

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Vytenis Darulis

**Veiklos taisyklių valdymo posistemių praturtinta
katedros darbo organizavimo
informacinė sistema**

Magistro darbas

Darbo vadovas

doc. dr. Tomas Skersys

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMACIJOS SISTEMŲ KATEDRA

Vytenis Darulis

**Veiklos taisyklių valdymo posistemių praturtinta
katedros darbo organizavimo
informacinė sistema**

Magistro darbas

Recenzentas

2010-05-28

Vadovas

doc. dr. Tomas Skersys

2010-05-28

Atliko

IFM-4/4 gr. stud.

Vytenis Darulis

2010-05-28

Kaunas, 2010

Turinys

1. ĮVADAS	6
2. INFORMACIJOS SISTEMŲ KATEDROS VEIKLOS IR INFORMACINIŲ POREIKIŲ ANALIZĖ	9
2.1. ANALIZĖS TIKSLAS	9
2.2. TYRIMO SRITIS, OBJEKTAS IR PROBLEMA	9
2.3. TYRIMO OBJEKTO ANALIZĖ	10
2.4. TYRIMO TIKSLAS IR UŽDAVINIAI	11
2.5. ANALIZĖS METODAI	11
2.6. VARTOTOJŲ ANALIZĖ	12
2.6.1. <i>Vartotojų aibė, tipai ir savybės</i>	12
2.6.2. <i>Vartotojų tikslai ir problemos</i>	12
2.7. VEIKLOS TAISYKLIŲ ANALIZĖ	13
2.8. ESAMŲ SPRENDIMŲ ANALIZĖ	15
2.8.1. <i>Quick rules</i>	15
2.8.2. <i>Infrex</i>	16
2.8.3. <i>Versata Logic Suite</i>	18
2.8.4. <i>ILOG Rules</i>	18
2.8.5. <i>Blaze Advisor</i>	19
2.9. SIEKIAMAS SPRENDIMAS	22
2.10. ANALIZĖS IŠVADOS	25
3. DĖSTYTOJŲ DARBO ORGANIZAVIMO INFORMACINĖS SISTEMOS REIKALAVIMAI ...	26
3.1. REIKALAVIMŲ SPECIFIKACIJA	26
3.1.1. <i>Panaudojimo atvejų specifikacija</i>	26
3.1.2. <i>Nefunkcinių reikalavimų specifikacija</i>	46
3.2. DALYKINĖS SRITIES ESYBIŲ MODELIS	47
3.3. REIKALAVIMŲ ANALIZĖS APIBENDRINIMAS	48
4. DĖSTYTOJŲ DARBO ORGANIZAVIMO INFORMACINĖS SISTEMOS PROJEKTAS	48
4.1. SISTEMOS ARCHITEKTŪROS PROJEKTAS	48
4.1.1. <i>Sistemos loginė architektūra</i>	48
4.1.2. <i>Vartotojo paslaugos</i>	49
4.1.3. <i>Duomenų paslaugos</i>	50
4.2. DETALUS PROJEKTAS	51
4.2.1. <i>Darbu organizavimo posistemio detalus projektas</i>	51
4.2.2. <i>Taisyklių valdymo posistemio detalus projektas</i>	54
4.2.3. <i>Blaze Advisor taisyklių serverio detalus projektas</i>	54
4.3. SISTEMOS ELGSENOS MODELIS	56
4.4. REALIZACIJOS MODELIS	69
5. VEIKLOS TAISYKLIŲ INTEGRACIJOS METODIKA	70
5.1. VEIKLOS TAISYKLIŲ KŪRIMAS	70
5.1.1. <i>Blaze Advisor saugyklos ir projekto sukūrimas</i>	70
5.1.2. <i>Blaze Advisor integracija su duomenų baze ir klasių sukūrimas</i>	72
5.1.3. <i>Pagrindinių taisyklių sukūrimas</i>	73
5.1.4. <i>Taisyklių srauto sukūrimas</i>	75
5.1.5. <i>EJB diegimas į taikomųjų programų serverį</i>	75
5.2. WEB PROGRAMOS PARUOŠIMAS	76
6. SISTEMOS REALIZACIJA	77
6.1. BLAZE ADVISOR BUILDER POSISTEMIO VEIKIMO APRAŠYMAS	77
6.2. DARBO ORGANIZAVIMO POSISTEMIO VEIKIMO APRAŠYMAS	78
6.3. TESTAVIMO MODELIS BEI DUOMENYS, KONTROLINIS PAVYZDYS	82
6.3.1. <i>Prisijungimas</i>	83
6.3.2. <i>Menu</i>	84
6.3.3. <i>Taisyklių valdymas</i>	85
6.3.4. <i>Taisyklių veikimas</i>	85

6.4.	SUKURTOS SISTEMOS REALIZACIJOS APIBENDRINIMAS	86
6.5.	TAIKYMO REKOMENDACIJOS	86
7.	IŠVADOS.....	88
8.	LITERATŪRA	90
9.	PRIEDAI.....	91
9.1.	BLAZE ADVISOR TAISYKLIŲ KŪRIMO VADOVAS	91
9.2.	BLAZE ADVISOR TAISYKLIŲ DIEGIMO Į TAISYKLIŲ SERVERĮ VADOVAS	100
9.3.	STRAIPSNIS „VEIKLOS TAISYKLIŲ TAIKYMAS INTERNETINIŲ INFORMACINIŲ SISTEMŲ KŪRIME“	108

Department's Work Planning Information System enhanced with Business Rules

SUMMARY

At the Information Systems Department of Kaunas University of Technology the employees are simultaneously working in several projects, carry out research activities, deliver lectures, participate in conferences and seminars. It is essential for the staff of the department that to perform all assignments within the specified time frame and complying with determined schedules. Currently used IS cannot ensure proper work coordination in this department. The study goal is to improve and computerize the process of the department labor organizing, creating a specialized information system enhanced with the business rules subsystem. The idea is that the system of the determined business rules would be helpful both for department administration and employees themselves in optimizing and organizing labor while evaluating present situation (in the employment rates and other factors).

The paper analyzes the business rules (BR) management systems (BRMS). It reviews the advantages and limitations of the system. It deals with the operating business rules of the use of information systems capabilities. BRMS Blaze Advisor was selected to create the business rules management subsystem of department's work planning information system. Architecture, Structured Rule Language (SRL) and components of BRMS were analyzed on purpose to reach our goals. Main methods of using BA advisor components (decision tables, decision trees, rule flows, "IF... THEN..." rules) are stated in this work. Processes of component management are presented in activity diagrams.

Methodology of BRMS Blaze Advisor integration with online information systems were created during the analysis, design, development, testing and documenting of the work planning information system. Also instructions for the business rules management user interface were created. It enables user to create and modify business rules. Employees of the organization get the ability to control business processes, if the systems are implemented using proposed methodology.

1. Įvadas

Organizacijos naudoja įvairias informacines ir programų sistemas, kurios yra jų veiklos dalis. IT sprendimai turi vis didesnę reikšmę organizacijos veiklai. Informacinių sistemų svarba išaugo labai stipriai.

Organizacijai svarbu, kad visos veiklos operacijos būtų atliekamos ne tik teisingai, bet ir numatytu laiku, arba griežtai apibrėžtu nuoseklumu, prisilaikant numatytos tvarkos, veiksmų scenarijaus, kur kiekvienam naudotojui griežtai apibrėžtos jo teisės ir galimybės atsižvelgiant į pareigas. Taip pat labai svarbu, kad visa valdymui reikalinga informacija būtų prieinama ne tik per lokalų kompiuterių tinklą, bet ir naudojantis internetu. Atsakingas darbuotojas, nesvarbu kur jis bebūtų – namuose, komandiruotėje, seminare ar kitur – visą reikalingą informaciją privalo turėti po ranka. Tam naudojamos įvairios darbo organizavimo sistemos.

Veikla vyksta nuolat besikeičiančioje aplinkoje, kartu su aplinka keičiasi pati veikla ir jos taisyklės. Pakeitimai taip pat turi atsispindėti informacinėse sistemose. Iškyla problema, kaip veiklos taisyklės, suformuluotas veiklos lygmenyje, kuo greičiau ir efektyviau perkelti į informacinę sistemą.

Tradicinės informacinės sistemos dažnai nebetenkina šiuolaikinių organizacijų poreikių. Kadangi šiose sistemose veiklos logika yra integruota į programinį kodą, jos negali greitai prisitaikyti prie pasikeitusių poreikių. Taigi vis dažniau kuriamos sistemos, kurios palaiko veiklos taisyklės atskirai nuo programinio kodo. Taip užtikrinamas greitesnis organizacijos veiklos valdymas. Taikomosios programos, tinkamos veiklos taisyklių technologijai taikyti, sutinkamos visur, kur egzistuoja poreikis greitai ir lengvai realizuoti pokyčius deklaratyviose organizacijos žiniuose.

Kauno technologijos universiteto informacijos sistemų katedroje darbuotojai vienu metu dirba keliuose projektuose, atlieka mokslinę veiklą, skaito paskaitas, dalyvauja konferencijose, seminaruose. Katedros darbuotojams yra svarbu, kad visi darbai būtų atliekami laiku, pagal nustatytą grafiką. Šiuo metu katedroje šiam tikslui naudojama IS negali užtikrinti tinkamo darbų koordinavimo. Tyrimo tikslas – kompiuterizuoti ir pagerinti katedros darbo organizavimo procesą, šiam tikslui sukuriant specializuotą, veiklos taisyklių posistemių papildytą, informacinę sistemą. Siekiama, kad sistema, praturtinta veiklos taisyklėmis, tiek katedros administracijai, tiek ir patiems darbuotojams padėtų optimaliai organizuoti darbą, įvertinus esamą situaciją (darbuotojų užimtumą ir kitas aplinkybes).

Šiam tikslui įgyvendinti iškelti uždaviniai:

1. Ištirti darbo organizavimo procesą Informacijos sistemų katedroje;
2. Ištirti veiklos taisyklių taikymo informacinėse sistemose galimybes;
3. Ištirti VT valdymo sistemos „Blaze Advisor“ pritaikymo iškelto tikslo realizacijoje galimybes;
4. Suprojektuoti ISK darbo organizavimo informacinę sistemą, papildytą veiklos taisyklių posistemiui;
5. Sukurti ir ištestuoti suprojektuotą darbo organizavimo sistemą;

Analizuojant veiklos taisyklių sąvoką bei veiklos taisyklų naudojimą informacinėse sistemose remtasi [1], [3], [4], [5], [6], [10], [11], [12] šaltiniais. VTVS sistemoms tirti buvo pasitelkti šaltiniai: [2], [7], [8], [9], [11], [16]. Kuriant metodiką ir vykdant sistemos realizaciją remtasi [2], [13], [14], [15] šaltiniais.

Darbe aptariamos veiklos taisyklių naudojimo informacinėse sistemose galimybės. Atliekant išsamią veiklos taisyklių (VT) valdymo sistemų analizę (VTVS), nustatyti šių sistemų privalumai ir trūkumai. Darbe nagrinėjama galimybė taikyti veiklos taisykles bei veiklos taisyklių valdymo sistemas plečiant sistemų funkcionalumą bei naudojant taisykles sprendimų priėmimui. Atlikta veiklos taisyklių analizė parodė, kad jas galima taikyti internetinių informacinių sistemų kūrime ir vartojime, tokiu būdu keičiant sistemų modulių vykdymo logiką.

Katedros darbo organizavimo informacinės sistemos veiklos taisyklių posistemiui kurti pasirinkta Blaze Advisor VTVS. Siekiant užsibrėžto tikslo, buvo išanalizuota VTVS (Blaze Advisor) architektūra, šios sistemos naudojama specifinė VT užrašymo kalba bei VT kūrimo komponentai.

Šiame darbe yra pateikiami penki skyriai:

2 skyriuje pateikiama veiklos taisyklės sąvoka, išnagrinėjami veiklos taisyklių tipai ir struktūros, analizuojami veiklos taisyklių valdymo įrankiai, jų architektūros ir funkcinės savybės. Taip pat aprašoma probleminės srities analizė, išskiriami vartotojų poreikiai ir jų problemos.

3 ir 4 skyriuose pateikiami dėstytojų darbo organizavimo informacinės sistemos reikalavimai ir projektas. Darbe aprašoma suprojektuota sistemos architektūra. Projektas sukurtas atsižvelgiant į išanalizuotoje literatūroje siūlomus sprendimus, vartotojų poreikius ir technines galimybes.

5 skyriuje aprašoma veiklos taisyklių integracijos į taikomųjų uždavinių sistemas metodika. Detaliai išanalizuojamas sistemų, praturtintų veiklos taisyklėmis, kūrimo procesas. Nurodomos gairės, kaip kuriama taisyklių saugykla, atliekama integracija su duomenų baze, kuriamos taisyklės („IF.. THEN..“ tipo taisyklės, sprendimo lentelės, sprendimo medžiai, taisyklių srautai). Išnagrinėjamas taisyklių projekto transformavimo į taisyklių paslaugas procesas. Metodikoje siūloma veiklos taisyklių valdymui naudoti šablonus. Šablonas - tai yra veiklos taisyklės teiginio bazinė struktūra ar kalbos išraiška, kuri naudojama taisyklei išreikšti [14]. Taip suteikiama galimybė organizacijos nariams, neturintiems techninių žinių, peržiūrėti, kurti, keisti, šalinti veiklos taisykles be įsikišimo į programinį kodą.

6 skyriuje apžvelgiama sukurta sistema: pagrindiniai vartotojo langai ir funkcijos, kurias galima atlikti sistemoje. Atlikta integracija su Blaze Advisor įrankiu padeda vartotojams valdyti veiklos taisykles. Tai suteikia galimybę priiminėti efektyvesnius sprendimus, vykdant katedros veiklą.

Atlikto darbo pagrindu suformuluoti tokie pagrindiniai teiginiai:

- Veiklos taisyklių sluoksnio atskyrimas informacinėje sistemoje suteikia galimybę vartotojams, neturintiems techninių žinių, peržiūrėti, kurti, keisti, šalinti veiklos taisykles (t.y. valdyti kompiuterizuotą veiklos logiką) be įsikišimo į programinį kodą;
- Veiklos taisyklėmis grindžiama sistema pasižymi didesniu lankstumu, plečiamumu ir pakartotiniu panaudojimu;
- Sistemos realizacija parodė, kad pasiūlytą metodiką tikslinga taikyti, kuriant veiklos taisyklėmis grindžiamas informacines sistemas.

2. Informacijos sistemų katedros veiklos ir informacinių poreikių analizė

2.1. Analizės tikslas

Analizės tikslas – išsiaiškinti dėstytojo darbo organizavimo informacinės sistemos atliekamas funkcijas, išanalizuoti katedroje vykstančius veiklos procesus ir veiklos dalyvių poreikius. Taip pat siekiama palyginti įvairias veiklos taisyklių valdymo sistemas.

Pagal atliktą analizę vėliau bus modeliuojami įvairūs sistemos aspektai, projektuojami sistemos komponentai.

2.2. Tyrimo sritis, objektas ir problema

Veiklos taisyklių taikymas informacinėse sistemose.

Veiklos taisyklė (VT) – tai:

- teiginys, kuris apibrėžia arba sąlygoja tam tikrą veiklos aspektą. VT skirta apibrėžti veiklos struktūrai arba valdyti, įtakoti veiklos elgseną;
- atominis, deklaratyviai specifiкуotas, daugkartinio panaudojimo veiklos logikos fragmentas.

Tyrimo objektas – informacijos sistemų katedros dėstytojų darbo organizavimo informacinė sistema.

Šiuolaikiniame nuolatos besikeičiančiame pasaulyje organizacijoms ir įmonėms vis sunkiau efektyviai ir harmoningai organizuoti savo kasdieninę veiklą.

Nagrinėjamoje organizacijoje, tai yra informacijos sistemų katedroje, darbuotojai vienu metu dirba keliuose projektuose, atlieka mokslinę veiklą, veda paskaitas, dalyvauja konferencijose, seminaruose. Katedros administracijai, o tuo pačiu ir kitiems katedros darbuotojams, yra svarbu, kad visi darbai būtų atliekami laiku, pagal nustatytą grafiką.

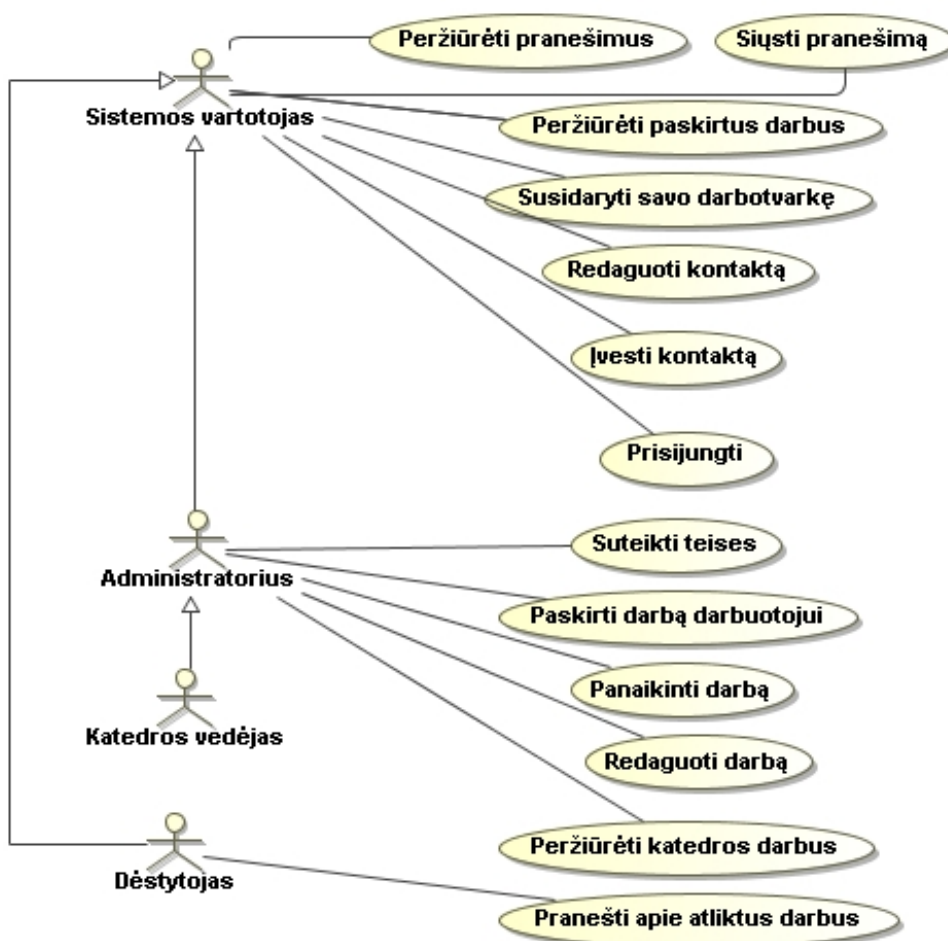
Šiuo metu katedroje šiam tikslui naudojama IS negali užtikrinti tinkamo darbų koordinavimo.

