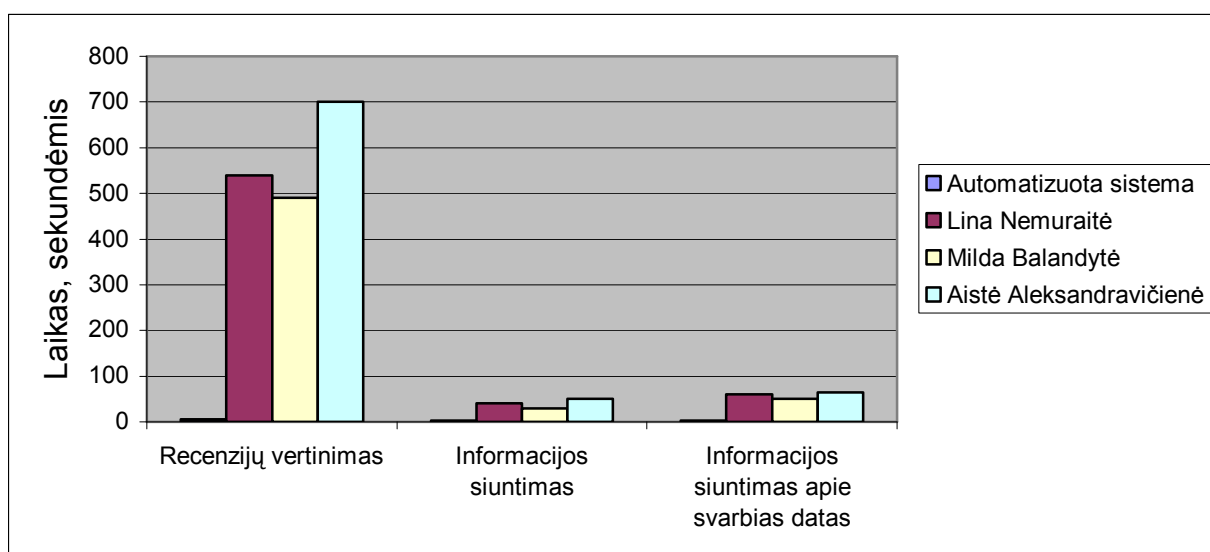
6.2. Automatizuotų procesų efektyvumo tyrimas

Norint suprasti automatizuotos veiklos procesų privalumus, galima palyginti neautomatizuotam procesui atlikti sugaištą laiką su laiku, sugaištam tam procesui vykdyti automatizuotu būdu.

Norint įvertinti programinių agentų efektyvumą, buvo skaičiuojamas vidutinis recenzijų vertinimų skaičiavimo, informacijos siuntimo, žinučių apie svarbias datas siuntimo laikas automatizuotu būdu (naudojant agentus) ir neautomatizuotu (kai tą daro administratorius). Recenzijų vertinimo skaičiavimui automatizuoto ir neautomatizuoto vykdymo laiko santykis buvo ~1:99, informacijos siuntimui – ~1:9, žinutėms apie svarbias datas ~1:19. Pati automatizuota sistema nereikalauja labai daug administratoriaus laiko. Pagrindinį laiką sudaro sistemos konfigūravimas. 3 lentelėje pateikti eksperimento rezultatai. Laikas pateiktas sekundėmis.

4 lentelė

Atlikėjas \ Kriterijus	Recenzijų vertinimas	Informacijos siuntimas	Informacijos siuntimas apie svarbias datas
Automatizuota sistema	5	3	3
Lina Nemuraitė	540	40	60
Milda Balandytė	490	30	50
Aistė Aleksandravičienė	700	50	65



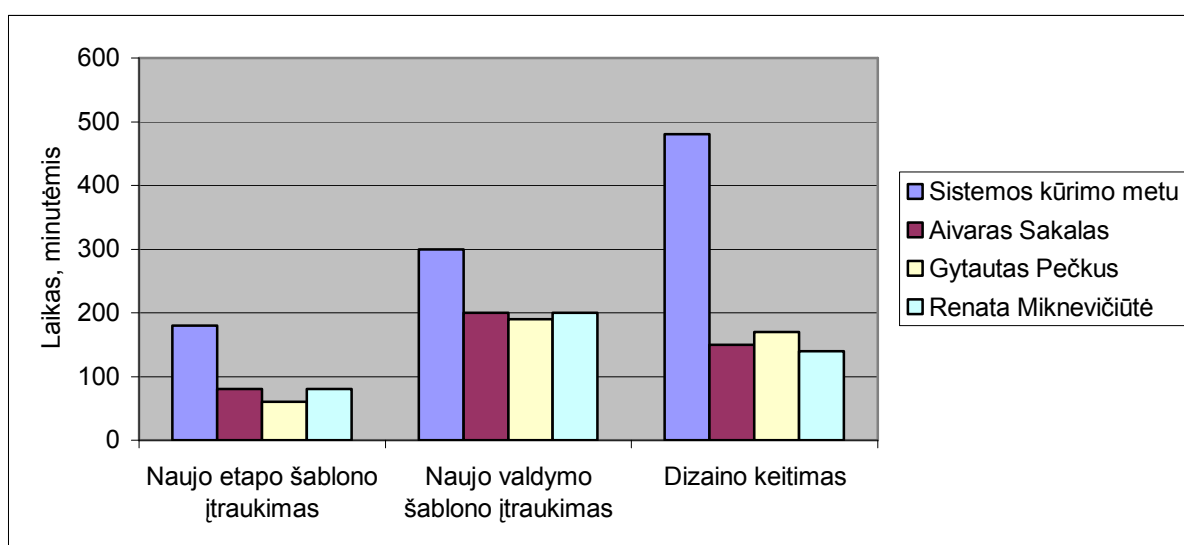
Kaip matyti iš grafiko automatizuotos sistemos sugaištas laikas yra labai mažas lyginant su rankiniu darbu.

6.3. Sistemos plečiamumo tyrimas

Kitas labai svarbus faktorius yra sistemos plečiamumas. Žinodami teorines Tapestry karkaso plėtimo galimybes, pabandėme įvertinti realias laiko sąnaudas reikalingas funkcionalumui ir dizainui plėsti. Tam buvo apklausti keli programuotojai, gerai susipažinę su šiuo karkasu. Jie vertino tokius sistemos pakeitimus: naujo etapo šablono įtraukimas į sistemą, naujo valdymo šablono įtraukimas, dizaino pakeitimas (lyginome su laiku, sugaištu pradiniu sistemos kūrimo metu): naujo etapo šablono įtraukimas: ~2:5; naujo valdymo šablono įtraukimas: ~3:5; dizaino keitimas: ~3:10. Gauti rezultatai patvirtina, kad sistema yra lengvai plečiama. 4 lentelėje pateikti eksperimento rezultatai. Laikas pateiktas minutėmis.

5 lentelė

Atlikėjas \ Kriterijus	Naujo etapo šablono įtraukimas	Naujo valdymo šablono įtraukimas	Dizaino keitimas
Sistemos kūrimo metu	180	300	480
Aivaras Sakalas	80	200	150
Gytautas Pečkus	60	190	170
Renata Miknevičiūtė	80	200	140