Darbų paskirstymo ir koordinavimo uždaviniams spręsti kuriamą informacinę sistemą ketinama praturtinti veiklos taisyklių posistemiu. Tikimasi, kad sukurtoji sistema tiek katedros administracijai, tiek ir patiems darbuotojams padės optimaliai organizuoti ISK darbą, įvertinus esamą situaciją (darbuotojų užimtumą ir kitas aplinkybes).

2.3. Tyrimo objekto analizė

Išanalizavus katedros naudojamą dėstytojo darbo organizavimo sistemą sudaromas panaudojimo atvejų modelis (1 pav.). Modelyje matome sistemos atliekamas funkcijas.

Kiekvieno panaudojimo atvejo aprašymas pateikiamas 1 lentelėje:



1 pav. Naudojamos sistemos panaudojimo atvejų modelis

1 lentelė panaudojimo atvejų aprašymai

PANAUDOJIMO ATVEJIS	PANAUDOJIMO ATVEJO APRAŠYMAS
Peržiūrėti pranešimus	Peržiūrimi gauti ir išsiųsti pranešimai
Siųsti pranešimą	PA apima pranešimų rašymą ir siuntimą
Prisijungti	Šis PA apima vartotojo prisijungimą prie sistemos. Pagal vartotojų kategorijas, vartotojai gali atlikti tik tam tikrus veiksmus.

Suteikti teises	Šis PA apima teisių suteikimą ir redagavimą.
Paskirti darbą darbuotojui	Kiekvienam katedros dėstytojui pateikiami darbai, kuriuos jis turi atlikti.
Panaikinti darbą darbuotojui	Panaikinamas darbas.
Redaguoti darbą darbuotojui	Redaguojamas darbas: keičiamas atlikimo laikas, darbuotojas, kuris atliks darbą, redaguojama darbo svarba.
Peržiūrėti katedros darbus	Stebimi katedros dėstytojų atlikti ir neatlikti darbai.
Susidaryti savo darbotvarę	Šis PA apima darbų laikų paskyrimą, darbų prioritetų nustatymą.
Pranešti apie atliktus darbus	Informuojamas katedros vedėjas apie atliktą darbą
Peržiūrėti paskirtus darbus	Pateikiama informacija apie paskirtus darbus

2.4. Tyrimo tikslas ir uždaviniai

Tyrimo tikslas – kompiuterizuoti ir pagerinti katedros darbo organizavimo procesą, šiam tikslui sukuriant specializuotą, veiklos taisyklių posistemiu papildytą, informacinę sistemą.

Tyrimo uždaviniai:

6. Ištirti darbo organizavimo procesą Informacijos sistemų katedroje;
7. Ištirti veiklos taisyklių taikymo informacinėse sistemose galimybes;
8. Ištirti VT valdymo sistemos „Blaze Advisor“ pritaikymo iškelto tikslo realizacijoje galimybes;
9. Suprojektuoti ISK darbo organizavimo informacinę sistemą, papildytą veiklos taisyklių posistemiu;
10. Sukurti ir ištestuoti suprojektuotąją darbo organizavimo sistemą;

2.5. Analizės metodai

Organizacijos analizei naudojama UML modeliavimo kalba.

Analizėje bus apžvelgtos panašios egzistuojančios sistemos tam, kad galima būtų panaudoti jų gerąsias savybes, atitinkančias keliamus reikalavimus.

2.6. Vartotojų analizė

2.6.1. Vartotojų aibė, tipai ir savybės

Kaip matome 1 pav. Sistemos vartotojai yra tokie:

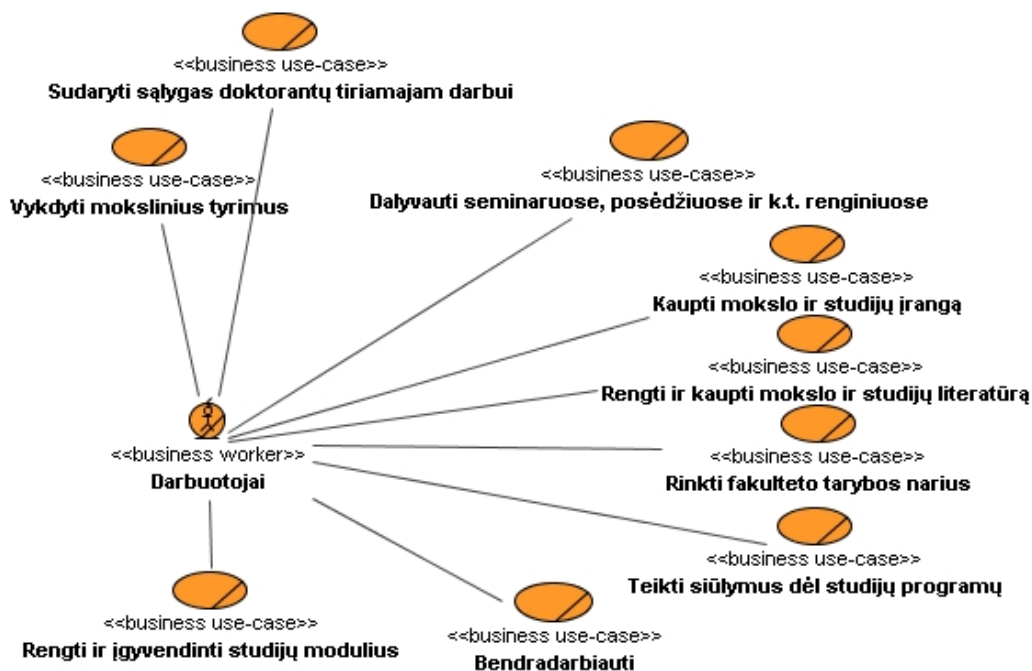
- Administratorius
- Katedros vedėjas
- Dėstytojas

2.6.2. Vartotojų tikslai ir problemos

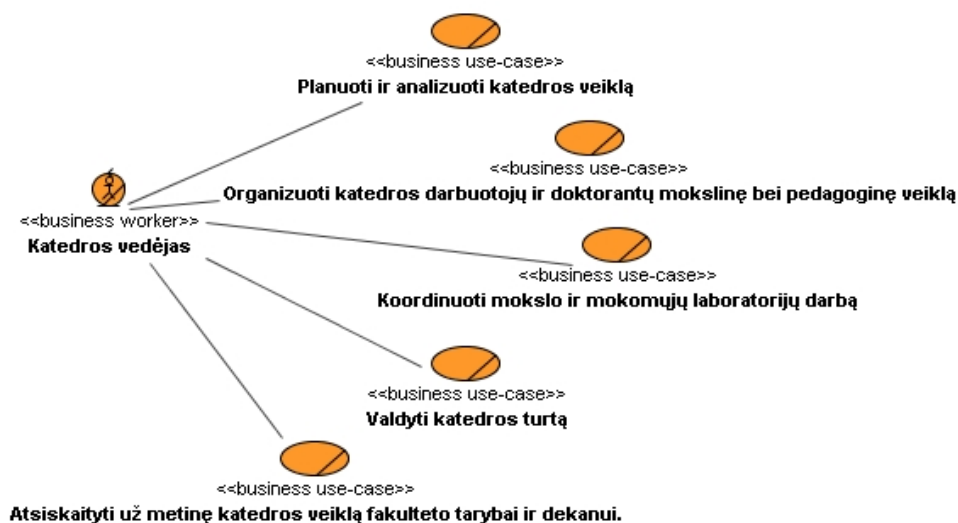
Katedros darbuotojų veiklos pavaizduotos panaudojimo atvejų diagrama (2pav.).

Iš panaudojimo atveju matome, kad darbuotojai ir vedėjas atlieka daug įvairių veiklų. Organizuojant katedros darbą yra svarbu suderinti visus darbus. Kuriama sistema turi užtikrinti tinkama katedros darbų koordinavimą, pasiremiant veiklos taisyklių panaudojimu.

Valdymo ir administravimo procese sistema leistų pasinaudoti tikslia ir laiku gauta informacija. Tai palengvintų valdymo sprendimų priėmimą ir efektyviai atliktų planavimo, koordinavimo, bendradarbiavimo ir kontrolės funkcijas.



2 pav. Darbuotojų veiklos panaudojimo atvejų modelis



3 pav. Katedros vedėjo veiklos panaudojimo atvejai

2.7. Veiklos taisyklių analizė

Veiklos taisyklė (VT) – tai:

- teiginys, kuris apibrėžia arba sąlygoja tam tikrą veiklos aspektą. VT skirta apibrėžti veiklos struktūrai arba valdyti, įtakoti veiklos elgseną;
- atominis, deklaratyviai specifikuotas, daugkartinio panaudojimo veiklos logikos fragmentas.

Veiklos taisyklės yra :

- deklaratyvios (neprocedūrinės);
- nedalomos, tačiau išsamios;
- išreikštos natūralia kalba;
- atskiros nepriklausomos konstrukcijos;
- orientuotos į DS, o ne IS kūrimo technologiją;
- „išplaukiančios“ iš DS, o ne IS kūrimo technologijos.

Veiklos taisyklių panaudojimas duoda šiuos privalumus:

- paprastesnis pakeitimų įvedimas;
- techninė nepriklausomybė;
- greitesnis programinis įrangos kūrimas;
- geresnis reikalavimų išpildymas;
- didesnis lankstumas.

VT sluoksnio atskyrimas nuo duomenų saugojimo ir jų vaizdavimo sluoksnių yra centrinė idėja trijų (ar daugiau) lygių loginėje kliento/serverio IS programinės įrangos architektūroje. Veiklos taisyklių atskyrimas suteikia lankstumo. Atskirai valdomas VT sluoksnis prireikus gali būti nesunkiai perkeltas į kitą techninę platformą, siekiant padidinti taikomosios sistemos efektyvumą.

VT sakiny – tai deklaratyvus sakiny, nusakantis veiklai nustatytą struktūrą arba apribojimą. Kiekvienas VT sakiny gali būti susijęs su vienu ar daugiau kitų VT sakinių.

VT sakiny vartojamas užrašant vieną ar daugiau nedalomų veiklos taisyklių. Veiklos taisyklė apibrėžia arba apriboja tam tikrą veiklos aspektą, tačiau ji negali būti toliau skaidoma į detalesnes VT. Tolesnis dekomponavimas lemtų svarbios veiklą liečiančios informacijos praradimą.

Formaliame taisyklės sakinyje vartojama specifinė formali gramatika. Formalus taisyklės sakiny privalo būti užrašytas pagal formalaus reiškinio tipo sudarymo taisykles.

Veiklos taisyklių kategorijos[5]:

- struktūrinis teiginys (structural assertion) – apibrėžta mintis arba fakto konstatavimas, išreiškiantis tam tikrą organizacijos struktūros aspektą;
- veiksmo teiginys (action assertion) – apribojimą ar sąlygą reiškiantis sakiny, kuris nustato arba valdo organizacijos veiksmus;
- išvestis (derivation) – sakiny, išreiškiantis žinias, kurios savo ruožtu išvedamos iš kitų veiklos žinių.

Struktūriniai teiginiai nusako veiklai svarbius dalykus, kurie egzistuoja kaip konceptualios esybės arba yra susiję su kitais reikšmingais dalykais. Struktūriniu teiginiu apibrėžiamas konkretus statinis veiklos aspektas, nusakantis žinomus dalykus arba kaip žinomi dalykai siejasi tarpusavyje. Struktūriniai teiginiai dažnai vaizduojami esybių/ryšių (entity/relationship) modeliuose.

Veiksmo teiginys – tai sakiny, kuris aprašo tam tikrą veiklos dinaminį aspektą. Veiksmo teiginys nustato apribojimus rezultatams, kurie gaunami atlikus veiksmus. Šie apribojimai aprašomi neprocedūriniu būdu, kaip ir kitos nedalomos veiklos taisyklės. Veiksmo teiginio veiklos taisyklės skirstomos į klases:

- sąlygas (condition);
- apribojimus (constraint);
- leidimus (authorization).

Išvestys (derivations) – tai veiklos taisyklės, kurios panaudojamos gaunant išvestinius faktus. Pirminiai faktai yra visuotinai suprantami ir įsimenami (išsaugomi) informacinėje sistemoje, o išvestiniai faktai gaunami loginiu išvedimu ar matematinium apskaičiavimu iš sistemoje saugomų sąvokų, faktų, išvesčių ar net veiksmo teiginių.

Išvestys skirstomos į:

- matematinius skaičiavimus (mathematical calculation);
- logines išvadas (inference).

Matematinis skaičiavimas gauna išvestinį faktą panaudodamas apibrėžtą matematinį algoritmą. Loginė išvada gauna išvestinį faktą panaudodama loginę indukciją arba dedukciją.

2.8. Esamų sprendimų analizė

Veiklos taisyklės (VT) yra būdas struktūrizuoti veiklos žinias. Veiklos taisyklėms nuolat keičiantis veiklos lygmenyje iškyla poreikis išlaikyti visus pakeitimus ir informacinės sistemos (IS) lygmenyje.[1]

Taikomosios programos, tinkamos veiklos taisyklių technologijai taikyti, sutinkamos visur, kur egzistuoja poreikis greitai ir lengvai realizuoti pokyčius deklaratyviose organizacijos žiniuose [11]. Pasaulyje yra keletas organizacijų, kurios užsiima sistemų, pagrįstų veiklos taisyklėmis, realizacija.

Buvo išnagrinėtos tokios veiklos taisyklių valdymo sistemos:

- Quick rules
- Infrex
- Versata Logic Suite
- ILOG Rules
- Blaze Advisor

2.8.1. Quick rules

Quick Rules yra pagrindinis bendrovės YASU Technologies produktas[9]. Quick rules naudoja savo verslo taisyklių sukurtą užrašymo kalbą – Quick rules Taisyklių Žymėjimo Kalbą (Quick rules Rules Markup Language (QRML)). Taisyklės yra saugomos XML formate.

Funkcinės savybės:

Versijų kontrolė: Versijų kontrolė padeda sekti taisyklių istoriją, koks vartotojas, kada, kurią taisyklę ir kaip pakeitė. Galima sulygtinti vykstančias tuo pačiu metu taisykles.

Veiklos naudotojui draugiška veiklos taisyklių kalba: Quick Rules veiklos taisykles leidžia įrašinėti supaprastinta anglų kalba. Tai padaryta tam, kad veiklos naudotojai suprastu, keistu ir valdytu veiklos taisykles.

EJB palaikymas: Quick Rules yra specialiai sukurtas lengvesniam integravimui į J2EE palaikanti taikomąjį serverį. Palaikomi tokie serveriai: BEA Weblogic, WSAD, IBM WebSphere, SunONE Application Server, Oracle 9i AS, Orion, Borland Enterprise Server, JBoss.

Konfliktų sprendimas: Sudėtinių veiklos taisyklių vykdyme gali atsirasti konfliktų, kuriuos reikia išspręsti.

Priėjimo kontrolė veiklos naudotojams: Naudojant šią funkcinę savybę galima kontroliuoti vartotojų priėjimą prie veiklos taisyklių. Quick Rules leidžia apribuoti arba suteikti teisę kurti, modifikuoti, trinti arba paleidinėti taisykles.

Sprendimų lentelių palaikymas: Sprendimų lentelės naudojamos konstruoti ir atnaujinti sąlygas ir veiksmus veiklos taisyklėse. Tipinis tokių sprendimų lentelių pavyzdys – lentelė naudojama kainų ir valiutų kursų skaičiavimui.

Įvykių detalizavimo API: Quick Rules įvykių detalizavimo API naudojamas įvykių, įvykusių taisyklių vykdymo metu analizei. Tokia informacija naudojama taisyklių derinime.

Veiklos taisyklių apibrėžimas naudojant XML: Quick Rules leidžia laikyti manipuluoti informacija XML dokumentuose.

2.8.2. Infrex

Inflex - tai įrankis, skirtas komercinėse taikomose programose nuolat besikeičiančioms veiklos taisyklėms specifikuoti ir interpretuoti.[8] Naudojant Inflex, taikomosios programos vartotojai gali valdyti pokyčius jų veiklos srityje, įvedant naujas ar pakeičiant senas taisykles, naudojant paprastą taisyklių kalbą (rule language).

Inflex suprojektuotas ne kaip savarankiškas, bet kaip įterptinis taisyklių interpretatorius ir yra integruojamas į taikomąsias programas, kurios parašytos

C/C++/Java/C# kalbomis. Infrex pateikia aibę produktų, kad patenkintų įvairius vartotojų poreikius.

Taisyklės gali būti apibrėžtos deklaratyvia IF-THEN forma kartu su prioritetu, pradžios data, pabaigos data ir versijos numeriais arba procedūrine IF-THEN-ELSE/ELSEIF forma. Taisyklių kalba turi tokius operatorius, kaip ir anglų kalba: INCREASE, DECREASE, SET, EVALUATE, IS, IN ir t.t. Palaikomi veiklos taisyklių šablonai ir leidžiama bendras sąlygas išvardinti kaip šabloną. Taisyklės taip pat gali būti apibrėžiamos kaip sprendimų medis, sprendimų lentelė ar orientuotas grafas[8].

Funkcinės savybės

Grupavimo mechanizmas: taisyklių aibės gali būti sustatytos į eilę ir sukomplektuotos įvairiais būdais. Taisyklės sekoje gali būti detalai įvertintos, kad tik viena konkreti taisyklė būtų vykdoma arba perspėjama, jei įvykdomos kelios taisyklės. Taisyklių eilės formavimas gali būti pradedamas nuo bet kurio taško.

Taisyklių sekų ir pritaikomumo parinkimas vykdymo metu: Taisyklių sekos ir pritaikomumas gali būti nusprendžiami dinamiškai. Tai labai naudinga, kai į paprastą taisyklių bazę reikia kreiptis skirtingais būdais vykdymo metu. Tokiu būdu pasiekiamas didelis taikomosios sistemos lankstumas.

Sudėtinės taisyklės, taisyklių versijos ir atgalinės datos transakcijos: Taisyklės gali būti sudėtinės ir remtis kitomis taisyklėmis. Vieną kartą įvertintos taisyklės gali būti testuojamos ar pakartotinai įvertinamos. Palaikomos daugelio versijų taisyklės ir galima nustatyti atitinkamos versijos taisyklės prioritetą. Galima nustatyti pradžios ir pabaigos datą kiekvienai taisyklei. Tai padeda vykdant atgalinės datos transakcijas.

Bendras duomenų naudojimas tarp taikomųjų programų ir taisyklių: Taisyklės dirba su taikomųjų programų klasėmis ir kintamaisiais. Šie duomenys perduodami veiklos taisyklių interpretatoriui be dubliavimo ar kopijavimo. Klasės ir kintamieji yra pasiekiami Infrex sistemos, todėl jiems gali būti apibrėžiamos veiklos taisyklės. Be to, gali būti apibrėžiami virtualūs kintamieji, kurie gauna reikšmes iš taikomosiuose programose apibrėžtų funkcijų, duomenų bazės ar grafinės vartotojo sąsajos.

Trasavimas, derinimas ir naršymas: Įvertintų taisyklių registravimas leidžia trasuoti įvykdytas taisykles. Taisyklių derinimo mechanizmas leidžia suderinti taisyklių bazę. Taisyklių bazė gali būti išsaugoma kaip HTML failas ir peržiūrėta per interneto naršyklę. Aprašytos veiklos taisyklės gali būti transliuojamos į C/C++/Java/C# kodą. Šis kodas yra kompiliuojamas ir jungiamas su taikomąja programa tam, kad sukurti vykdomąją programą.

2.8.3. Versata Logic Suite

Versata Logic Suite [16] yra verslo taisyklių kūrimo ir diegimo įrankiai, kurių komponentės atlieką verslo taisyklių automatizavimą bei vykdomo kodo generavimą informacinei sistemai iš deklaratyvių taisyklių teiginių.

Veiklos taisyklių kūrimo aplinka susideda iš dviejų įrankių: Versata Transaction Logic Designer ir Versata Process Logic Designer.

Funkcinės savybės

Transakcijų logiką apima apdorojimą prieš arba duomenų pakeitimo metu. Tai apima tokias funkcijas kaip apribotas įvedimas, duomenų reikšmių kalkuliavimas, atributų gavimą ir atnaujinimą, iškvietimą išorinių įvykių kai transakcijos atmetamos arba įvykdytos.

Procesų logiką koordinuoją sistemų ir vartotojų veiklą, ji apima tokias funkcijas kaip transakcijų ir uždavinių sekų nustatymas, progreso sekimas, išimčių nustatymas.

2.8.4. ILOG Rules

Veiklos taisyklių valdymo sistemos ILOG Rules (C++) ir ILOG Jrules (Java) yra produktai kompanijos ILOG [7]. ILOG Rules (C++) naudoja C++ klasių bibliotekos, o ILOG Jrules (Java) – Java klasių bibliotekos. Sistema gali valdyti iki 50000 taisyklių.

ILOG taisyklės yra ECA tipo taisyklės, ir turi tokią pat struktūra kaip ir ECA taisyklės. ILOG taisyklė sudaryta iš trijų dalių: antraštės, sąlygos ir veiksmo. Bendras taisyklės užrašymo formatas yra toks:

```
rule ruleName { [ priority = value; ]  
[ packet = packetName; ]  
when { conditions ... }  
then { [ actions ... ] }  
};
```

Taisyklės antraštė aprašo taisyklės pavadinimą, prioritetą ir paketo pavadinimą. Sąlyga prasideda raktiniu žodžiu when ir po jo užrašomos sąlygos atskirtos kabliataškiais. Veiksmo dalys prasideda raktiniu žodžiu then, ir po jo aprašomas veiksmas, kuris yra vykdomas, kada visos sąlygos yra išpildytos.

Taisyklės prioritetas kontroliuoja taisyklių paleidimo eilės tvarką. Paketo pavadinimas suriša taisyklę su taisyklių paketu. Naudojant paketus susietos taisyklės yra grupuojamos.

ILOG Rules priemonėje verslo taisyklės galima aprašinėti ir kitais būdais: naudojant verslo veiksmų kalbą - BAL (Business Action Language), intuityvus IF-THEN-ELSE taisyklės formatas ir TRL (Technical Rule Language), be to naudojamas ir “sprendimų lentelės formatas.”

Funkcinės savybės

Versijų kontrolė: Versijų kontrolė padeda sekti taisyklių istoriją, koks vartotojas, kada, kurią taisyklę ir kaip pakeitė. Galima sulygtinti tuo pačiu metu vykdomas taisykles.

Veiklos naudotojui draugiška veiklos taisyklių kalba: ILOG Rules veiklos taisyklės gali įrašinėti keliais būdais, kaip ir programuotojui skirta kalba, taip ir veiklos naudotojui suprantama kalba. Tai leidžia veiklos naudotojams valdyti veiklos procesus.

ILOG Rules API: ILOG Rules API leidžia nesunkiai kontroliuoti veiklos taisyklių vykdymą, ir integruoti jas į C++ ir Java taikomas programas. Tai padaryti galima COM, COBRA arba IBM MQSeries technologijų pagalba.

Taisyklių derinimo programa: Naudojama įvykių, vykstančių taisyklių vykdymo metu kontrolei. Yra kontroliuojamas taisyklių vykdymo procesas. Be to galima inspektuoti objektų, su kuriais taisyklė yra susijusi, būseną, peržiūrėti visų įvykių istoriją.

2.8.5. Blaze Advisor

Blaze Advisor – veiklos taisyklių valdymo sistema, kurios pagalba galima kurti, diegti ir prižiūrėti į verslo taisyklės orientuotas sistemas. Lengvai suprantama verslo taisyklių aprašymo kalba leidžia ne techniniam verslo personalui kurti, valdyti bei prižiūrėti verslo taisyklės.

Blaze Advisor yra Java sprendimas, kuris palaiko Enterprise Java Beans (EJB) ir Java2 Enterprise Edition (J2EE) platformos, Microsoft Transaction Server (MTS) ir Microsoft Component Object Model (COM), IBM mainfreimus.

Pagrindinis naudojamas verslo taisyklių atvaizdavimo metodas – struktūrinė taisyklių kalba (Structured Rule Language SRL). Dažnai galima naudoti taisyklių šablonus, tokiu

atveju taisyklių rašymas tampa laibiau intuityvus, ir taip sukurtos taisyklės yra lengviau skaitomos.

Daugelis realaus pasaulio sistemų turi didelį kiekį taisyklių, kurios atitinka tam tikrus fiksuotus šablonus. Pavyzdžiui, vartotojas gali nustatyti skirtingas sąlygas, kurios gali būti pagrįstos norimais kriterijais. Viena taisyklė gali būti tokia:

- Jei pirkėjas yra iš Vilniaus, tada pasiūlyti paslaugą nemokamai.

Kita taisyklė:

- Jei pirkėjo amžius > 70 , tada pasiūlyti 3 paslaugas nemokamai.

Iš pirmo žvilgsnio atrodo, kad šios taisyklės turi mažai bendro, tačiau detaliau išanalizavus matome, kad abi taisyklės atitinka tą patį šabloną:

Jei pirkėjas \diamond savybė \diamond palyginimas \diamond reikšmė

Tada pasiūlyti nemokamai \diamond skaičius \diamond prekė

Blaze Advisor leidžia sukurti tokį šabloną. Jo pagalba galima greitai ir be klaidų rašyti veiklos taisykles. Tai pagreitina panašių taisyklių kūrimą bei suteikia nuoseklumą jų apibrėžimui.

Specifinis veiklos taisyklių atvejis, kai taisyklės atitinka konkretų šabloną, yra kuomet taisyklės struktūra yra nekintanti, o keičiasi tik tikrinamos reikšmės bei rezultatai. Blaze Advisor suteikia galimybę sukurti lentelę, kuri aprašo tokio tipo sąlygas ir rezultatus nerašant programinio kodo.

Lentelės duomenys gali būti importuojami iš Excel'io lentelių, reikšmės keičiamos pačioje lentelėje ir lentelės gali būti formatuojamos daugeliu būdų, leidžiant sukurti daug stulpelių/eilučių turinčias lenteles. Patikrinimo rezultatai gali būti sudėtingi taisyklių veiksmai ar tiesiog paprastos gražinamos reikšmės. Sprendimų lentelės tai vienas iš būdų, kuriais Blaze Advisor suteikia aiškumą ir sumažina kūrimo laiką bei pastangas specifikuojant veiklos logiką.

Blaze Advisor turi galimybes, leidžiančias supaprastinti dinamišką interaktyvių taikomųjų programų informacijos atnaujinimą interneto puslapiuose. Taisyklės gali apibrėžti, kokia informacija turėtų būti įtraukiama į interneto puslapius, įtraukiant papildomus klausimus, kurie turėtų būti užduodami, atsižvelgiant į ankstesnius atsakymus. Kaip ir anksčiau, technologiniai sprendimai, nusakantys kaip fiziškai atvaizduoti interneto puslapius, yra palikti išorinėms sistemoms, todėl yra palaikomas sprendimų atskyrimas nuo infrastruktūros kodo. Lengvai apibrėžiamos ir palaikomos sąsajos su išorinėmis sistemomis teikia greitį ir efektyvumą kūrimo ir vykdymo metu.

Funkcinės savybės

Taisyklių versijų kontrolė ir valdymo autorizacija: vartotojų priėjimo kontrolė prie veiklos taisyklių valdymo funkcijų.

Taisyklių valdymas: Ne techninis personalas gali peržiūrėti, modifikuoti, trinti arba kurti taisyklės nenaudojant programavimo.

Operacinis suderinamumas su išoriniais duomenų šaltiniais: XML, Java/J2EE objektų, COBRA & COM objektų palaikymas.

Sudėtingų taisyklių kūrimas: Galimybė naudoti daugiažingsnes šakojančias taisykles.

Sprendimų lentelių palaikymas: Sprendimų lentelės naudojamos konstruoti ir atnaujinti sąlygas ir veiksmus veiklos taisyklėse. Tipinis tokių sprendimų lentelių pavyzdys – lentelė naudojama kainų ir valiutų kursų skaičiavimui.

Taisyklių testavimo, derinimo priemonės: taisyklių vykdymo siekimas naudojant testinius kintamuosius. Ryšių tarp taisyklių, parametrų ir objektų nustatymas.

Taisyklių grupavimas: taisyklės atskiriamos pagal grupes, tam kad jas būtų lengviau valdyti ir kontroliuoti.

2. Taisyklių valdymo sistemų palyginimų lentelė

	Quick rules	Infrex	Versata Logic Suite	ILOG Rules	Blaze Advisor
Taisyklių aprašymas	Quick rules Rules Markup Language	Deklaratyvi forma, sprendimų lentelės, orientuoti grafai	Deklaratyvi forma	Sprendimų lentelės	Taisyklių kalba, sprendimų lentelės
Platformos	BEA Weblogic, WSAD, IBM WebSphere, SunONE Application Server, Oracle 9i AS, Orion, Borland Entrprise Server, JBoss.	Unix (Linux, HP-UX, AIX), SunSolaris, IBMOS/390, Windows95/NT/2000/XP/.NET	Microsoft MTS/COM+, IBM MQ	COM, COBRA arba IBM MQSeries	J2EE, Microsoft MTS/COM+, IBM MQ
Sprendimų medžiai	-	+	-	-	+
Šablonai	-	+	+	+	+
Versijų kontrolė	+	+	-	+	+

Atlikus veiklos taisyklių valdymo sistemų analizę, nuspręsta darbo organizavimo sistemą papildyti Blaze Advisor taisyklių valdymo sistema. Kaip matome iš lentelės Blaze Advisor turi daugiau galimybių. Tai yra šis produktas palaiko sprendimų lenteles, sprendimų medžius. Taip pat gali naudoti šablonus.

2.9. Siekiamas sprendimas

Šiuo metu katedroje naudojama IS negali užtikrinti tinkamo darbų koordinavimo.

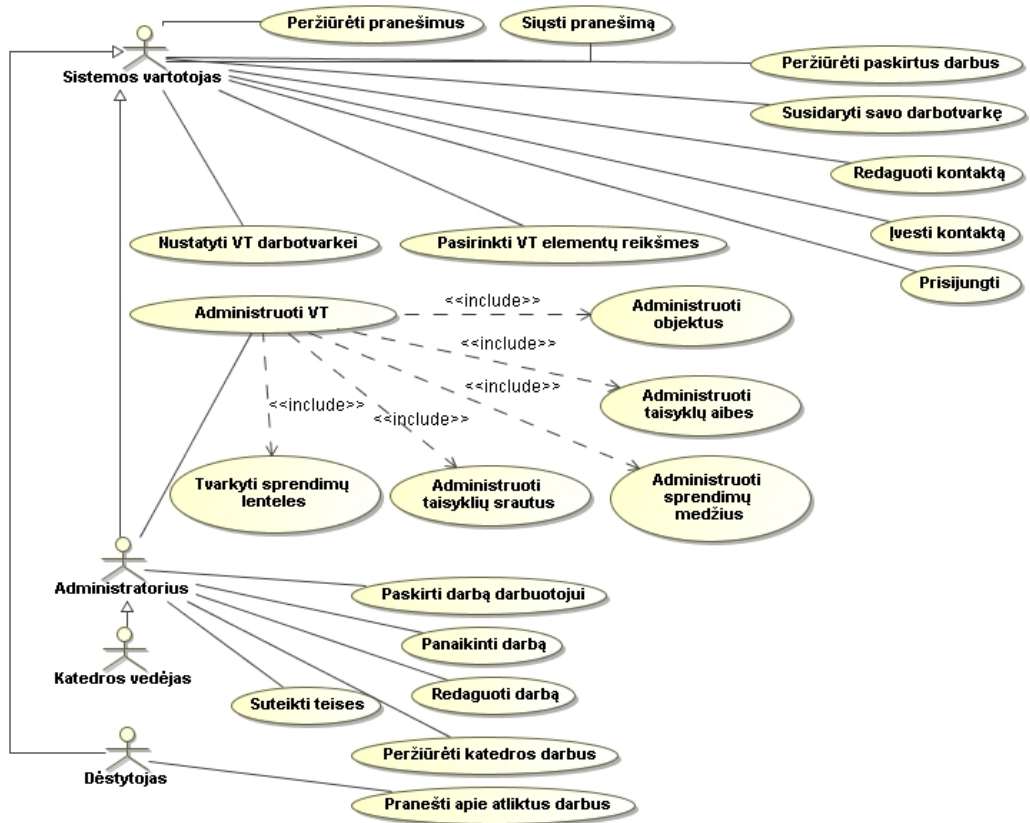
Darbų paskirstymo ir koordinavimo uždaviniams spręsti kuriamą darbo organizavimo informacinę sistemą ketinama praturtinti veiklos taisyklių posistemiu. Papildžius sistemą Blaze Advisor įrankiu, vartotojai galės priiminėti efektyvesnius sprendimus.

Patobulintos sistemos panaudojimo modelis pavaizduotas 4 pav. Sistema pasipildo keliomis naujomis funkcijomis.

Funkcija „Administruoti VT šablonus“ leidžia administratoriui sukurti ir koreguoti veiklos taisyklių šablonus, kuriuos naudos vartotojai. Tam naudojamos sprendimų lentelės, taisyklių srautai, taisyklių aibės, duomenų objektai, sprendimų medžiai. Šiems elementams administruoti skirti panaudojimo atvejai: „Tvarkyti sprendimų lenteles“, „Administruoti taisyklių srautus“, „Administruoti sprendimų medžius“, „Administruoti taisyklių aibes“, „Administruoti objektus“.

Funkcija „Pasirinkti VT elementų reikšmes“ suteikia vartotojui galimybę nusistatyti veiklos taisyklių parametrus pačiam. Apsirašyti apribojimus ir tam tikrus veiksmus, kurie bus atliekami, atsitikus aprašytam įvykiui.

Naudojant funkciją „Nustatyti VT darbotvarkei“ darbuotojai galės naudoti VT sudarant savo darbotvarkę, tvarkant priminimus. Tai taip pat padės katedros vedėjui efektyviau skirstyti užduotis. Darbai bus paskiriami pagal aprašytas veiklos taisykles. Tai padės išvengti per didelio darbuotojų darbo krūvio.