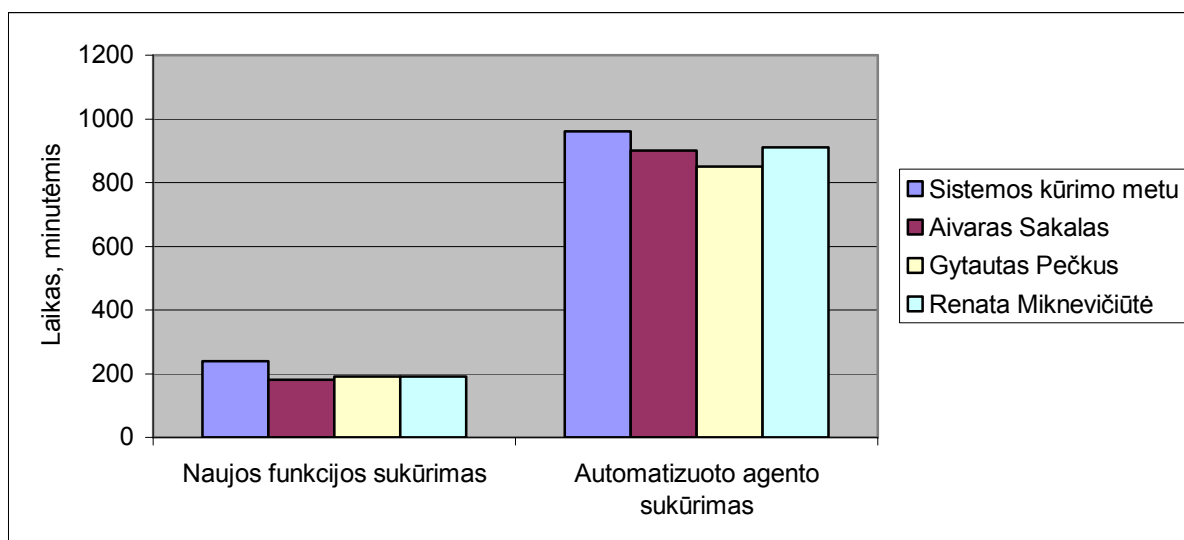


Tiriant sistemos plečiamumą, buvo nagrinėjama ir naujos funkcijos įtraukimo galimybė, naujo agento sukūrimas. Šie funkcionalumo keitimo akcentai nėra labai adekvatūs,

tai yra negali lengvai nustatyti jų atlikimo ir įdiegimo laiko sąnaudas. Kiekvienas naujas funkcionalumas turi savus reikalavimus ir realizavimo niuansus. Tačiau tyrimo tikslais buvo priimta, jog reikalavimai ir realizacijos sudėtingumas prilygsta vidutinėms laiko sąnaudoms, skirtoms jau sukurtų funkcijų realizacijai. Rezultatai pateikti minutėmis 6 lentelėje.

6 lentelė

Atlikėjas \ Kriterijus	Naujos funkcijos sukūrimas	Automatizuoto agento sukūrimas
Sistemos kūrimo metu	240	960
Aivaras Sakalas	180	900
Gytautas Pečkus	190	850
Renata Miknevičiūtė	190	910



Kaip matyti iš grafikų – naujo funkcionalumo kūrimas nėra toks efektyvus kaip tam tikro funkcionalumo plėtimas.

Lengvą komponentų integraciją suteikia Tapestry karkaso naudojimas. Visą komponentų sąveikavimo funkcionalumo realizavimą atlieka karkasas. Užtenka realizuoti atskirus komponentus, juos teisingai aprašyti konfigūracijos XML failuose, o visa kita pasirūpina Tapestry karkasas.

Buvo projektuojami bei kuriami veiklos modeliai, atsižvelgiant į tai jog sistema ateityje bus plečiama ir keičiamas funkcionalumas. Panaudota Torque technologija, leidžia

lengvai pildyti duomenų struktūrą – minimaliai keičiamas programos kodas, jei pasikeičia buvusi struktūra.

Plečiant sistemą, tai yra keičiant funkcionalumą ar dizainą, sugaištas laikas yra mažesnis nei kuriant sistemą. Tačiau kuriant naujus komponentus, laikas nežymiai sutaupomas. Naujiems komponentams kurti retai panaudojami jau sukurti funkcionalumai ir juos reikia sukurti iš naujo.

Plėtimui vystyti buvo specialiai kuriami modeliai tuo tikslu. Tapestry ir Torque technologijų naudojimas žymiai prisidėjo prie lengvo sistemos plečiamumo.

Plečiamumą ir lankstumą riboja topologija. Paanalizuosime kiek reikės keisti ar sukurti elementų, keičiant jau sukurtą elemento vienetą. Naudojamos Tapestry ir Torque technologijos, kuriant sistemą, mažina topologijos sudėtingumą.

Skyrelyje 5.8 paveiksle 25 pateikta Tapestry karkasui skirtas klasių realizacijos modelis. Jame pateiktos atskiros klasės, realizuojančios atskirus internetinius puslapius. Keičiant vieną iš šių klasių, kitos klasės keisti nereikia. Jei keičiami bendri atributai ar pildoma naujais, tai reikia keisti Tapestry realizacijos klases, kurios naudoja tą atributą ir bendrą klasę, kuri naudojama atributo informacijai saugoti.

Taigi apibendrinant galima teigti, jog keičiant vieną Tapestry realizacijos klasę, tai nereikia keisti kitų klasių. O darant pakeitimą, kuris įtraukia bendrą atributą, tai reiktų keisti mažiausiai tris klases.

Darant pakeitimus Torque generuotose klasėse, klasės pateiktos 5.8 skyrelyje 26 paveiksle, nereikia keisti kitų klasių. Klasės yra sugeneruotos taip, kad žemiausias lygmuo yra nekintamas, o kinta tik aukštesnis lygmuo, tai klasės paveldėjusios iš žemiausio lygio klasių.

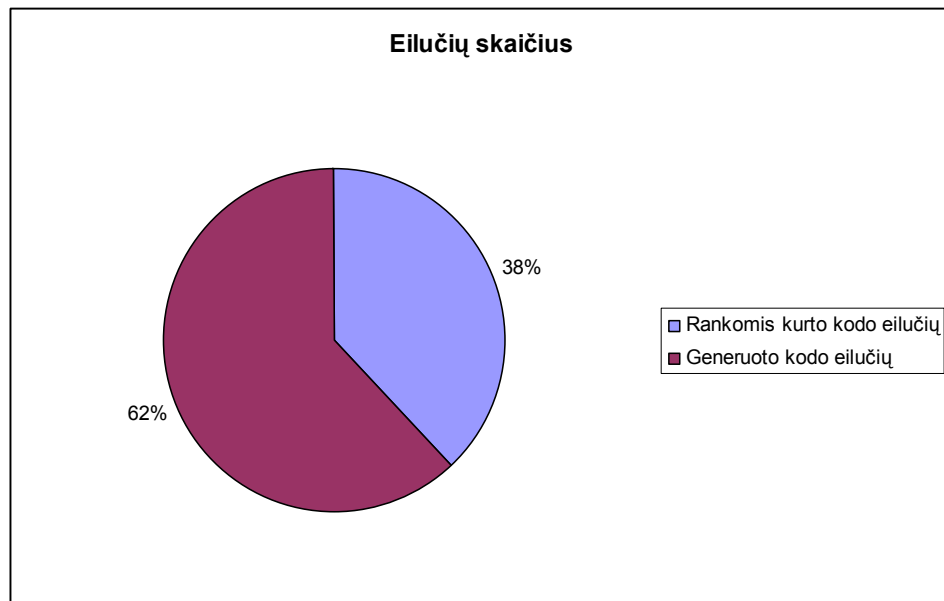
6.4. Tapestry karkaso ir Torque technologijos įtaka sistemos kūrimui

Pagrindinė sistemos dalis realizuota Tapestry karkaso komponentais. Mūsų sukurti komponentai susipina su Tapestry komponentais. Pateiksime keletą metrikų, nurodydami Tapestry įtaką šioms metrikoms.