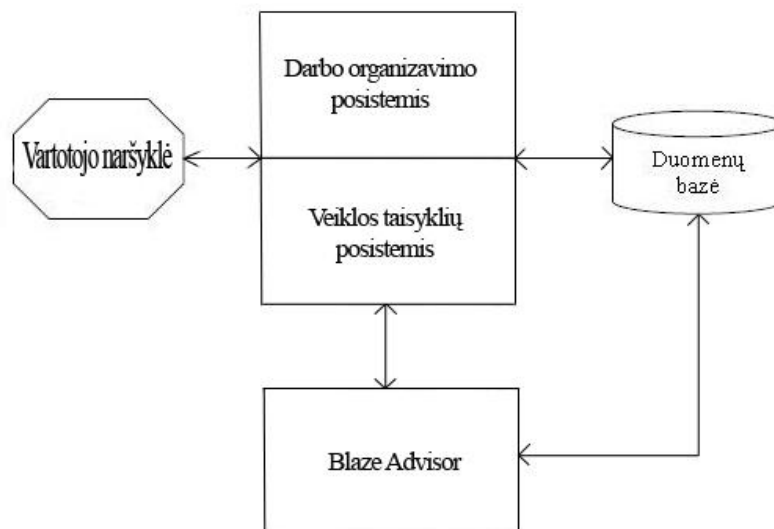


4 pav. Sistemos su veiklos taisyklių posistemių panaudojimo atvejų modelis

Bendra siekiamo sprendimo architektūra pavaizduota 6 pav. Sistemą sudarys du posistemiai: „Darbo organizavimo“ ir „Veiklos taisyklių“, kurie tarpusavyje bendradarbiaus.

Jei reikalingi duomenys taikomųjų programų serveris kreipiasi į duomenų bazę. Duomenų bazėje taip pat saugomi duomenys apie veiklos taisykles.

Vartotojas sistemą valdys naršyklės pagalba.



5 pav. Sistemos architektūra

Blaze Advisor pateikia įrankius, kurie leidžia parodyti specifines veiklos taisyklių dalis specialaus techninio išsilavinimo neturintiems vartotojams, taip, kad taisyklės gali būti pakeičiamos kontroliuojamu bei saugiu būdu be jokio programavimo mokymo. Sukūrimas, pakeitimas ir taisyklių ištrynimasis gali būti atliekamas per interneto puslapius. Blaze Advisor be viso to turi galimybę sekėti ir registruoti visus pokyčius, kurie buvo padaryti sistemos logikai.

Blaze Advisor susideda iš keturių pagrindinių komponentų:

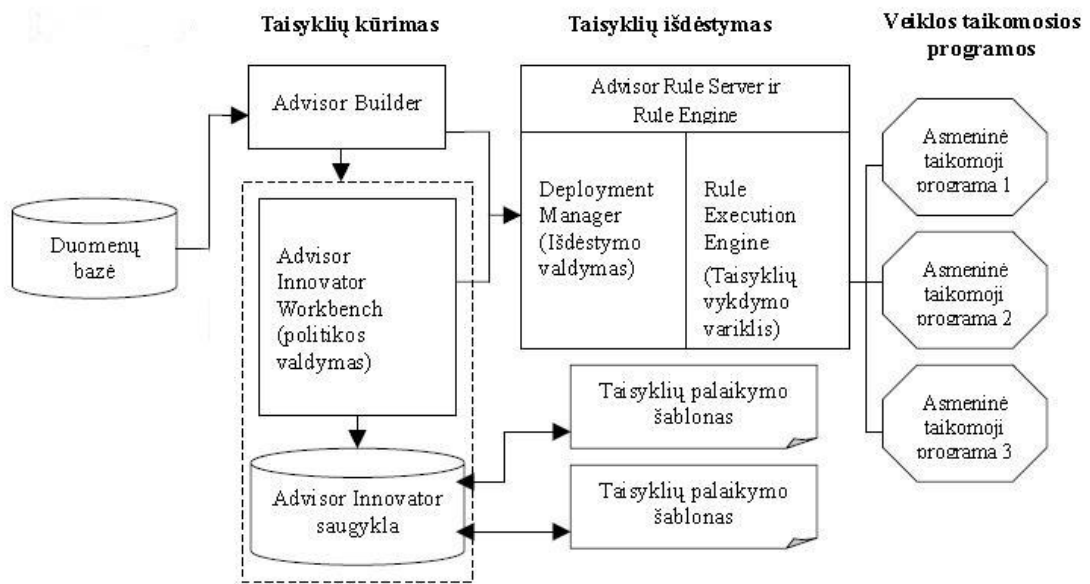
1. **Blaze Advisor Builder** – aplinka, skirta lengvam veiklos taisyklių kūrimui. Ji yra tiesiogiai orientuota į vartotojus/kūrėjus ir aprūpina žemo lygio programavimo bei grafine aplinka, skirta taisyklių rašymui, redagavimui, peržiūrai ir testavimui.

2. **Blaze Advisor Innovator** – taisyklių valdymo ir palaikymo įrankis, skirtas vartotojams. Vartotojai kuria ir modifikuoja taisykles centralizuotai apibrėžtų meniu aibės pagalba. Darbastalis aprūpina internetinėmis sąsajomis, skirtomis taisyklių struktūrizavimui remiantis veiklos politika, tokiomis kaip leidžiamos reikšmės ar apribojimai. XML pagrindu sukurta saugykla leidžia išsaugoti visus taisyklės komponentus ir pokyčius. Taisyklių šaltiniai saugomi ir indeksuojami standartinės struktūros faile.

3. **Blaze Advisor Rule Engine** – išplečiamas veiklos taisyklių vykdymo interpretatorius, kuris apibrėžia ir vykdo visus veiklos taisyklių valdymo srautus, kurie buvo sukurti su Blaze Advisor Builder. Interpretatorius veikia kartu su Blaze Advisor Rule Server tam, kad įvykdytų tas taisykles, kurių reikia taikomosioms programoms. Taisyklių apdorojimas gali būti iššaukiamas ir išorinių įvykių.

4. **Blaze Advisor Rule Server** – veiklos taisyklių serveris, skirtas visai organizacijai. Serveris palaiko lygiagrečių taisyklių vykdymą, sesijos valdymą, planavimą ir dinamišką darbo krūvio valdymą. Deployment Manager komponentas tvarko visus kreipinius į duomenų bazę.

6 pav. pavaizduota Blaze Advisor architektūra:



6 pav. Blaze Advisor architektūra

2.10. Analizės išvados

1. Atlikus organizacijos veiklos ir turimos sistemos atliekamų funkcijų analizes, galime teigti, kad sistema netenkina visų organizacijos poreikių ir reikia sistemą patobulinti, tam panaudojant veiklos taisyklių valdymo sistemą.
2. Atlikus veiklos taisyklių valdymo sistemų analizę, buvo pastebėta, kad sistemų funkcionalumas yra labai panašus, skiriasi tik VT aprašymo būdai, bei platformos, su kuriomis gali bendradarbiauti šie įrankiai.
3. Išanalizavus visas veiklos taisyklių valdymo sistemų savybes nuspręsta, darbą tęsti su Blaze Advisor paketu, dėl jo plataus funkcionalumo ir galimybių dirbti su internetinėmis sistemomis.

3. Dėstytojų darbo organizavimo informacinės sistemos reikalavimai

3.1. Reikalavimų specifikacija

3.1.1. Panaudojimo atvejų specifikacija

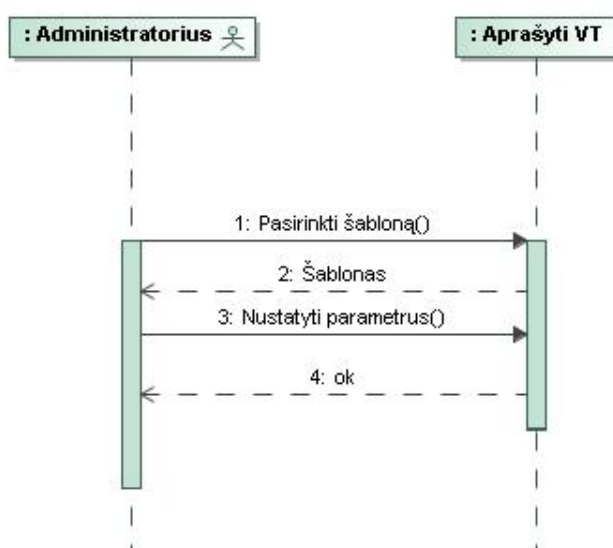
Analizės dalyje buvo nustatyti IS reikalavimai (panaudojimo atvejų diagrama 4 pav.). Siekiant aiškumo, pateikiama kiekvieno panaudojimo atvejo specifikacijos lentelė. (3- 21 lentelės.)

Sistemoje turi būti kompiuterizuotos tokios funkcijos:

- Aprašyti veiklos taisykles (7 pav.)
- Nustatyti veiklos taisykles darbotvarkei (8pav.)
- Administruoti veiklos taisykles(9 pav.)
- Peržiūrėti pranešimus (10 pav.)
- Siųsti pranešimą (11 pav.)
- Prisijungti (12 pav.)
- Suteikti teises (13 pav.)
- Paskirti darbą darbuotojui (14 pav.)
- Panaikinti darbą darbuotojui (15 pav.)
- Redaguoti darbą darbuotojui (16 pav.)
- Peržiūrėti katedros darbus (17 pav.)
- Susidaryti savo darbotvarkę (18 pav.)
- Pranešti apie atliktus darbus (19 pav.)
- Peržiūrėti paskirtus darbus (20 pav.)
- Administruoti objektus (21 pav.)
- Administruoti taisyklių aibes (22 pav.)
- Administruoti sprendimų medžius (23 pav.)
- Administruoti taisyklių srautus (24 pav.)
- Tvarkyti sprendimų lenteles (25 pav.)

3 lentelė Panaudojimo atvejo „Aprašyti veiklos taisykles“ specifikacija

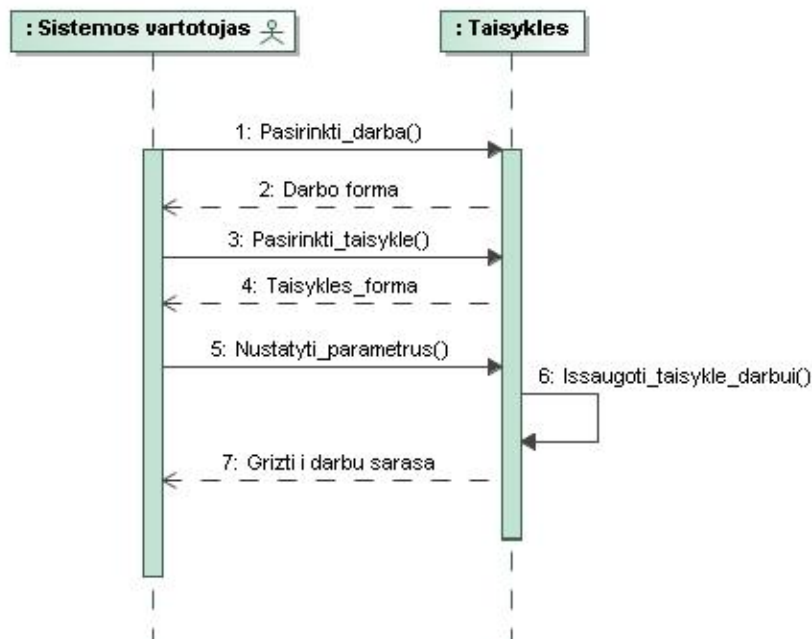
PA „Aprašyti veiklos taisykles“	
Tikslas. Aprašyti veiklos taisykles, naudojant darbų koordinavimui	
Aprašymas. Šis PA apima veiklos taisyklių valdymą, kūrimą, redagavimą, šalinimą	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Sistemos vartotojas
Sužadinimo sąlyga	Vartotojui reikia veiklos taisyklių
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka reikalingą VT šabloną	Sistema atidaro šablono formą
2. Vartotojas nustato reikalingus parametrus	Sistema išsaugo veiklos taisyklę
3. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.
2. Jei vartotojas redaguoja VT	Sistema išsaugo pakeistus VT parametrus
3. Jei vartotojas šalina VT	Sistema išsaugo pakeitimus



7 pav. Panaudojimo atvejo „Aprašyti veiklos taisykles“ sekos diagrama

4 lentelė Panaudojimo atvejo „Nustatyti veiklos taisykles darbotvarkei“ specifikacija

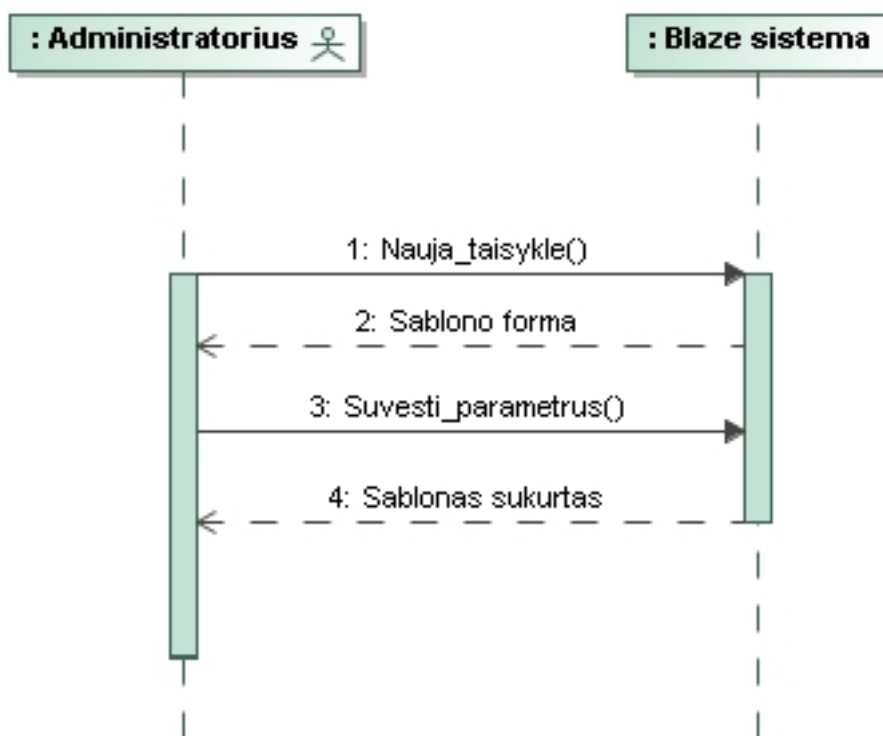
PA „Nustatyti veiklos taisykles darbotvarkei“	
Tikslas. Priskirti veiklos taisykles darbams, užduotims, darbuotojų grupėms.	
Aprašymas. Šis PA tvarko veiklos taisykles ir darbų koordinavimą.	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Sistemos vartotojas
Sužadinimo sąlyga	Vartotojui reikia priskirti darbui veiklos taisyklę
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka reikalingą darbą	Sistema atidaro darbo redagavimo formą
2. Vartotojas pasirenka, kokio tipo taisyklę nori priskirti	Sistema atidaro taisyklės formą
3. Vartotojas nustato reikalingus parametrus	Sistema išsaugo taisyklę darbui
4. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.



8 pav. Panaudojimo atvejo „Nustatyti veiklos taisykles darbotvarkei“ sekos diagrama

5 lentelė Panaudojimo atvejo „Administruoti veiklos taisykles“ specifikacija

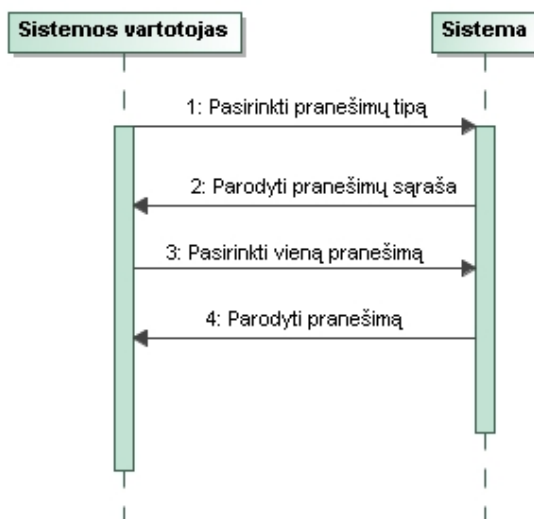
PA „Administruoti veiklos taisykles“	
Tikslas. Tvarkyti veiklos taisykles	
Aprašymas. Šis PA sukuria ir redaguoja veiklos taisykles.	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Sistemos administratorius
Sužadinimo sąlyga	Vartotojui reikia tvarkyti veiklos taisykles
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka, kokio tipo šabloną nori sukurti.	Sistema pateikia atitinkamus langus
2. Vartotojas sukuria naują taisyklę	Sistema atidaro taisyklės redagavimo formą
3. Vartotojas nustato reikalingus parametrus	Sistema išsaugo taisyklę
4. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.



9 pav. Panaudojimo atvejo „Administruoti veiklos taisykles“ sekos diagrama

6 lentelė Panaudojimo atvejo „Peržiūrėti pranešimus“ specifikacija

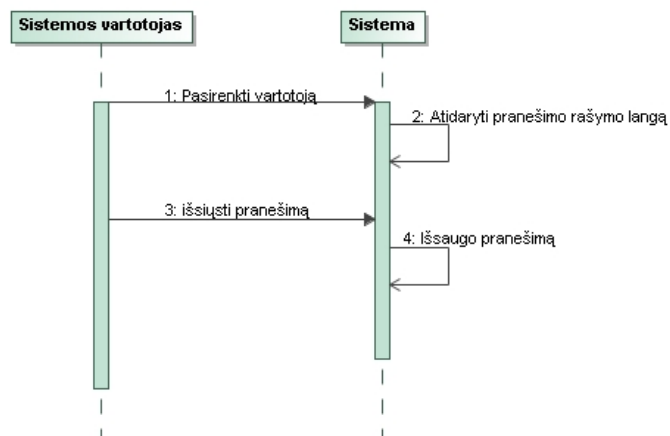
PA „Peržiūrėti pranešimus“	
Tikslas. Peržiūrėti gautus pranešimus	
Aprašymas. Šis PA apima išsiųstų ir gautų pranešimų peržiūrėjimą	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Sistemos vartotojas
Sužadinimo sąlyga	Vartotojas nori peržiūrėti pranešimus
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka „išsiųsti“ arba „gauti“ pranešimai	Sistema atidaro pasirinktą sritį
2. Vartotojas pasirenka vieną pranešimą	Sistema atidaro pasirinktą pranešimą.
3. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.



10 pav. Panaudojimo atvejo „Peržiūrėti pranešimus“ sekos diagrama

7 lentelė Panaudojimo atvejo „Siųsti pranešimą“ specifikacija

PA „Siųsti pranešimą“	
Tikslas. Siųsti pranešimus	
Aprašymas. Šis PA apima pranešimų rašymą ir siuntimą	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Sistemos vartotojas
Sužadinimo sąlyga	Vartotojas nori siųsti pranešimą
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka vartotoją, kuriam siųs pranešimą	Sistema atidaro langą į kurį vartotojas gali rašyti tekstą.
2. Vartotojas išsiunčia pranešimą	Sistema išsaugo išsiųstą pranešimą.
3. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.
2. Jei vartotojas atsisako siųsti pranešimą	Sistema grįžta į gautų pranešimų langą.

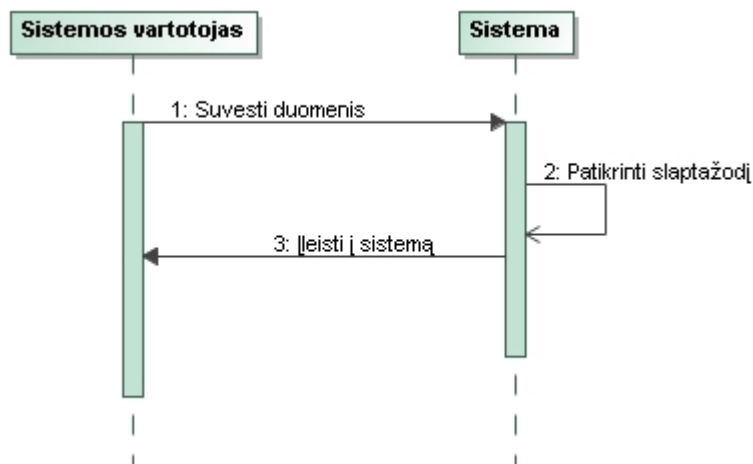


11 pav. Panaudojimo atvejo „Siųsti pranešimą“ sekos diagrama

8 lentelė Panaudojimo atvejo „Prisijungti“ specifikacija

PA „Prisijungti“	
Tikslas. Vartotojų prisijungimas prie sistemos	
Aprašymas. Šis PA apima vartotojo prisijungimą prie sistemos	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos

Aktorius	Sistemos vartotojas	
Sužadavimo sąlyga	Vartotojas nori naudotis sistema	
Susiję panaudojimo atvejai	Išplečia PA	
	Apima PA	Tikrinti teises
	Specializuoja PA	
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai	
1. Vartotojas suveda savo duomenis	Sistema patikrina ar prisijungimo vardas ir slaptažodis atitinka ir suteikia teises.	
2. Sistema baigia PA		
Alternatyvūs scenarijai		
1. Jei vartotojas įveda neteisingą slaptažodį	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.	

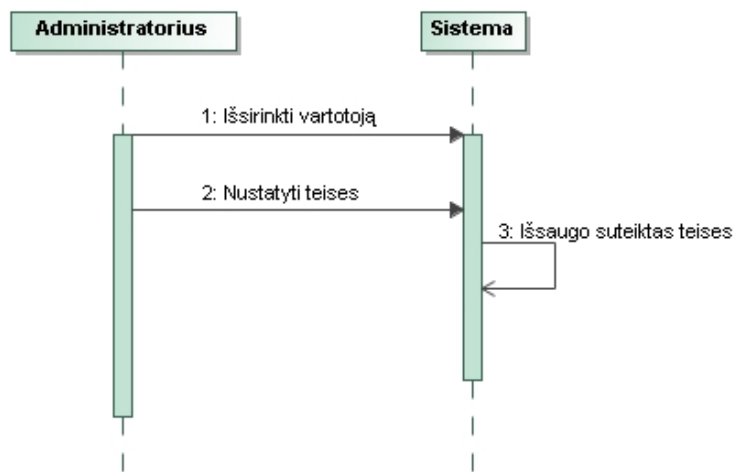


12 pav. Panaudojimo atvejo „Prisijungti“ sekos diagrama

9 lentelė panaudojimo atvejo „Suteikti teises“ specifikacija

PA „Suteikti teises“	
Tikslas. Pagal vartotojų kategorijas, vartotojai gali atlikti tik tam tikrus veiksmus.	
Aprašymas. Šis PA apima teisių suteikimą ir redagavimą.	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Sistemos administratorius
Sužadavimo sąlyga	Administratorius nori nustatyti teises
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius išsirenka vartotoją	Sistema atidaro langą, kuriame administratorius

	nustato teises
2. Administratorius patvirtina pakeitimus	Sistema išsaugo suteiktas teises
3. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei administratorius neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.

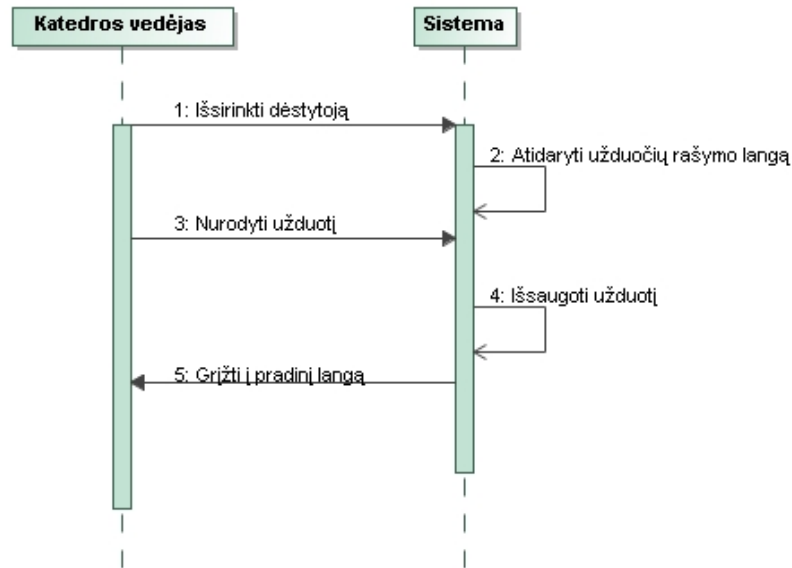


13 pav. Panaudojimo atvejo „Suteikti teises“ sekos diagrama

10 lentelė Panaudojimo atvejo „Paskirti darbą darbuotojui“ specifikacija

PA „Paskirti darbą darbuotojui“	
Tikslas. Kiekvienam katedros dėstytojui pateikti darbus, kuriuos jis turi atlikti	
Aprašymas. Šis PA apima darbų paskirstymą	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Katedros vedėjas
Sužadinimo sąlyga	Katedros vedėjas nori paskirstyti darbus
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vedėjas išsirenka dėstytoją	Sistema atidaro langą, kuriame vedėjas gali įrašyti užduotį.
2. Vedėjas nurodo užduotį ir ją patvirtina	Sistema išsaugo patvirtintą užduotį ir grįžta į pradinį langą.
3. Sistema baigia PA	

Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.
2. Jei vedėjas atsisako skirti užduotį	Sistema grįžta į pradinį langą

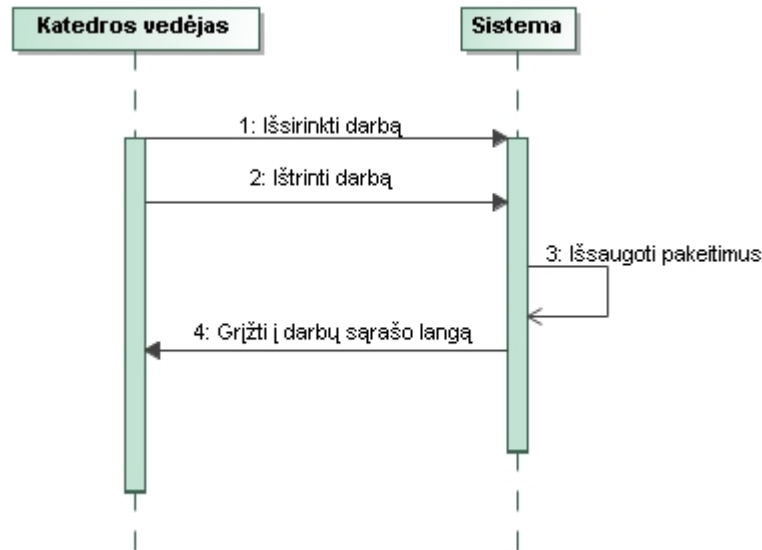


14 pav. Panaudojimo atvejo „Paskirti darbą darbuotojui“ sekos diagrama

11 lentelė Panaudojimo atvejo „Panaikinti darbą darbuotojui“ specifikacija

PA „Panaikinti darbą darbuotojui“	
Tikslas. Pasikeitus aplinkybėms panaikinti darbą.	
Aprašymas. Šis PA apima darbų panaikinimą	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Katedros vedėjas
Sužadinimo sąlyga	Katedros vedėjas nori panaikinti darbą
Pagrindinis įvykių srautas	
1. Vedėjas išsirenka darbų sąrašą	Sistema atidaro langą, kuriame vedėjas gali matyti visus darbus.
2. Vedėjas nurodo užduotį ir ją panaikina	Sistema išsaugo pakeitimus ir grįžta į pradinį langą.
3. Sistema baigia PA	

Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.
2. Jei vedėjas atsisako naikinti užduotį	Sistema grįžta į darbų sąrašo langą

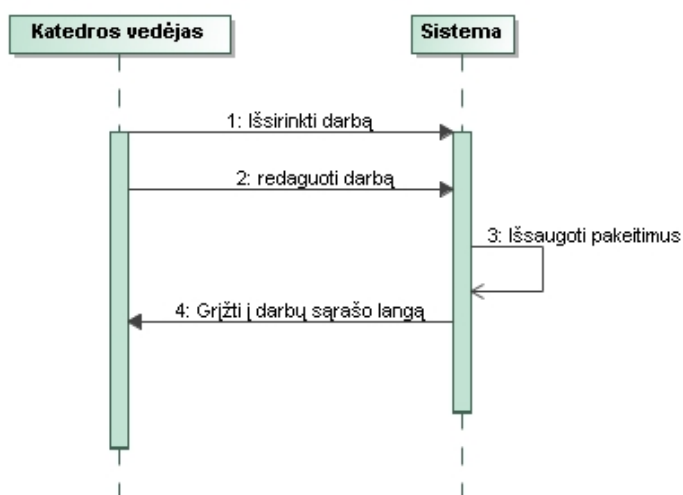


15 pav. Panaudojimo atvejo „Panaikinti darbą darbuotojui“ sekos diagrama

12 lentelė Panaudojimo atvejo „Redaguoti darbą darbuotojui“ specifikacija

PA „Redaguoti darbą darbuotojui“	
Tikslas. Pasikeitus aplinkybėms redaguoti darbą.	
Aprašymas. Šis PA apima darbų redagavimą	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Katedros vedėjas
Sužadinimo sąlyga	Katedros vedėjas nori redaguoti darbą
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vedėjas išsirenka darbų sąrašą	Sistema atidaro langą, kuriame vedėjas gali matyti visus darbus.
2. Vedėjas nurodo užduotį ir ją poredaguoja	Sistema išsaugo pakeitimus ir grįžta į pradinį langą.
3. Sistema baigia PA	

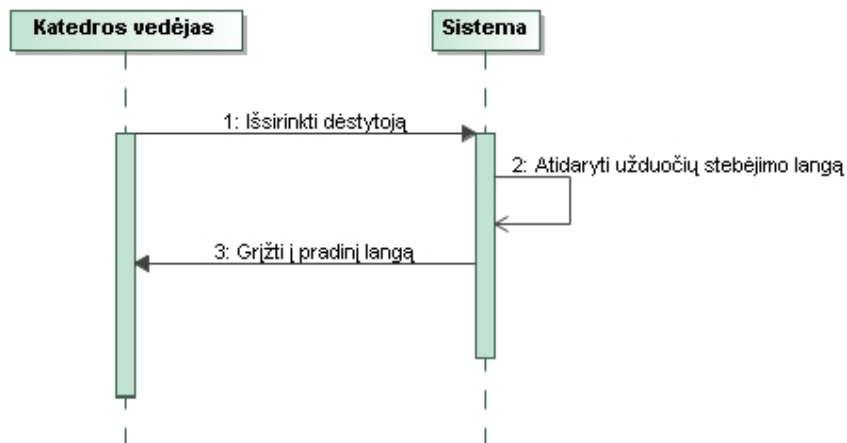
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.
2. Jei vedėjas atsisako redaguoti užduotį	Sistema grįžta į darbų sąrašo langą



16 pav. Panaudojimo atvejo „Redaguoti darbą“ sekos diagrama

13 lentelė Panaudojimo atvejo „Peržiūrėti katedros darbus“ specifikacija

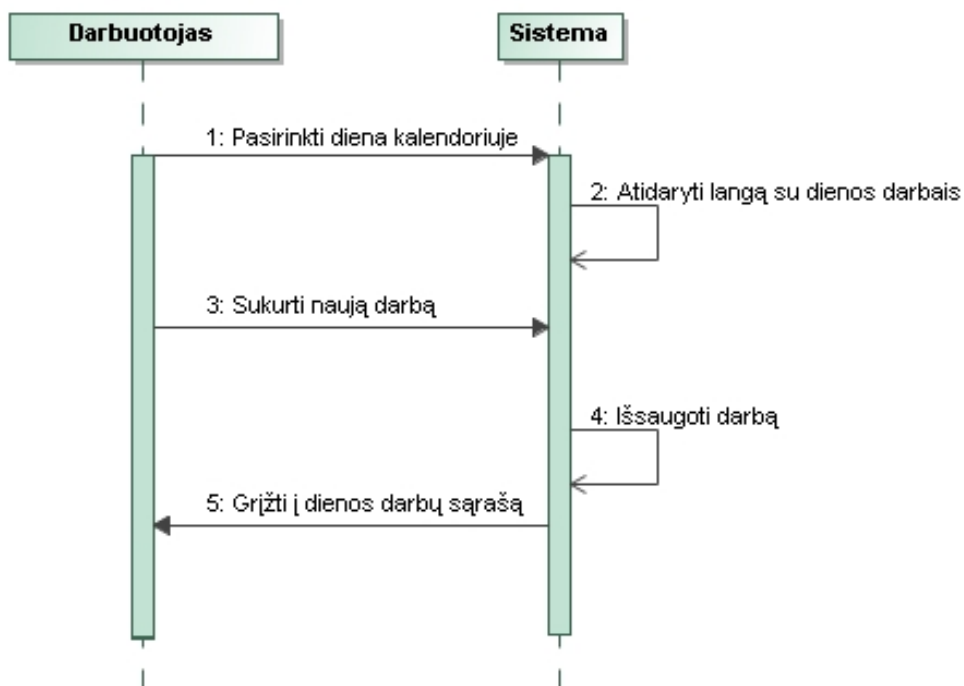
PA „Peržiūrėti katedros darbus“	
Tikslas. Stebėti katedros dėstytojų atliktus ir neatliktus darbus	
Aprašymas. Šis PA apima darbų stebėjimą	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Katedros vedėjas
Sužadinimo sąlyga	Katedros vedėjas nori peržiūrėti darbus
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vedėjas išsirenka dėstytoją	Sistema atidaro langą, kuriame vedėjas gali matyti dėstytojui paskirtas užduotis.
2. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.
2. Jei vedėjas pasirenka tam tikrą laikotarpį	Sistema filtruoja tik tuos darbus, kurie priklauso pasirinktam laikotarpiui.
3. Jei vedėjas pasirenka prioritetą	Sistema filtruoja tik tuos darbus, kurie atitinka pasirinktą prioritetą.



17 pav. Panaudojimo atvejo „Peržiūrėti katedros darbus“ sekos diagrama

14 lentelė Panaudojimo atvejo „Susidaryti savo darbotvarkę“ specifikacija

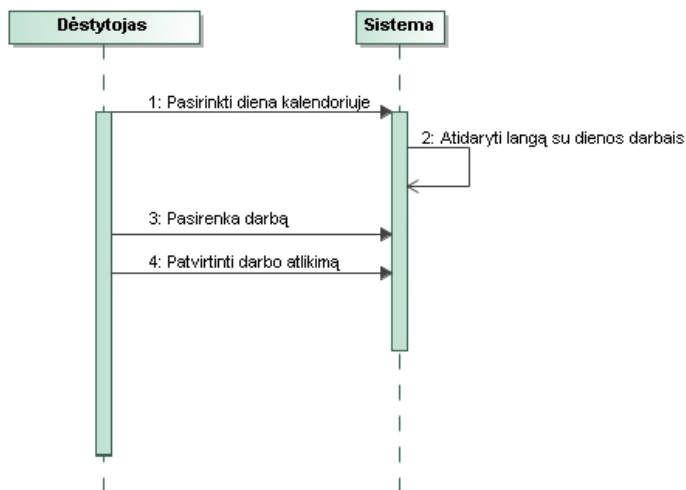
PA „Susidaryti savo darbotvarkę“	
Tikslas. Atvaizduoti darbotvarkės sudarymą	
Aprašymas. Šis PA apima darbų laikų paskirimą, darbų prioritetų nustatymą	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Katedros vedėjas, dėstytojas
Sužadinimo sąlyga	Darbotvarkės įjungimas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Darbuotojas pasirenka kalendoriuje dieną	Sistema atidaro langą, kuriame darbuotojas gali matyti tos dienos darbus.
2. Darbuotojas sukuria naują darbą	Sistema atidaro langą, kuriame vartotojas gali suvesti duomenis apie darbą (laikas, vieta, komentaras)
3. Darbuotojas patvirtina darbo duomenis	Sistema grįžta į pagrindinį darbotvarkės langą.
4. Darbuotojas koreguoja darbą	Sistema atidaro langą, kuriame vartotojas gali keisti duomenis apie darbą (laikas, vieta, komentaras)
5. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.
2. Jei darbuotojas pasirenka darbą ir pasižymi jį kaip atliktą.	Sistema neberodo darbo darbotvarkėje.