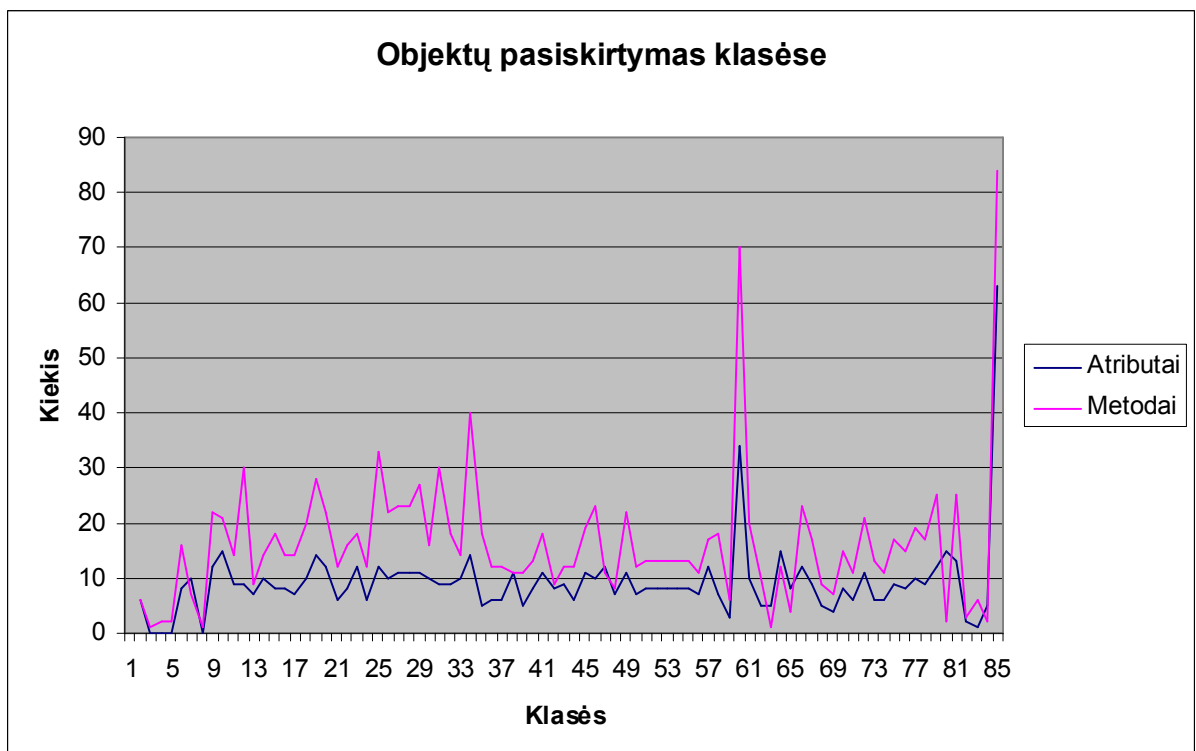
Sistema kurta objektinio programavimo principais. Taigi metrikas analizuosime klasių lygmenyje.

Skaičiuodami kodo eilučių skaičių, vidutinį eilutės ilgį laikėme 80 simbolių. Visos sistemos gautas kodo eilučių skaičius apytiksliai lygus 23000. Iš jų apytiksliai 8770 kodo eilučių sudaro rankomis kurto kodo. Santykis tarp rankomis kurto kodo ir generuoto kodo

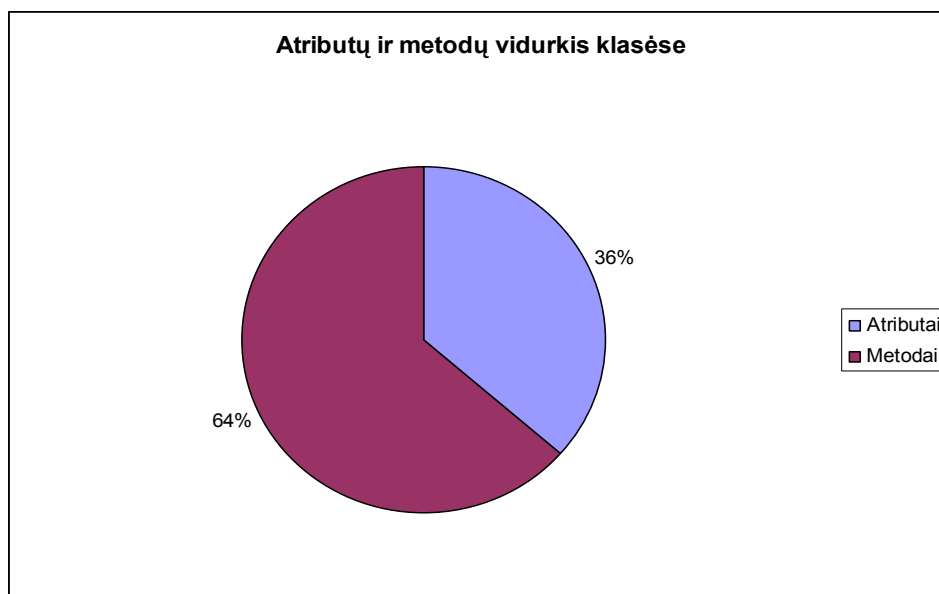
pateiktas žemiau esančiame grafike. Kaip matyti didžioji dalis veiklos klasių yra sugeneruota, taip sutaupyta daug laiko kūrimo procese. Sugeneruotos klasės pritaikytos pakartotiniam panaudojimui.



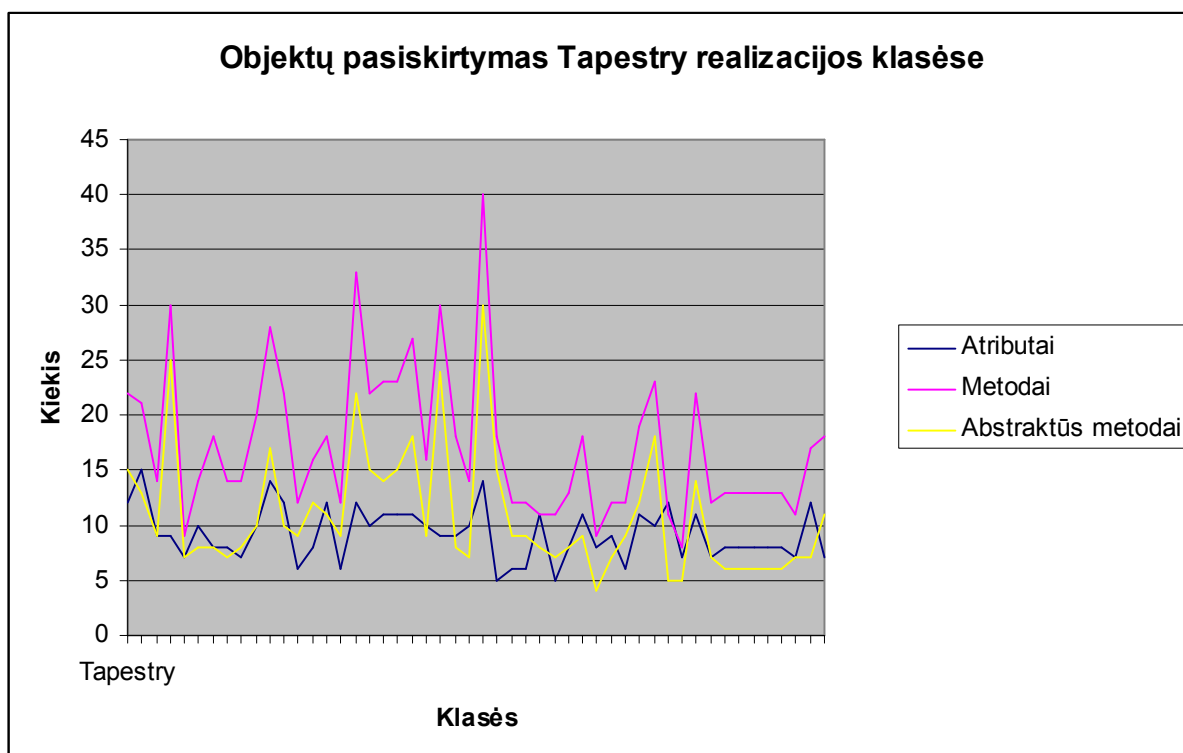
Analizuosime rankomis kurta kodą. Bendras atributų kiekis kode yra 777, o metodų – 1364. Iš kito grafiko, kuriame pateiktas objektų pasiskirstymas klasėse, matyti, jog atributų kiekis ir metodų kiekis yra proporcingas. Taip yra, nes dauguma metodų realizuoja JavaBeans technologiją. Kūrimo aplinka Eclipse, kurią naudojantis buvo kurta sistema, suteikia galimybę generuoti šiuos JavaBeans technologija paremtus metodus. Ši dalis taipogi labai palengvina kūrimo procesą, kai dalį rankinio kodo galima sukurti naudojant kodo generatorius.



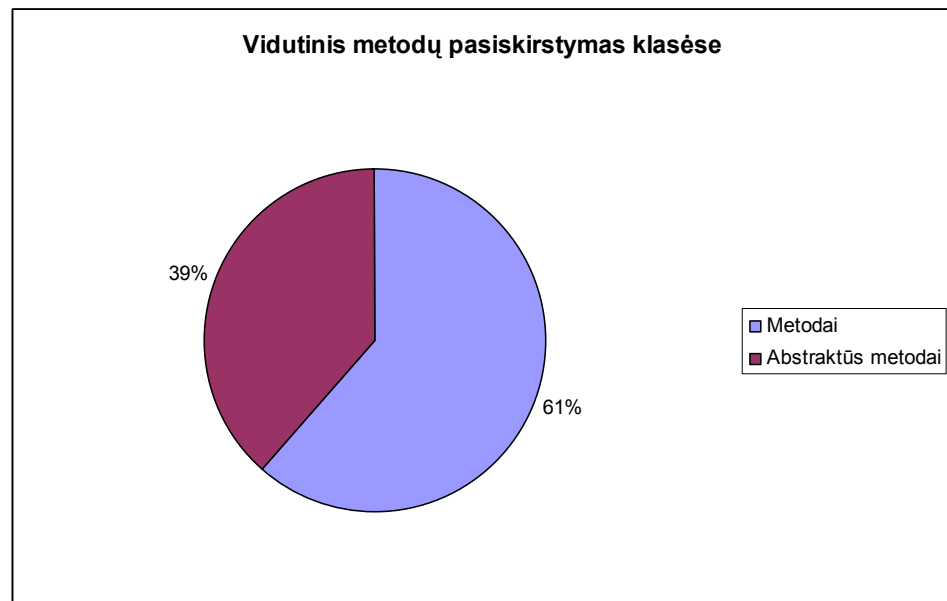
Suminiai rezultatai pateikti „Atributų ir metodų vidurkis klasėse“ grafike.



Paskaičiuokime kaip pasiskirsto objektai klasėse, kurios realizuoja Tapestry karkaso puslapius. Šiose klasėse atsiranda dar viena objektų grupė, tai abstraktūs metodai. Jais yra realizuojama puslapiuose naudojamų objektų veikla. Šie metodai taipogi pagrįsti JavaBeans technologija. Tik skirtumas tas, jog šių klasių realizacija palikta Tapestry karkasui, na o funkcinio požiūriu, tai objektai naudojami tiesiogiai HTML puslapyje. Žemiau pateiktame grafike matyti, kiek abstraktūs metodai sudaro visų metodų kiekio.



Žemiau pateiktame grafike, pateikiama kiek vidutiniškai abstraktūs metodai sudaro bendrų metodų kiekio.



Užbaigiant šį skyrių, galime apibendrinti, jog Tapestry ir Torque technologijų naudojimas, palengviną sistemos kūrimą. Programuotojui tereikia sukurti apytiksliai vieną trečdalį kodo, visa kita atlieka pasirinktos technologijos. O kad jos atliktų tai ką reikia, programuotojas turi teisingai sutvarkyti konfigūracinius failus.

7. IŠVADOS

- Norint sukurti konferencijų sistemą, automatizuojančią administratoriaus darbą, buvo atlikta panašių sistemų ir galimų realizacijos būdų analizė, kurios pagrindu apibrėžti kuriamos sistemos reikalavimai. Sistema neįgyvendina maksimalaus galimų funkcijų rinkinio, tačiau turi keletą savybių, kurių trūksta esamoms tokio tipo sistemoms.
- Sistemai kurti pasirinktas Tapestry karkasas, kuris sudaro galimybes plėsti ją ateityje, ir Torque technologija, kuri leidžia lengvai keisti komandų, dirbančių su duomenų bazės duomenimis, funkcionalumą.
- Sukurtas konferencijų sistemos modelis, kurį sudaro veiklos procesų modelis, susidedantis iš tipinių etapų, ir etapų bei žinučių siuntimo valdymo agentų modelis. Keičiant modelio parametrus, galima nustatyti etapų pradžią ir pabaigą, agentų veikimo dažnumą, atsisakyti nereikalingų etapų. Administratoriui paliekama galimybė, reikalui esant, veikti lygiagrečiai arba pakeisti agentų veiksmus.
- Sukurta ir įdiegta konferencijų sistema, leidžianti automatizuotai vykdyti pasikartojančius veiklos procesus – žinučių siuntimą nustatytais laiko momentais, recenzijų apdorojimą. Sistemą galima naudoti keliomis kalbomis, iš karto organizuoti kelias konferencijas, išsaugoti jų istoriją.
- Sistemos kokybė atitinka lauktus rezultatus. Atlikti skaičiavimai ir eksperimentas parodė, kad sistema mažina administratoriaus laiko sąnaudas ir gali būti nesunkiai plečiama.
- Sukurtą sistemą numatyta panaudoti realioms konferencijoms, kurios vyks lietuvių ir anglų kalbomis.
- Tikėtina, kad pasiūlyti automatizavimo modeliai sėkmingai bus naudojami konferencijų organizavimo rinkoje bei kuriant kitas panašias sistemas, kur reikia kompiuterizuoti laike valdomus procesus.
- Sukurta sistema lengvai pritaikoma kitoms, panašius procesus atliekančioms, sritims. Vienas iš pavyzdžių – akademinė mokytojo ir mokinio sistema.
- Konferencijos tematika padarytas pranešimas.

8. LITERATŪRA

1. Konferencijų valdymo sistema [interaktyvus]. [Vokietija]: The Conference Management System, 2003 [žiūrėta 2006-02-25]. Prieiga per internetą: <<http://www.confmaster.net/t3/index.php?id=64>>
2. Konferencijų valdymo sistema [interaktyvus]. [Graikija]: Confious, 2006 [žiūrėta 2006-02-25]. Prieiga per internetą: <<http://confioussite.ics.forth.gr/index.php>>
3. Tapestry projektas [interaktyvus]. [Amerika]: *Tapestry project*, 2006 [žiūrėta 2006-03-10]. Prieiga per internetą: <<http://jakarta.apache.org/tapestry/>>
4. Howard M. Lewis Ship, *Tapestry in Action*. JAV, 2004. 585p. ISBN 1-932394-11-7.
5. Ford, N. *Art Of Java Web Development*. JAV, 2004. 627p. ISBN 1-932394-06-0.
6. Kochmer, C. *An Introduction to Struts*. [interaktyvus]. 2002 [žiūrėta 2006-02-26]. Prieiga per internetą: <<http://www.jspinsider.com/content/jsp/struts/strutsintro.jsp>>
7. Struts karkaso struktūra [interaktyvus]. [Prancūzija]: Struts 1.1 Controller UML diagrams, 2003 [žiūrėta 2006-02-26]. Prieiga per internetą: <<http://rollerjm.free.fr/pro/Struts11.html>>
8. Raible, M. *Comparing Web frameworks: Struts, Spring MVC, WebWork, Tapestry & JSF*. JAV, 2005.
9. Vasilecas, O. et al. *Proceeding of the Thirteenth International Conference on Information Systems Development: tarptautinės konferencijos pranešimų medžiaga*. Vilnius, 2004, p. 399 – 406.
10. Wesley, A. *UML Profile for Framework Architecture*. Didžioji Britanija, 2001. 240p. ISBN 0-201-67518-8.
11. Budreika, R. et al. *11-oji tarpuniversitetinė doktorantų ir magistrantų konferencija: konferencijos pranešimų medžiaga*. Kaunas, 2006, II dalis, p. 121 – 126.
12. Torque projektas [interaktyvus]. [Amerika]: Torque project, 2006 [žiūrėta 2006-05-01]. Prieiga per internetą: <<http://db.apache.org/torque/>>
13. Howard M. Lewis Ship. *Developers Guide*. JAV, 2004.
14. Howard M. Lewis Ship. *User Guide*. JAV, 2004.
15. Hibernate [interaktyvus]. [Amerika]: Hibernate, 2005 [žiūrėta 2006-05-05]. Prieiga per internetą: <<http://www.hibernate.org/>>

Automated business processes on the Web: conference system

SUMMARY

More and more business processes are moved to the web to have all information easily accessible to everyone. High requirements are raised for business process automation – software systems should be easy to use, multilingual, expandable, and flexible. They should do the major part of organizational tasks. In the current work, the conference system model is proposed and implemented using Tapestry framework, which gives conceptual clarity and possibilities to expand system in the future. Much attention is devoted to process automation – sending of messages at predefined time moments, calculations, process stages management is executed by software agents. Only parameters are set by the administrator. The system enables executing of several conferences simultaneously and saves their history.