18 pav. Panaudojimo atvejo „Susidaryti savo darbotvarkę“ sekos diagrama

15 lentelė Panaudojimo atvejo „Pranešti apie atliktus darbus“ specifikacija

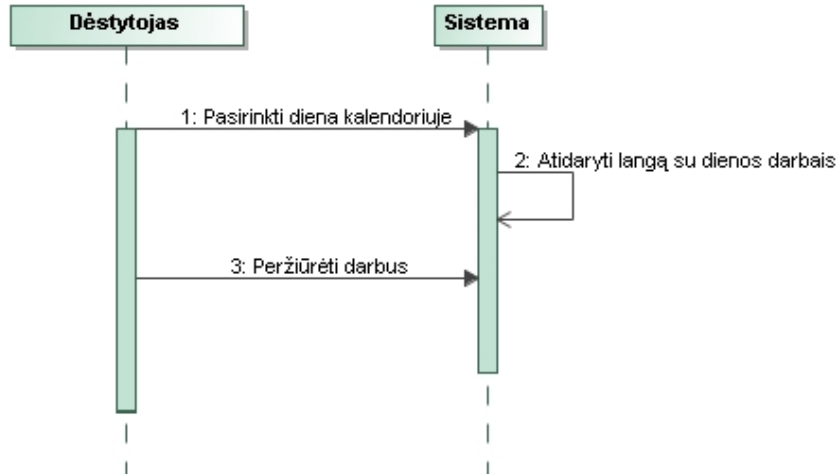
PA „Pranešti apie atliktus darbus“	
Tikslas. Informuoti katedros vedėją apie atliktus darbus	
Aprašymas. Šis PA apima informacijos pateikimą apie atliktus darbus	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Dėstytojas
Sužadinimo sąlyga	Darbotvarkės įjungimas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Darbuotojas pasirenka kalendoriuje dieną	Sistema atidaro langą, kuriame darbuotojas gali matyti tos dienos darbus.
2. Darbuotojas pasirenka darbą	Sistema atidaro langą, kuriame vartotojas gali suvesti komentarą apie darbą
3. Darbuotojas patvirtina darbo atlikimą	Sistema grįžta į pagrindinį darbotvarkės langą.
4. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.



19 pav. Panaudojimo atvejo „Pranešti apie atliktus darbus“ sekos diagrama

16 lentelė Panaudojimo atvejo „Peržiūrėti paskirtus darbus“ specifikacija

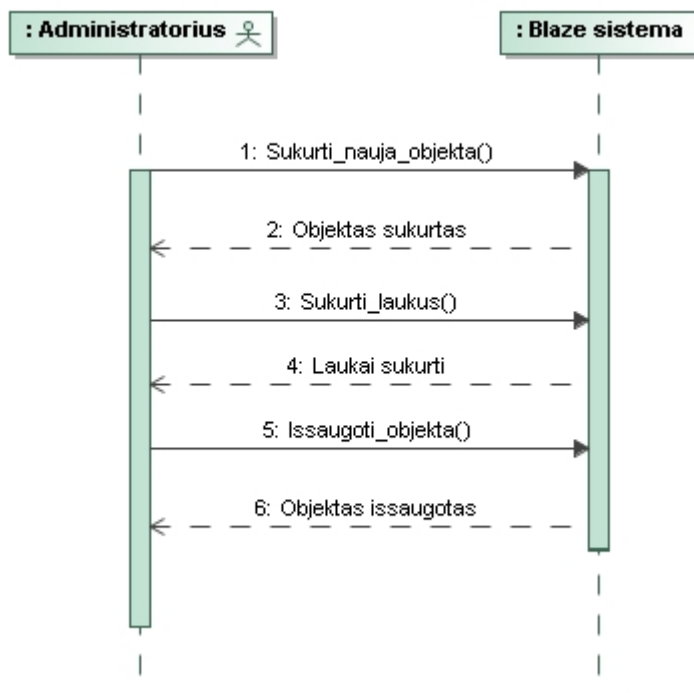
PA „Peržiūrėti paskirtus darbus“	
Tikslas. Peržiūrėti paskirtus darbus	
Aprašymas. Šis PA apima informacijos pateikimą apie paskirtus darbus	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Dėstytojas
Sužadinimo sąlyga	Darbotvarkės įjungimas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Darbuotojas pasirenka kalendoriuje dieną	Sistema atidaro langą, kuriame darbuotojas gali matyti tos dienos darbus.
2. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.



20 pav. Panaudojimo atvejo „Peržiūrėti paskirtus darbus“ sekos diagrama

17 lentelė Panaudojimo atvejo „Administruoti objektus“ specifikacija

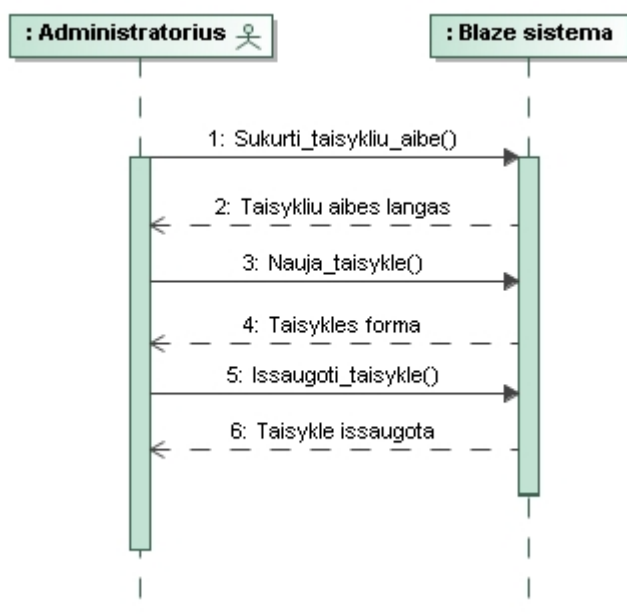
PA „Administruoti objektus“	
Tikslas. Administruoti objektus	
Aprašymas. Šis PA apima objektų sukūrimą ir jų administravimą.	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Administratorius
Sužadinimo sąlyga	Projekto pasirinkimas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius sukuria naują objektą	Sistema sukuria objektą ir atidaro langą, kuriame administratorius gali matyti objekto savybes.
2. Administratorius sukuria objekto laukus	Sistema sukuria objekto laukus
3. Administratorius išsaugo objektą	Sistema išsaugo objektą ir jo savybes
4. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.



21 pav. Panaudojimo atvejo „Administruoti objektus“ sekos diagrama

18 lentelė Panaudojimo atvejo „Administruoti taisyklių aibes“ specifikacija

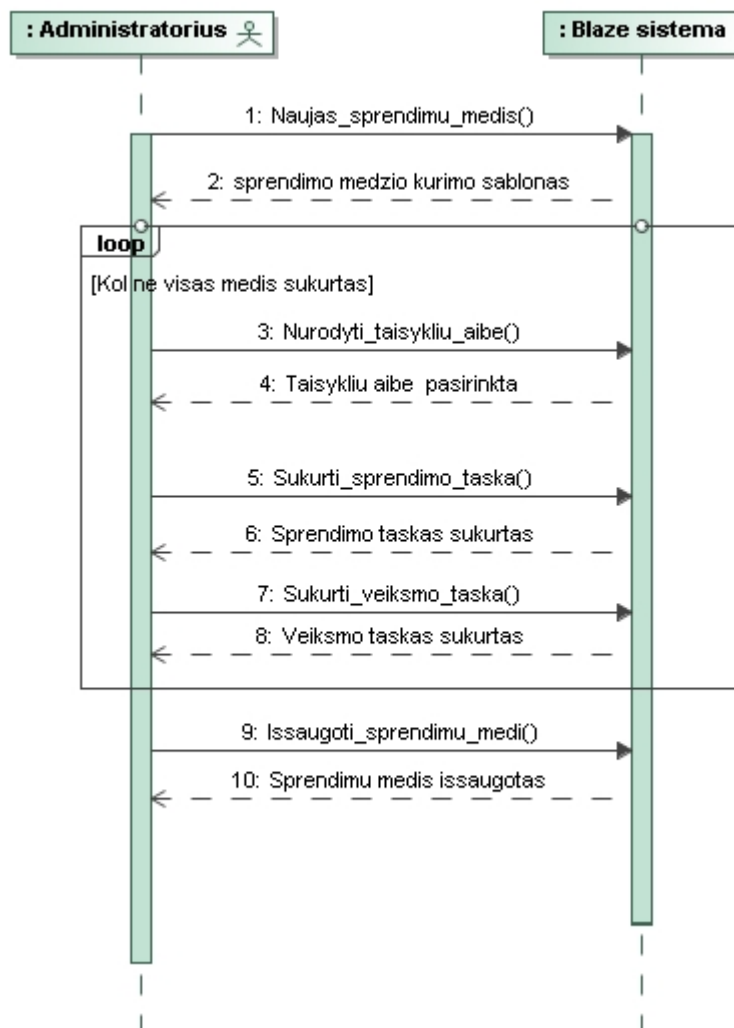
PA „Administruoti taisyklių aibes“	
Tikslas. Administruoti taisyklių aibes	
Aprašymas. Šis PA apima taisyklių aibės sukūrimą ir administravimą.	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Administratorius
Sužadinimo sąlyga	Projekto pasirinkimas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius sukuria naują taisyklių aibę	Sistema sukuria taisyklių aibę ir atidaro langą, kuriame administratorius gali matyti aibės savybes.
2. Administratorius sukuria naują taisyklę	Sistema atidaro taisyklės formą
3. Administratorius suveda taisyklės aprašymą ir išsaugo taisyklę	Sistema išsaugo taisyklę
4. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.



22 pav. Panaudojimo atvejo „Administruoti taisyklių aibes“ sekos diagrama

19 lentelė Panaudojimo atvejo „Administruoti sprendimų medžius“ specifikacija

PA „Administruoti sprendimų medžius“	
Tikslas. Administruoti sprendimų medžius	
Aprašymas. Šis PA apima sprendimų medžio sukūrimą ir administravimą.	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Administratorius
Sužadinimo sąlyga	Projekto pasirinkimas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius sukuria naują sprendimų medį	Sistema sukuria sprendimų medį ir atidaro langą, kuriame administratorius gali matyti sprendimų medžio savybes.
2. Administratorius nurodo taisyklių aibę	Sistema išsaugo pasirinkimą
3. Administratorius sukuria sprendimo tašką	Sistema išsaugo sprendimo tašką
4. Administratorius sukuria veiksmo tašką	Sistema išsaugo veiksmo tašką
5. Administratorius išsaugo sprendimo medį	Sistema išsaugo sprendimo medį ir visas jo savybes
6. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.

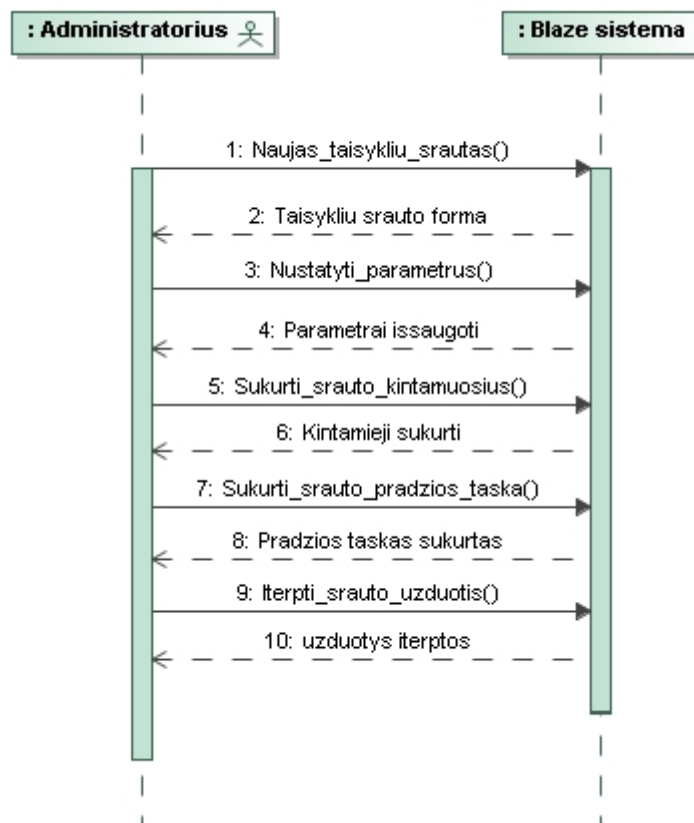


23 pav. Panaudojimo atvejo „Administruoti sprendimų medžius“ sekos diagrama

20 lentelė Panaudojimo atvejo „Administruoti taisyklių srautus“ specifikacija

PA „Administruoti taisyklių srautus“	
Tikslas. Administruoti taisyklių srautus	
Aprašymas. Šis PA apima taisyklių srauto sukūrimą ir administravimą.	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos
Aktorius	Administratorius
Sužadinimo sąlyga	Projekto pasirinkimas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius sukuria naują taisyklių srautą	Sistema sukuria taisyklių srautą ir atidaro langą, kuriame administratorius gali matyti taisyklių srauto savybes.
2. Administratorius nustato parametrus	Sistema išsaugo parametrus

3. Administratorius sukuria srauto kintamuosius	Sistema išsaugo srauto kintamuosius
4. Administratorius sukuria srauto pradžios tašką	Sistema išsaugo pradžios tašką
5. Administratorius įterpia srauto užduotis	Sistema išsaugo įterptas užduotis
6. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.

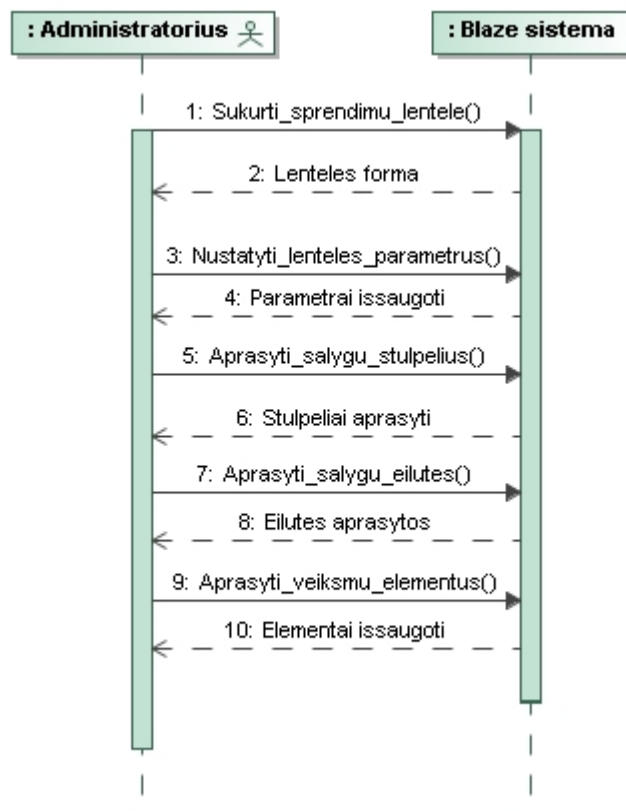


24 pav. Panaudojimo atvejo „Administruoti taisyklių srautus“ sekos diagrama

21 lentelė Panaudojimo atvejo „Tvarkyti sprendimų lenteles“ specifikacija

PA „Tvarkyti sprendimų lenteles“	
Tikslas. Tvarkyti sprendimų lenteles	
Aprašymas. Šis PA apima sprendimų lentelės sukūrimą ir administravimą.	
Prieš sąlyga	Sistemos vartotojas prisijungia prie sistemos

Aktorius	Administratorius
Sužadavimo sąlyga	Projekto pasirinkimas
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Administratorius sukuria naują sprendimų lentelę	Sistema sukuria sprendimų lentelę ir atidaro langą, kuriame administratorius gali matyti sprendimų lentelės savybes.
2. Administratorius nustato parametrus	Sistema išsaugo parametrus
3. Administratorius aprašo sąlygų stulpelius	Sistema išsaugo sąlygų stulpelius
4. Administratorius aprašo sąlygų eilutes	Sistema išsaugo sąlygų eilutes
5. Administratorius aprašo veiksmų elementus	Sistema išsaugo veiksmų elementus
6. Sistema baigia PA	
Alternatyvūs scenarijai	
1. Jei vartotojas neprisijungęs	Sistema pateikia pranešimą „Jūs nesate prisijungęs“ ir baigia PA.