SANTRUMPŲ IR TERMINŲ ŽODYNAS

Apache	daugiafunkcinis interneto paslaugų serveris
Eclipse	universali programavimo aplinka
IADIS	<i>international association for development of the information society</i> – informacijos visuomenės vystymo tarptautinė asociacija
Internet Explorer	Microsoft sukurta interneto naršyklė
ISD	informacijos sistemų katedra
Java	objektinė programavimo kalba
JavaBean	specialios architektūros Java klasė
JavaSpring	Java kalba realizuotas karkasas
J2EE	kolekcija Java technologijų aukštesnio lygio architektūrai kurti
JSP	<i>JavaServer Pages</i> – technologija, skirta kurti dinامينius web puslapius
JSF	<i>JavaServer Faces</i> – technologija, skirta JavaServer puslapių varotojų sąsajos kūrimo supaprastinimui
Hibernate	JAVA ir .Net kalboms skirta technologija prieigai prie duomenų bazės
HiveMind	šablonas, realizuojamas Java kalba, skirtas formuoti servisus bei valdyti kitų šablonų branduolį
HTML	programavimo kalba kurti internetiniams puslapiams
Firefox	internetu naršyklė
MVC arba Model 2	<i>Model-View-Controller</i> – architektūrinis sprendimas (<i>design template</i>) atskiriantis logiką (<i>model</i>), pateikimą (<i>view</i>), valdiklį (<i>controller</i>), naudojamas interneto aplikacijose
MySQL	nemokama duomenų bazė
Opera	internetu naršyklė
PHP	programavimo kalba, skirta internetiniams puslapiams kurti
Servletų talpykla	internetu serveris apdorojantis dinامينius JSP puslapius bei Java komponentus, naudojamus internetinei aplikacijai.

SQL	struktūrizuota užklausų kalba
Struts	atvirojo kodo MVC šablonas
Tapestry	„Jakarta“ projekto dalyvių sukurtas MVC šablonas
Tomcat	„Jakarta“ projekto dalyvių sukurtas servletų talpykla
Torque	„Jakarta“ projekto dalyvių sukurta technologija, leidžianti iš programos kodo kreiptis į duomenų bazę be tiesioginių SQL užklausų
Turbine	„Jakarta“ projekto dalyvių sukurtas servletų karkasas internetinėms svetainėms kurti
VLDB	<i>Very Large Data Base</i> – nekomercinis konferencijų portalas
XML	išplečiamoji formatavimo kalba
WebWork	šablonas, realizuojamas Java kalba, skirtas internetinius puslapius kurti
WORLDCOMP	<i>world academy of science</i> – daugelio konferencijų sistema

I PRIEDAS. STRAIPSNIS

Straipsnis, atspausdintas Vilniaus Universiteto Kauno Humanitarinio Fakulteto išleistame leidinyje „Informacinės technologijos 2006, 11-oji tarpuniversitetinė doktorantų ir magistrantų konferencija, konferencijos pranešimų medžiaga“. Kaunas, 2006, II dalis, p. 121 – 126.

Automatizuoti veiklos procesai internete: konferencijų sistema

Rytis Budreika, Lina Nemuraitė

Kauno Technologijos universitetas
Informacijos sistemų katedra

Vis daugiau ir daugiau veiklos perkeliama į internetą, kad informacija būtų lengvai prieinama kiekvienam suinteresuotam asmeniui. Veiklos procesų kompiuterizavimui keliami dideli reikalavimai – tokios sistemos turi būti lengvai naudojamos vartotojų, daugiakalbės, plečiamos, lanksčios pokyčiams. Be to, jos gali pačios atlikti didelę dalį organizacinių darbų. Straipsnyje pateikiamas konferencijų informacinės sistemos modelis ir jo įgyvendinimas naudojant Tapestry karkasą, kuris suteikia sistemai konceptualų aiškumą ir galimybes plėsti ją ateityje. Didelis dėmesys skiriamas veiklos proceso automatizavimui – visą žinučių siuntimą, įvertinimų skaičiavimą reikiama laiko momentais vykdo programos komponentai, administratorius tik nustato jų parametrus. Sistema leidžia vykdyti iš karto kelias konferencijas ir išsaugo jų istoriją.

2 Įvadas

Veiklos perkėlimas į internetą šiandien yra ne naujovė, bet būtinybė. Internetas tapo vienu iš pagrindinių informacijos apskaitimo būdų, prieinamu dideliame žmonių skaičiui.

Konferencijų organizavimas jau ne pirmi metai vyksta internetu ir yra gana didelė tokių sistemų įvairovė – nuo paprastų informacinių puslapių iki didelių, sudėtingų sistemų. Tačiau Lietuvoje šiuo metu lietuviškų konferencijų sistemų nėra, nors paklausa yra didelė. Rengiant universitetų darbuotojų, magistrantų, doktorantų konferencijas, pasikliaujama tik elektroniniu paštu.

Pasaulinio lygio konferencijos turi gana išvystytas sistemas [1], [2]. Tačiau šios sistemos dažniausiai pritaikytos tam tikram specifiniam procesui, tam tikra kalba, ir nėra tinkamos nedidelėms konferencijoms, vykstančioms lietuvių kalba. Sukurti informacinę konferencijos puslapį nesunku, tačiau to nepakanka. Konferencijos organizatoriams tenka didelis darbo krūvis, kurį gali atlikti ar pagreitinti tinkama programinė įranga.

Vienas pagrindinių konferencijų organizavimo reikalavimų – nuolatinis organizatorių ir konferencijos dalyvių bendravimas, kuris dažniausiai vyksta elektroniniu paštu. Elektroninių laiškų siuntimą tikslinga automatizuoti, kad sistema pati išsiųstų standartinius laiškus ir sumažintų organizatorių apkrovimą. Šiame darbe siekiama maksimaliai automatizuoti laiškų siuntimą, iš anksto nustatant datas, kada, kokiu periodiškumu ir kokioms dalyvių grupėms reikia siųsti tam tikrų tipų pranešimus.

Kitas šio darbo tikslas – unifikuoti konferencijos valdymą. Tipiniai etapai (paskelbimas, registracija, straipsnių priėmimas, recenzavimas ir taip toliau) skirtingose konferencijose išsidėsto įvairia tvarka ir įvairiais laiko periodais. Kitaip tariant, konferencijos skiriasi savo valdymo taisyklėmis. Turint galimybes nustatyti atitinkamus etapų parametrus (pradžią, pabaigą), galima juos įvairiais būdais išdėlioti laike, atsakyti nereikalingų etapų. Reikalui esant, parametrus galima keisti, pavyzdžiui, pratęsti straipsnių priėmimą. Pagal nustatytus parametrus sistema suteikia vartotojams galimybes tam tikram etape atlikti tam tikrus veiksmus ir padaro juos neprieinamus, etapui pasibaigus. Tokiu būdu sistema pati save „administruoja“, aišku, palikdama administratoriui galimybę tuos nustatymus pakeisti. Automatizuotai atliekamas recenzijų rezultatų skaičiavimas, organizatoriams paliekant tik galutinį įvertinimą. Sistemą galima pritaikyti įvairioms konferencijoms įvairiomis kalbomis, vienu metu organizuoti kelias konferencijas ir išsaugoti jų istoriją.

Galiausiai, sistemą buvo siekiama sukurti taip, kad ją būtų lengva plėsti ateityje. Todėl sukurtam konferencijų sistemos modeliui įgyvendinti pasirinktas Tapestry – interneto sistemų kūrimo karkasas, kuris leidžia atskirti puslapių valdymą nuo veiklos logikos ir sudaro prielaidas lengvam, aiškiam kūrimui.

Konferencijų sistemos modelį galima pritaikyti ir kitoms sritims, pavyzdžiui, mokymo proceso organizavimo ir žinių vertinimo sistemai, kurioje bendravimas vykėtų tarp studentų ir dėstytojų.

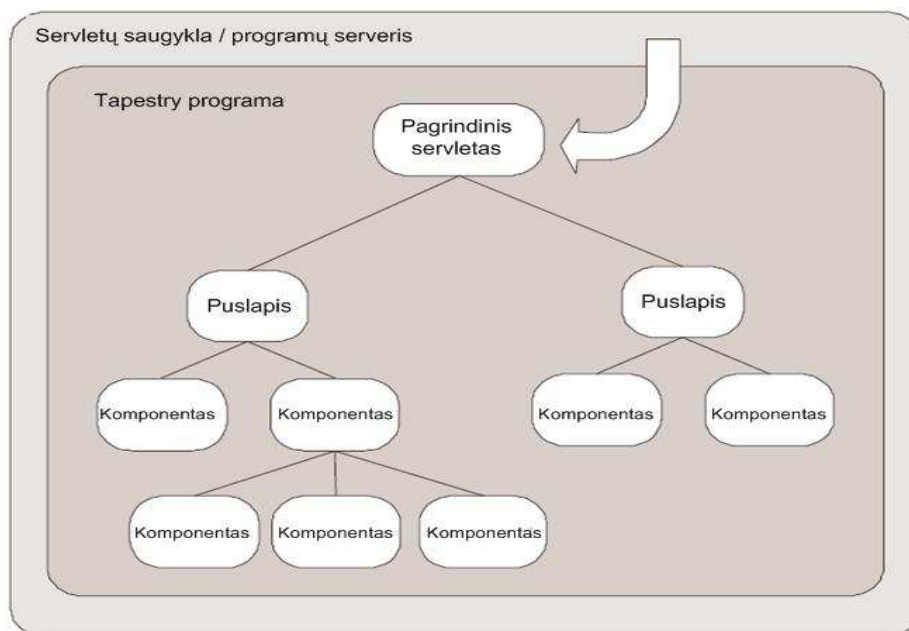
3 Tapestry karkasas

Tapestry yra atvirojo kodo karkasas skirtas kurti dinamines, lengvai keičiamas interneto sistemas [3]. Ši technologija realizuojama Java programavimo kalboje. Tapestry paremta standartine Java servletų technologija. Ji veikia bet kurioje servletų saugykloje ar programų serveryje.

Interneto sistema, sukurta naudojant Tapestry, skaidoma į apibrėžtą skaičių puslapių, kur kiekvienas puslapis gali būti sudarytas iš komponentų [4]. Komponentai gali būti sudaryti iš kitų komponentų. Pavyzdys pateiktas 1 paveiksle.

Šiuo karkasu paremtos sistemos puslapių šablonai kuriami naudojant paprastą HTML technologiją, įterpiant nedidelius gabaliukus Java kodo. Java kodo kiekis HTML šablonuose gali būti dar labiau sumažintas naudojant XML failus, kurie veikia kaip tarpininkai tarp HTML ir Java kodo [5].

Viena iš Tapestry karkaso privalumų yra „grynas Java kodas“ – funkcionalumas rašomas Java kodu visiškai atskirai nuo HTML.



1 paveikslas. Tapestry karkaso komponentinis išskaidymas

Technologija pritaikyta lengvai kurti naujus komponentus, kurie gali būti pakartotinai naudojami – kas labai sumažina klaidų kiekį programoje bei pagreitina kūrimo procesą. Komponentinis programavimas leis ateityje lengvai keisti bei plėsti jau esamą sistemą.

Tapestry gali būti naudojamas tiek mažose sistemose, tiek didelės apimties projektuose.

Lengvai integruojamas su J2EE, HiveMind, JavaSpring technologijomis.

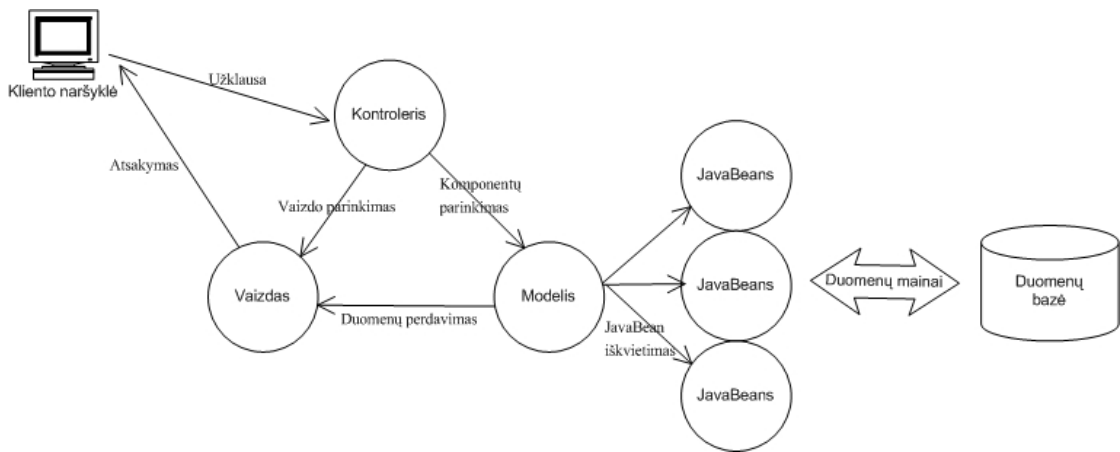
Kaip ir dauguma kitų karkasinių sistemų taip ir Tapestry naudoja modelio-vaizdo-valdiklio šabloną (MVC – model-view-controller) [5].

2 paveiksle pateikiame pateikiame schemą, kaip veikia Tapestry karkasu paremta sistema – kaip ji bendrauja su vartotoju, veiklos logika, duomenų baze bei kaip komponentai sąveikauja tarpusavyje. Sistemos veikimas remiasi MVS šablonu, kuris valdo visus procesus. Vartotojui pateikus užklausa, ją apdoroja valdiklis, išskviesdamas atitinkamą komponentą. Komponentas pateikiamas vaizdo kontrolieriui – čia suformuojamas dinaminis HTML puslapis iš HTML šablono bei Java kodo. Suformuotas puslapis siunčiamas atgal į vartotojo naršyklę.

Tapestry karkasą pasirinkome dėl jo pranašumų prieš kitus interneto puslapių kūrimo karkasus. Vienas iš rimtesnių kandidatų dar buvo Struts karkasas. Trumpai apžvelgsime šį karkasą bei pateiksime abiejų karkasų palyginimą.

Struts galima laikyti Java interneto projektų standartu - šiuo metu jis labiausiai paplitęs pasaulyje. Tai yra pakankamai lengvai perprantama technologija. Internete apie jį galima rasti daug literatūros [6], [7].

Patogiai naudojamos HTML kalbos žymės (angl. tags). Tačiau norint pilnai išnaudoti jų privalumus tenka naudoti Struts žymes, o tai tampa trūkumu, neleidžiančiu atskirti kodo nuo dizaino. Sudėtingai naudojamas „ActionForms“ valdiklis, įnešantis daug neaiškumo. Tai galima paaiškinti daugybę dinaminių dalių, užtrunka nemažai laiko kol išsiaiškini veikimo principą. Kaip dar vieną technologijos trūkumą reikėtų paminėti sunkų testavimą bei klaidų aptikimą. HTML puslapiuose yra nemažai kodo eilučių [8]. Kadangi Struts nėra komponentinis karkasas, tai apsunkins sistemos komponentų plečiamumą.