25 pav. Panaudojimo atvejo „Tvarkyti sprendimų lenteles“ sekos diagrama

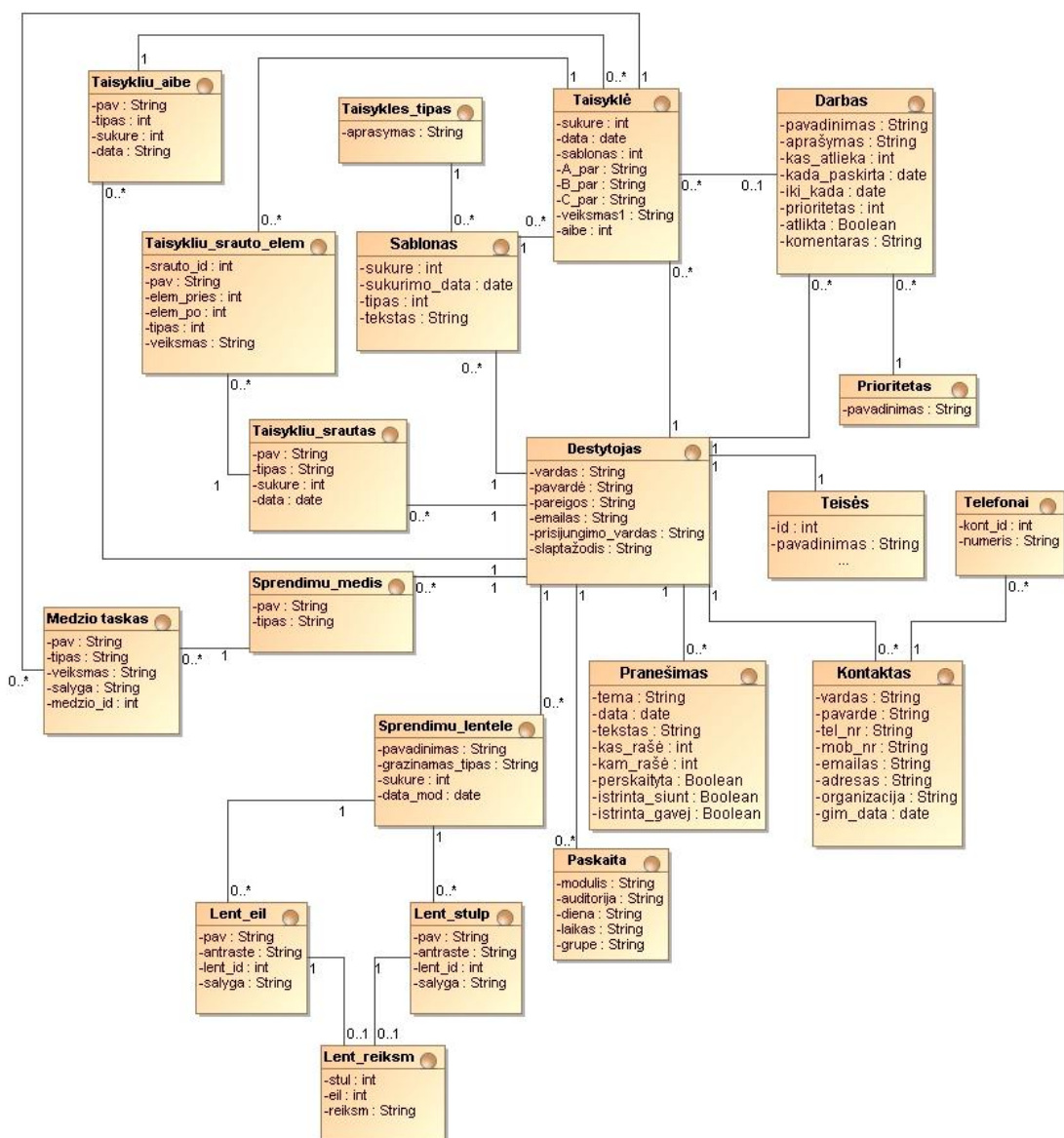
3.1.2. Nefunkcinių reikalavimų specifikacija

1. Veiklos taisyklių valdymui:
 - Veiklos taisyklės valdomos naudojant interneto naršyklę arba Blaze advisor paketą.
 - Taisyklės negali prieštarauti viena kitai.
2. Vartotojo duomenims:
 - Sistemos kaupiamus duomenis ir veiklos taisykles gali pažiūrėti tik sistemoje registruoti vartotojai ir turintys atitinkamas teises;
 - Nutrūkus ryšiui tarp kliento ir serverio, turi būti atstatomas duomenų vientisumas iki paskutinės transakcijos;
 - Duomenys turi būti saugomi MYSQL duomenų bazėje.
 - Veiklos taisyklės saugomos Blaze advisor taisyklių saugykloje.
3. Vartotojo sąsajai:
 - Vartotojo sąsaja turi būti sukurta pagal šiuolaikines technologijas: meniu valdomas pele;
 - Darbo organizavimo sistema turi turėti sąsają per interneto naršyklę.
 - Sąsaja turi būti paprasta ir aiški;
 - Informacinė sistema privalo būti rodoma visose naršyklėse vienodai, t.y. turi būti suderinamumas su visomis naršyklėmis.
4. Pasiekiamumui:
 - Sistema turi būti pasiekama bet kurioje pasaulio vietoje, kur yra interneto ryšys. Pagrindinė sistemos logika atliekama serverio pusėje.
5. Įrangai:
 - Sistema turi būti sukurta naudojant: PHP, MYSQL, JavaScript, JAVA, AJAX, XHTML, CSS;
 - Sistema veikia Apache web serveryje.
 - Veiklos taisyklės valdomos Blaze advisor įrankiu.
6. Saugumui
 - Sistema privalo garantuoti apsaugą ir autorizavimą.
 - Registruotas vartotojas negali keisti kitų vartotojų informacijos, valdyti kitas informacijos sritis, kur jam nėra suteiktos valdymo teisės.

- Sistema privalo užtikrinti vartotojų prisijungimo slaptažodžių slaptumą ir saugumą.

3.2. Dalykinės srities esybių modelis

Išanalizavus dalykinės srities esybes, buvo sudaryta dalykinės srities esybių klasių diagrama (26 pav.). Išskirtos tokios esybės: „Taisyklė“, „Šablonas“, Taisyklės tipas“, „Teisės“, „Prioritetas“, „Dėstytojas“, „Darbas“, „Paskaita“, „Pranešimas“, „Kontaktas“, „Telefonai“.



26 pav. Dalykinės srities esybių modelis

3.3. Reikalavimų analizės apibendrinimas

Reikalavimų analizės etape buvo specifiukuoti panaudos atvejai, kurie atvaizduojami sekų diagramomis (3.1.1. skyrius), parodant sistemos aktorių ir pačios sistemos sąveikas.

3.1.2. skyriuje nustatyti nefunkciniai reikalavimai specifiukuojamai sistemai.

Išsiaiškinus reikalavimus taip pat buvo nustatytos dalykinės srities objektų klasės (3.2. skyrius).

Projekto diagramoms sudaryti buvo pasirinktas MagicDraw 16.0 programinės įrangos paketas.

4. Dėstytojų darbo organizavimo informacinės sistemos projektas

4.1. Sistemos architektūros projektas

4.1.1. Sistemos loginė architektūra

Sistemą sudaro trys dalys: „Taisyklių valdymo posistemis“ (Blaze Advisor Builder) ir „Darbų organizavimo posistemis“ ir „Blaze Advisor taisyklių serveris“.

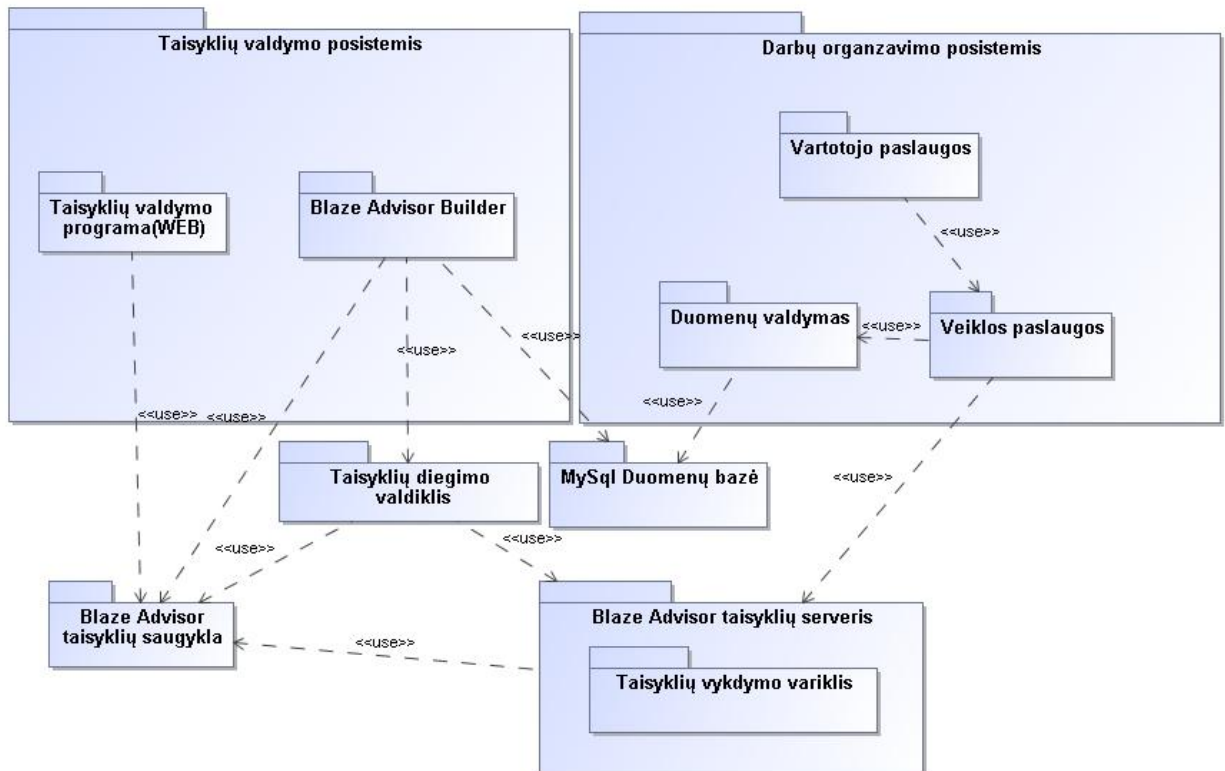
Blaze Advisor Builder – aplinka, skirta veiklos taisyklių kūrimui. Ji yra tiesiogiai orientuota į vartotojus/kūrėjus ir aprūpina žemo lygio programavimo bei grafine aplinka, skirta taisyklių rašymui, redagavimui, peržiūrai ir testavimui.

Blaze Advisor taisyklių serveris, skirtas visai organizacijai. Serveris palaiko lygiagrečių taisyklių vykdymą, sesijos valdymą, planavimą ir dinamišką darbo krūvio valdymą. Taisyklių vykdymo variklis - veiklos taisyklių vykdymo interpretatorius, kuris apibrėžia ir vykdo visus veiklos taisyklių valdymo srautus, kurie buvo sukurti su Blaze Advisor Builder. Interpretatorius veikia kartu su Blaze Advisor taisyklių serveriu tam, kad įvykdytų tas taisykles, kurių reikia taikomosioms programoms.

Taisyklių diegimo valdiklio komponentas tvarko bendradarbiavimą tarp taisyklių saugyklos ir taisyklių vykdymo.

Darbų organizavimo posistemis atsakingas už katedros darbų organizavimą. Posistemį sudaro vartotojo paslaugos, veiklos paslaugos, duomenų valdymas. Veiklos sluoksnį sudaro programos logika, kuri bendradarbiauja su Blaze Advisor taisyklių serveriu ir iškviečia taisyklių vykdymą.

Sistemos loginės architektūros modelis pavaizduotas 27 pav.



27 pav. Sistemos loginė architektūra

4.1.2. Vartotojo paslaugos

Sudarius navigavimo planą (28 pav.) galime matyti, kokie langai ir formos bus sistemoje. Prisijungus, vartotojui atveriamas pagrindinis langas, iš kurio galima pasiekti kitus langus ir valdyti sistemą. Navigavimo plane aptariamas tik darbo organizavimo posistemio navigavimas. Kadangi taisyklių valdymas ir jų kūrimas bus aptariamas kitose dalyse, kur bus detaliau aprašomas ir navigavimas.



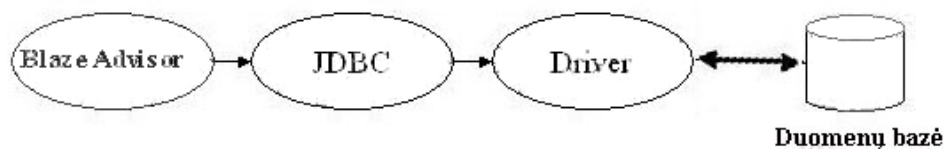
28 pav. Navigavimo planas

4.1.3. Duomenų paslaugos

Veiklos taisyklės saugomos Blaze Advisor saugykloje. Tai mechanizmas leidžiantis taisyklių kūrėjui saugoti ir valdyti katalogus ir elementus. Vartotojas gali prisijungti prie taisyklių saugyklos ir dirbti su saugomais failais naudojant Blaze Advisor Builder.

Sistemos duomenys saugomi MySQL duomenų bazėje. Veiklos logiką realizuojančios klasės kreipiasi į duomenis, esančius duomenų bazėje. MySQL duomenų bazė taip pat saugo veiklos taisyklių parametrus, kuriuos nusistato vartotojai.

Su Blaze Advisor esančiu vedliu galime nustatyti klases, kurios komunikuos su duomenų bazių lentelėmis per JDBC jungtį (29 pav.). JDBC tai standartinė sąsaja SQL duomenų bazių prieigai.



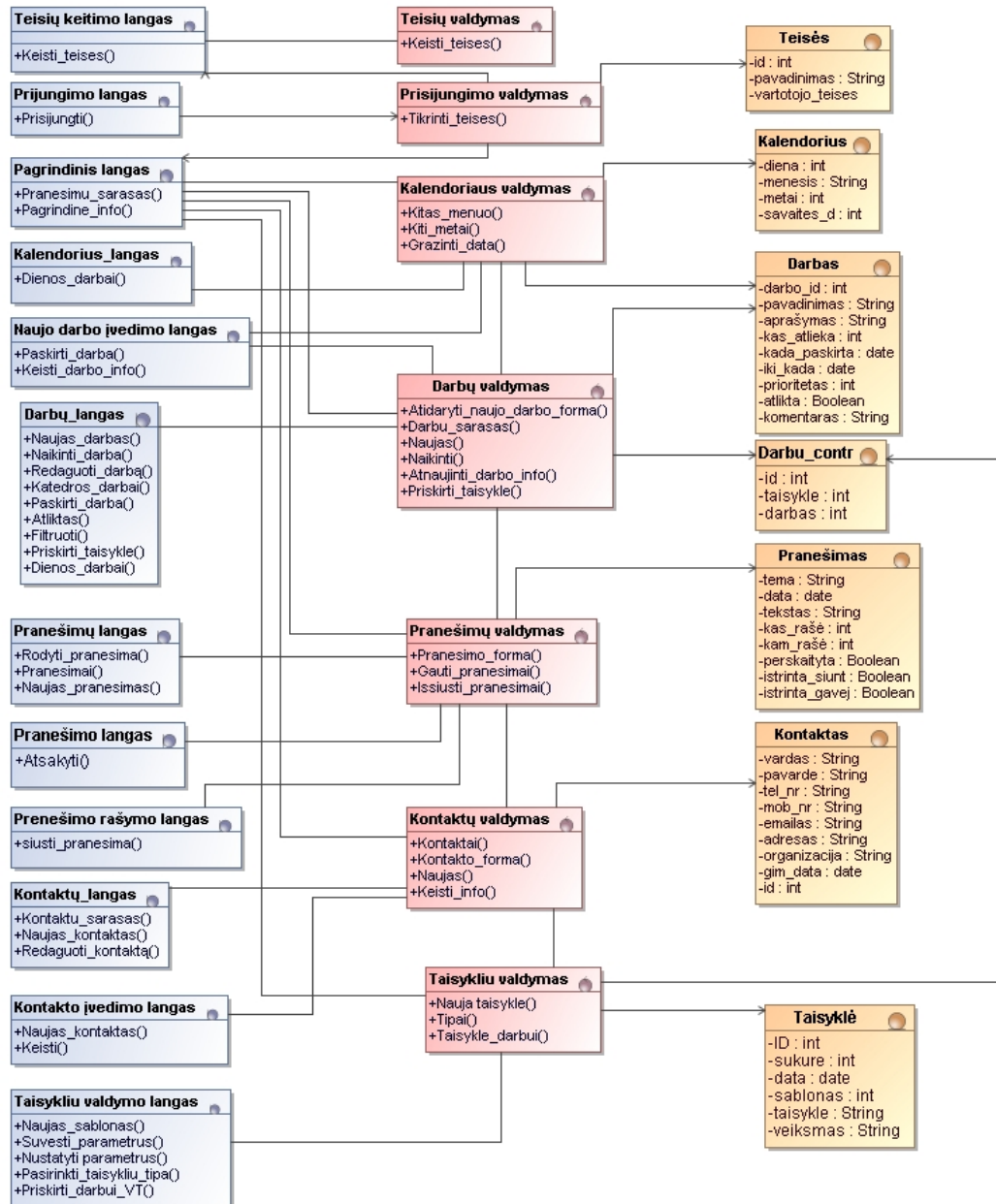
29 pav. Prisijungimas prie duomenų bazės naudojant JDBC jungtį

Blaze advisor objektai gali naudoti ne tik lentelių duomenis, bet ir poschemių (angl.view). Taip pat galima naudoti SQL užklausas duomenų filtravimui.

4.2. Detalus projektas

4.2.1. Darbų organizavimo posistemio detalus projektas

Darbo organizavimo posistemio detalus projektas (30 pav.) atvaizduoja klasių diagramą su operacijomis. Klasių operacijų aprašymai pateikti lentelėse.



30 pav. UML klasių diagrama su operacijomis

Klasė „Prisijungimo langas“

Metodas	Paskirtis
Prisijungti()	Paima duomenis iš formos ir siunčia į valdiklį „Prisijungimo valdymas“

Klasė „Prisijungimo valdymas“

Metodas	Paskirtis
Tikrinti_teises()	Patikrina prisijungimo duomenis ir palygina su esančiais duomenų bazėje.

Klasė „Teisių keitimo langas“

Metodas	Paskirtis
Keisti_teises()	Paima duomenis iš formos ir siunčia į valdiklį „Teisių valdymas“

Klasė „Teisių valdymas“

Metodas	Paskirtis
Keisti_teises()	Keičia prisijungimo duomenis

Klasė „Pagrindinis langas“

Metodas	Paskirtis
Pranešimų_sąrašas()	Išveda į ekraną naujus neperskaitytus pranešimus
Pagrindinė_info()	Išveda į ekraną svarbią informaciją, dienos darbus, priminimus.

Klasė „Kalendoriaus langas“

Metodas	Paskirtis
Dienos_darbai()	Išveda į ekraną dienos darbų sąrašą.

Klasė „Kalendoriaus valdymas“

Metodas	Paskirtis
Kitas_menuo()	Gražina sekančio mėnesio numerį.
Kiti_metai()	Gražina sekančius metų skaičius.
Grazinti_data()	Gražina data.