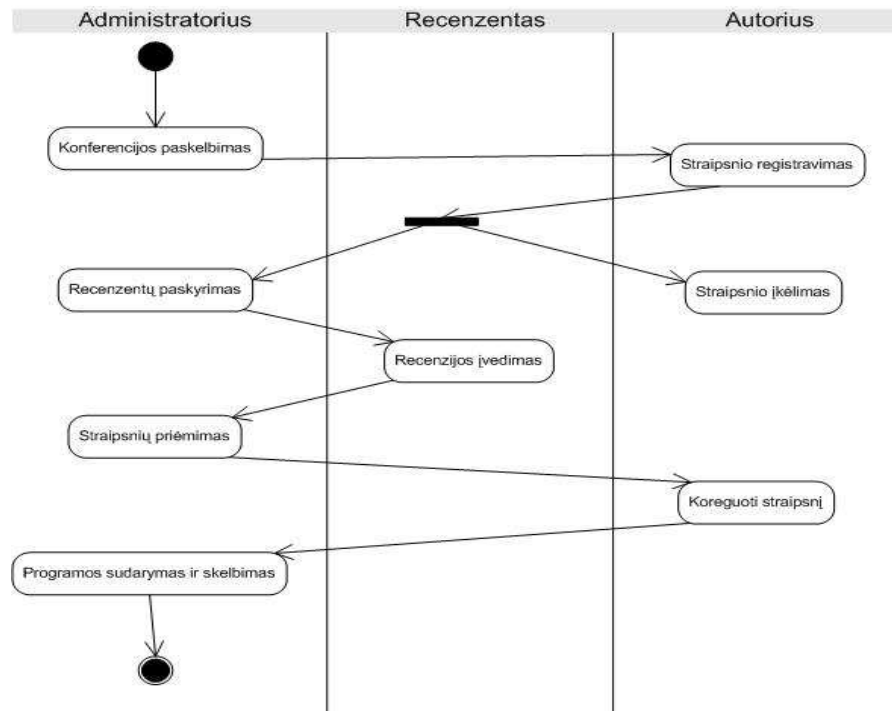
2 paveikslas. Tapestry karkasu paremtos sistemos komponentų sąveika

4 Konferencijų organizavimo etapai

Konferencijų organizavimas yra jau nusistovėjusių procesų rinkinys [9]. Šie procesai kiekvienai konferencijai gali nedaug skirtis, tačiau standartinę konferenciją sudaro šie pagrindiniai etapai:

- Konferencijos skelbimas
- Straipsnio registravimas
- Recenzentų paskyrimas
- Straipsnio įkėlimas
- Recenzijos įvedimas
- Straipsnių priėmimas
- Straipsnio koregavimas
- Programos sudarymas

Šie etapai bei jų vykdomieji asmenys pateikti 3 paveiksle.



3 paveikslas. Konferencijos rengimo etapai

Kiekviename iš šių etapų galima atlikti tam tikrus veiksmus. Šiuos etapus nustatinėja administratorius bei apie jų pradžią ir pabaigą informuoja atitinkamą konferencijos dalyvį. Tačiau tai yra pasikartojantis ir daug resursų reikalaujantis darbas.

Šių etapų valdymas gali būti automatizuotas, paliekant administratoriui tik konfigūravimo darbus.

Kiekvieną etapą galima detalizuoti bei iš visų jų išvesti bendrą formulę, pagal kurią sukursime šabloną automatizuotam etapui.

Išskiriame tokius konferencijos etapų bendrus parametrus: pavadinimas, pradžios data, pabaigos data, interesantų grupė. Tokiu būdu aprašę kiekvieną etapą, galime juos suvesti į sistemą. Sistemos administravimo langų pavyzdžiai pateikti 4 paveiksle.

The image displays four screenshots of a web-based conference management system interface. Each screenshot shows a different administrative view with a sidebar menu on the left and a main content area.

1. Konferencijos (Conference List): Shows a table of conferences with columns: Pavadinimas, Dalyviai, Temos, Pradžia, Pabaiga, Baigta, Vieša.

Pavadinimas	Dalyviai	Temos	Pradžia	Pabaiga	Baigta	Vieša
2005 metai	2005-12-01	2005-12-30	Taip	Taip
Bandomoji	2005-12-01	2006-12-01	Taip	Taip
Šventiniai išpūdžiai	2005-12-24	2006-01-10	Ne	Taip

2. Konferencijos dalyvių sąrašas (Conference Participants List): Shows a table for a specific conference: 'Konferencija: Šventiniai išpūdžiai'. Columns: Dalyvis, Tema, Straipsnis, Recenzentas.

Dalyvis	Tema	Straipsnis	Recenzentas
Kristina Anastasjeva	Kalėdos	Tradicijos	±
Rytis Budreika	Nauji metai	Raudonojo šuns metai	Petras Stankis
Rytis Budreika	Nauji metai	Testas	+

3. Pranešimų administravimo langas (Submission Management): Shows a table of submissions with columns: Pavadinimas, Tekstas, Lygmuo, Data, Siųsti.

Pavadinimas	Tekstas	Lygmuo	Data	Siųsti
PRANESIMAS 2	dirbam	Administratorius	2005-12-27	Ne X
PRANESIMAS	dirbam	Administratorius	2005-12-28	Ne X

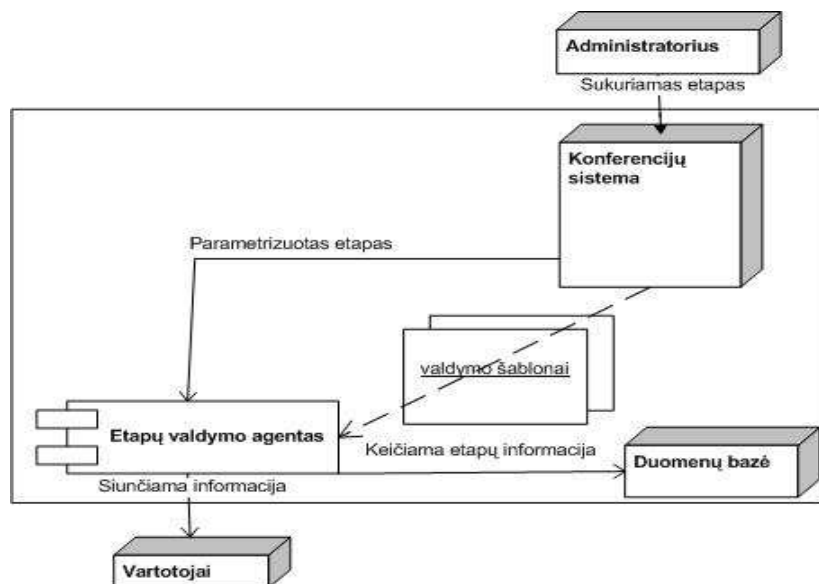
4 paveikslas. Konferencijos sistemos keli administravimo langai

5 Automatizuojami sistemos veiklos procesai

Kuriant automatinę sistemą, reikia nustatyti kokius procesus norime automatizuoti. Pirmiausia tai turėtų būti etapų valdymas laike (etapus pateikėme šio straipsnio 3 skyrelyje). Taipogi reikia automatizuoti informacijos siuntimą vartotojams priklausomai nuo vykstančio etapo, administratoriui paliekant tik nustatymo darbus. Galiausiai, automatizuojama straipsnių recenzijų vertinimo sistema, kuri pateikia suminius straipsnių vertinimų rezultatus. Toliau pateikiami šių procesų automatizavimo modeliai.

Kiekvieną iš etapų galima laikyti automatizuoto modelio kintamuoju, kur modelis vaizduoja savarankišką sistemą – informacijos apdorojimo ir valdymo agentą. 3 skyrelyje pateikėme etapo suskaidymą į parametrus, dabar pateiksime modelį tiems etapams valdyti.

5 paveiksle pateiktame modelyje vyksta tokie procesai: duodama valdymo signalas iš vartotojo naujo etapo sukūrimui ar jo redagavimui. Tuomet informacija apie etapą bei valdymo šablonai perduodami į etapų valdymo agentą. Valdymo šablonais nurodomi sąlyginiai laiko vienetai bei informacija kuri turi būti siunčiama vartotojams nurodyto etapo metu. Agentas renka informaciją apie kiekvieną iš etapų bei pagal ją atlieka tam tikrus veiksmus – keičia informaciją duomenų bazėje ar siunčia ją vartotojams.

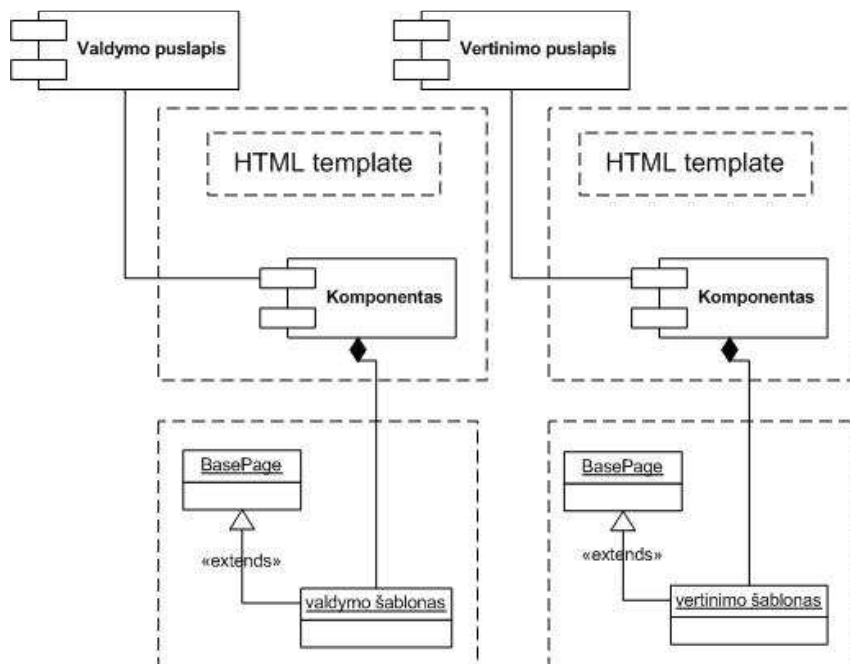


5 paveikslas. Etapų valdymo modelis

Automatizuotoje vertinimų sistemoje panašiai kaip ir etapų valdymo modelyje yra vertinimo šablonai. Šie šablonai sukuriama konferencijos pradžioje ir naudojami recenzavimo etape vertinant straipsnius. Surinkta informacija vėliau bus naudojama skaičiuojant straipsnių recenzijų vertinimus bei pateikiamos suminės reikšmės administratoriui.

6 Automatizuotų procesų integravimas į Tapestry karkasu sukurtą sistemą

Automatizuoto modelio šablonai realizuojami Tapestry komponentais bei integruojami į visą sistemą. Kaip jau minėta straipsnio pradžioje, vienas iš Tapestry karkaso privalumų yra komponentų naudojimas. Komponentus nesunku integruoti į sistemą ar pakeisti vieną komponentą kitu. Integravimo modelis pateiktas 6 paveiksle.



6 paveikslas. Automatizuotų procesų integravimo modelis

7 Sistemos efektyvumo analizė

Norint suprasti automatizuotos veiklos procesų privalumus, galima palyginti neautomatizuotam procesui atlikti sugaištą laiką su laiku, sugaištam tam procesui vykdyti automatizuotu būdu.

Norint įvertinti programinių agentų efektyvumą, buvo skaičiuojamas vidutinis recenzijų vertinimų skaičiavimo, informacijos siuntimo, žinučių apie svarbias datas siuntimo laikas automatizuotu būdu (naudojant agentus) ir neautomatizuotu (kai tą daro administratorius). Recenzijų vertinimo skaičiavimui automatizuoto ir neautomatizuoto vykdymo laiko santykis buvo 1:99, informacijos siuntimui – 1:9, žinutėms apie svarbias datas 1:19. Pati automatizuota sistema nereikalauja labai daug administratoriaus laiko. Pagrindinį laiką sudaro sistemos konfigūravimas.

Kitas labai svarbus faktorius yra sistemos plečiamumas. Žinodami teorines Tapestry karkaso plėtimo galimybes, pabandėme įvertinti realias laiko sąnaudas reikalingas funkcionalumui ir dizainui plėsti. Tam buvo apklausti keli programuotojai, gerai susipažinę su šiuo karkasu. Jie vertino tokius sistemos pakeitimus: naujo etapo šablono įtraukimas į sistemą, naujo valdymo šablono įtraukimas, dizaino pakeitimas (lyginome su laiku, sugaištu pradinio sistemos kūrimo metu): naujo etapo šablono įtraukimas: ~2:5; naujo valdymo šablono įtraukimas: ~3:5; dizaino keitimas: ~3:10. Gauti rezultatai patvirtina, kad sistema yra lengvai plečiama.

8 Išvados

- Sukurta konferencijų sistema, leidžianti automatizuotai vykdyti pasikartojančius veiklos procesus – žinučių siuntimą nustatytais laiko momentais, recenzijų apdorojimą. Sistemą galima naudoti keliomis kalbomis, iš karto organizuoti kelias konferencijas, išsaugoti jų istoriją
- Sistema sukurta naudojant Tapestry karkasą, kuris sudaro galimybes plėsti ją ateityje.
- Atlikti skaičiavimai ir eksperimentas parodė, kad sistema efektyviai mažina administratoriaus gaištamo laiko sąnaudas.
- Sukurtą sistemą numatyta panaudoti realioms konferencijoms, kurios vyks lietuvių ir anglų kalbomis.
- Tikėtina, kad pasiūlyti automatizavimo modeliai sėkmingai bus naudojami konferencijų organizavimo rinkoje.

Literatūros sąrašas

- [1] The Conference Management System. [interaktyvus]. [žiūrėta 2006-02-25]. Prieiga internetu: <http://www.confmaster.net/t3/index.php?id=64>
- [2] The Conference Management System. [interaktyvus]. [žiūrėta 2006-02-25]. Prieiga internetu: <http://confioussite.ics.forth.gr/index.php>
- [3] Tapestry project. [interaktyvus]. [žiūrėta 2006-03-10]. Prieiga internetu: <http://jakarta.apache.org/tapestry/>
- [4] Howard M. Lewis Ship. „Tapestry in Action“, 2004
- [5] Neal Ford. Art Of Java Web Development 2004
- [6] Kochmer C. An Introduction to Struts. [interaktyvus]. [žiūrėta 2006-02-26]. Prieiga internetu: <http://www.jspinsider.com/content/jsp/struts/strutsintro.jsp>
- [7] Struts 1.1 Controller UML diagrams. [interaktyvus]. [žiūrėta 2006-02-26]. Prieiga internetu: <http://rollerjm.free.fr/pro/Struts11.html>
- [8] Matt Raible. „Comparing Web frameworks: Struts, Spring MVC, WebWork, Tapestry & JSF“.
- [9] Olegas Vasilecas and others. Proceeding of the Thirteenth International Conference on Information Systems Development, 2004: 399 – 406 p.
- [10] Addison Wesley “UML Profile for Framework Architecture”

Automated processes on the web: conference system

More and more business processes are moved to the web to have all information easily accessible to everyone. High requirements are raised for business process automation – software systems should be easy to use, multilingual, expandable, and flexible. They should do the major part of organizational tasks. In the current work, the conference system model is proposed and implemented using Tapestry framework, which gives conceptual clarity and possibilities to expand system in the future. Much attention is devoted to process automation – sending of messages at predefined time moments, calculations, process stages management is executed by software agents. Only parameters are set by the administrator. The system enables executing of several conferences simultaneously and saves their history.