Klasė „Naujo darbo įvedimo langas“

Metodas	Paskirtis
Paskirti_darbą()	Paima duomenis iš formos ir siunčia į valdiklį „Darbų valdymas“
Keisti_darbo_info()	Paima duomenis iš formos ir siunčia į valdiklį „Darbų valdymas“

Klasė „Darbų langas“

Metodas	Paskirtis
Naujas_darbas()	Iškviečiama naujo darbo forma
Naikinti_darbą()	Pasirinktų darbų panaikinimas
Redaguoti_darbą()	Iškviečiama darbo redagavimo forma
Katedros_darbai()	Į ekraną išvedamas katedros darbų sąrašas

Paskirti_darbai()	Darbuotojui išvedamas paskirtų darbų sąrašas
Atliktas()	Pažymimi atlikti darbai.
Filtruoti()	Į ekraną išvedami darbai pagal tam tikrą filtrą
Dienos_darbai()	Išveda į ekraną dienos darbų sąrašą.

Klasė „Darbų valdymas“

Metodas	Paskirtis
Darbų_sąrašas()	Formuojamas darbų sąrašas
Naujas()	Sukuriamas naujas darbas
Naikinti()	Panaikinamas darbas
Atnaujinti_darbo_info()	Atnaujinama darbo informacija

Klasė „Pranešimų langas“

Metodas	Paskirtis
Rodyti_pranešimą()	Į ekraną išveda pranešimo informaciją
Pranešimai()	Į ekraną išveda pranešimų sąrašą
Naujas_pranešimas()	Į ekraną išveda naujo pranešimo formą

Klasė „Pranešimų valdymas“

Metodas	Paskirtis
Pranešimo_forma()	Sukuriamas naujas pranešimas
Gauti_pranešimai()	Formuojamas gautų pranešimų sąrašas
Išsiųsti_pranešimai()	Formuojamas išsiųstų pranešimų sąrašas

Klasė „Kontaktų langas“

Metodas	Paskirtis
Kontaktų_sarasas()	Į ekraną išvedamas kontaktų sąrašas
Naujas_kontaktas()	Iškviečiama naujo kontakto forma
Redaguoti_kontakta()	Iškviečiama redagavimo forma

Klasė „Kontakto įvedimo langas“

Metodas	Paskirtis
Naujas_kontaktas()	Paima duomenis iš formos ir siunčia į valdiklį „Kontaktų valdymas“
Keisti()	Paima duomenis iš formos ir siunčia į valdiklį „Kontaktų valdymas“

Klasė „Kontaktų valdymas“

Metodas	Paskirtis
Kontaktai()	Formuojamas kontaktų sąrašas
Kontakto_forma()	Iškviečiama naujo kontakto forma
Naujas()	Sukuriamas naujas kontaktas
Keisti_info()	Keičia kontakto info

4.2.2. Taisyklių valdymo posistemio detalus projektas

Taisyklių valdymo posistemį sudaro Blaze Advisor komponentai. Blaze Advisor kūrėjai nepateikia, kaip sudaryta programa viduje, todėl jų klasių detaliau nenagrinėsime.

Blaze Advisor turi galimybę sukurti internetinę taisyklių valdymo dalį. Tam naudojama Java 2 Platform, Enterprise Edition (J2EE). Tai standartinė daugialypių programų kūrimo Java kalba platforma, paremta moduliniais komponentais, vykdomais programų serveryje. J2EE naudojama keletas technologijų, tokių kaip JDBC ar CORBA, praplėstų pritaikant tikslinėms sistemoms. J2EE sistema yra kelių lygių (angl. tiers).

Vartotojui artimiausias lygis yra interneto naršyklė, veikianti vartotojo kompiuteryje. Bendrauti su naršyklėmis J2EE apibrėžia servletus ir JSP specifikaciją. Būtent šie komponentai, naudojami kartu, leidžia atskirti verslo logiką nuo vaizdinės dalies, taip supaprastinant interneto sistemos kūrimą ir palaikymą.

Servletai – tai Java klasės, paveldinčios tam tikrą abstrakčią klasę ir realizuojančios metodus, sukuriančius interneto puslapį.

Kiekvienas JSP puslapis paverčiamas servletu, o galiausiai ir sukompiliuojamas. Pirmą kartą paprašius puslapio, jo gali tekti palaukti (nes puslapis verčiamas servletu ir kompiliuojamas), tačiau antrą ir paskesnius kartus puslapis klientui bus pateikiamas greičiau.

Anksčiau minėti komponentai atsakingi išimtinai už interneto puslapių kūrimą (jų vaizdą). Už visą darbo logiką yra atsakingi interneto komponentų kviečiami EJB (angl. enterprise java beans) moduliai.

Šiame darbe naudojami tik EJB komponentai, kurie diegiami Blaze Advisor taisyklių serveryje. Vartotojo aplinka, skirta taisyklių parametrų redagavimui, integruojama į PHP kalba sukurtą aplinką. Dėl šios priežasties buvo atsisakyta Blaze Advisor siūlomų galimybių ir sukurtos nuosavos klasės, bei formos, kurios paprasčiau susiintegruotu su darbų organizavimo posistemiu.

4.2.3. Blaze Advisor taisyklių serverio detalus projektas

Blaze Advisor taisyklių serveris (Rule Server) yra Blaze Advisor veiklos taisyklių valdymo sistemos komponentas. Naudojant taisyklių serverį galima įdiegti su BA Builder sukurtus taisyklių projektus kaip taisyklių paslaugas. Taisyklių paslaugos veikia kaip programos komponentas, kuris atlieka taisyklių projekto aprašytus veiklos procesus, ir yra

prieinamas kitiems sistemos komponentams. Naudojant EJB tipo įdiegimą, taisyklių serveris teikia paslaugas, kai EJB konteineris apdoroja klientų užklausas.

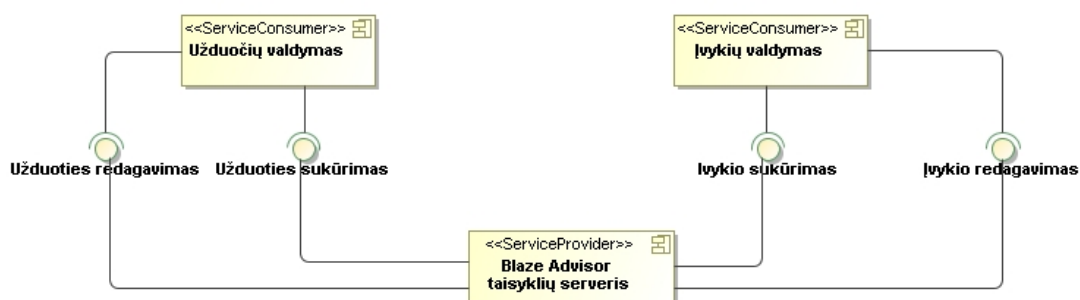
Galima sukonfigūruoti Blaze Advisor taisyklių serverį taip, kad jis apdorotų vieną arba daugiau taisyklių paslaugų. Taisyklių paslaugos gali būti prieinamos vieno arba kelių nutolusių klientų naudojant paketinį apdorojimą.

Taisyklių projekto kaip taisyklių paslaugų diegimas susideda iš šių etapų:

- Taisyklių serverio sukonfigūravimas pagal platformą, į kurią diegsime taisykles.
- Taisyklių projekto nustatymas paslaugoms teikti.
- Aprašymas paslaugų įėjies taškų, kurie leis klientams pasiekti paslaugas.

Įėjies taškai tai yra veiklos metodai, kuriuos aprašo argumentai, per kuriuos perduodami duomenys į taisykles, bei grąžinamas rezultatas.

31 paveikslėlyje pavaizduoti darbų organizavimo posistemio elementai („Užduočių valdymas“ ir „Įvykių valdymas“), kurie yra veiklos taisyklių paslaugų naudotojai. Paslaugas teikia taisyklių serveris, kuris šiuo atveju turi keturi įėjies taškus: „Užduoties redagavimas“, „Užduoties sukūrimas“, „Įvykio sukūrimas“, „Įvykio redagavimas“.



31 pav. Blaze Advisor taisyklių serverio teikiamos paslaugos

Blaze Advisor pateikia greitąjį diegimo vedlį, kuris padeda sugeneruoti atitinkamą programinį kodą ir diegimui reikalingus konfigūracinius failus.

EJB greitis diegimo vedlys gali sugeneruoti failus šiems EJB konteinerių tipams:

- J2EE 1.4 (Sun 8.2)
- JavaEE5
- JBoss 4.0.4
- OC4J (10.1.3)

- Oracle 9iAS Release 2 (9.0.3)
- WebLogic 8.1, 9.1, and 9.2
- WebSphere 6.0, and 6.1

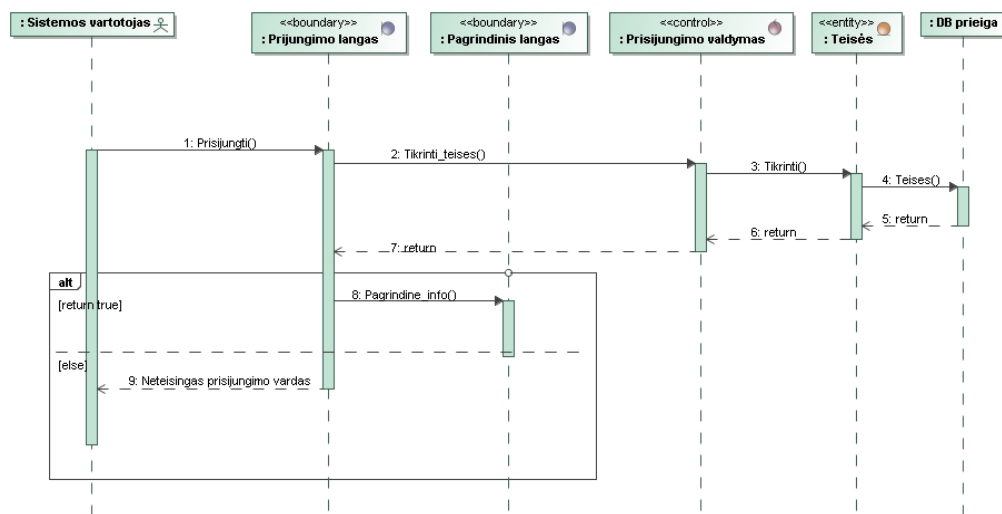
Norint generuoti kitiems konteinerių tipams reiktų pasitikrinti J2EE suderinamumo lygį.

Šiame darbe yra pasirinktas naudojimui J2EE 1.4 (Sun 8.2) serveris.

4.3. Sistemos elgsenos modelis

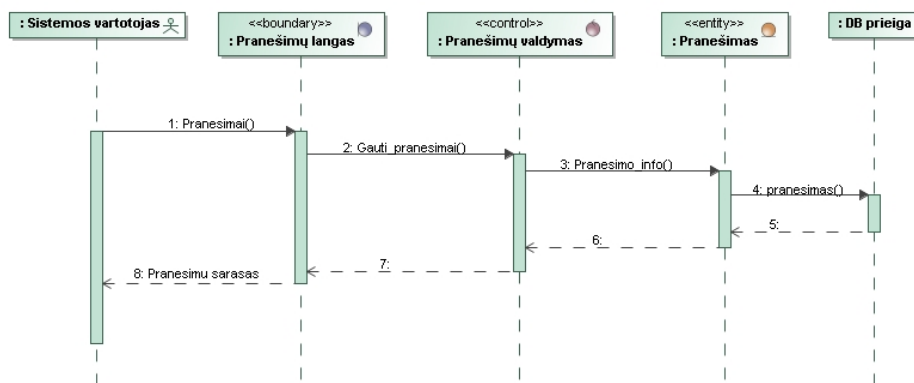
Analizuojant darbų organizavimo posistemį naudojamos panaudojimo atvejų sekų diagramos, kurios atvaizduoja kaip sistemos klasės dirba tarpusavyje kiekvieno panaudojimo atveju. Aprašant veiklos taisyklių valdymo dalį, panaudojamo atvejai aprašomi veiklos diagramomis. Tai leidžia lengviau suprasti Blaze Advisor veiklos principus.

Panaudojimo atvejo „Prisijungti“ sekos diagramoje (32 pav.) matome, kaip vartotojas kreipiasi į sistemą. Patikrinami jo duomenys ir jei duomenys patvirtinami, vartotojas prijungiamas prie sistemos.



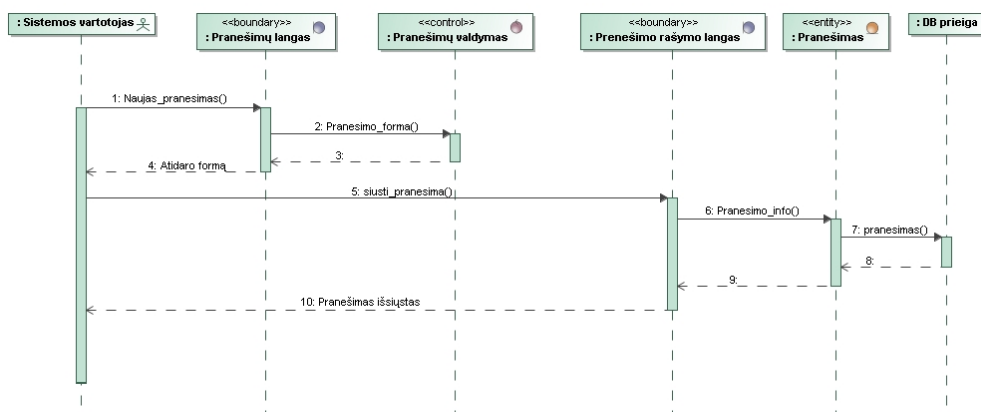
32 pav. Panaudojimo atvejo „Prisijungti“ sekų diagrama

Panaudojimo atvejo „Peržiūrėti pranešimus“ sekos diagramoje (33 pav.) matome, kaip peržiūrimi pranešimai.



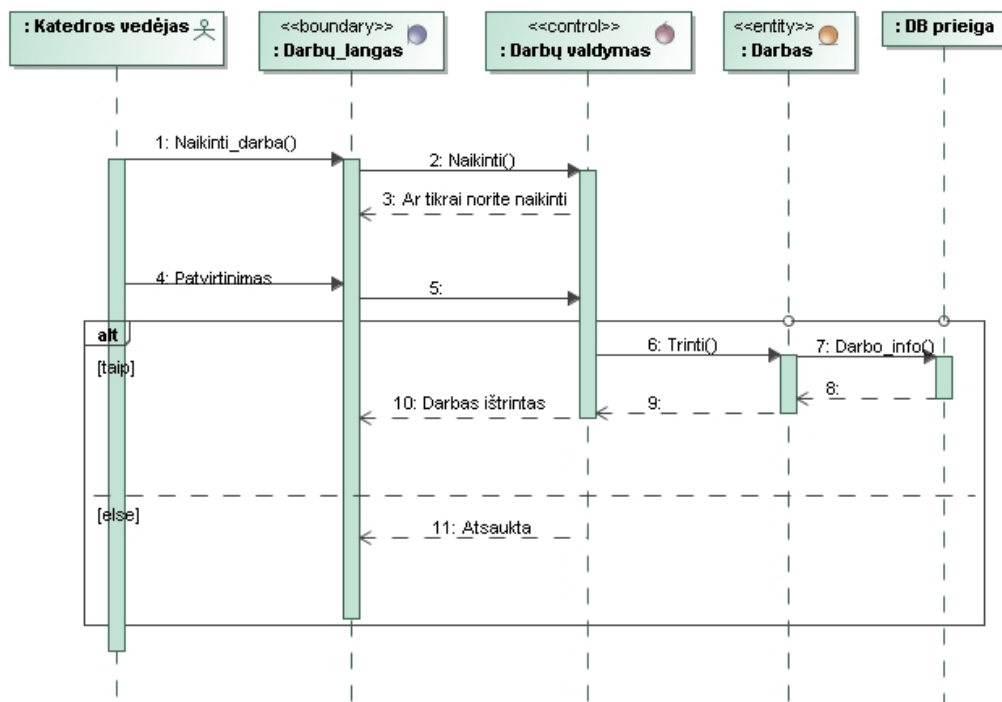
33 pav. Panaudojimo atvejo „Peržiūrėti pranešimus“ sekos diagrama

Panaudojimo atvejo „Siųsti pranešimą“ sekos diagramoje (34 pav.) matome, kaip išsiunčiami pranešimai.



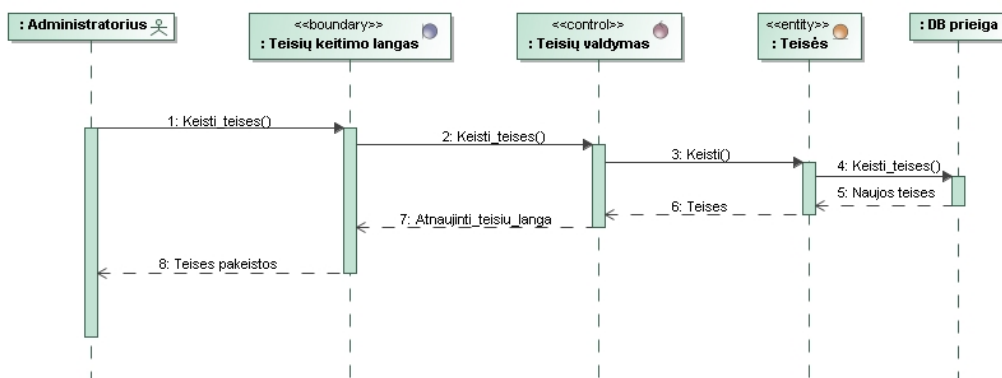
34 pav. Panaudojimo atvejo „Siųsti pranešimą“ sekos diagrama

Panaudojimo atvejo „Panaikinti darbą darbuotojui“ sekos diagramoje (35 pav.) pavaizduota, kaip panaikinamas darbas, jei pasikeičia kokios nors sąlygos.



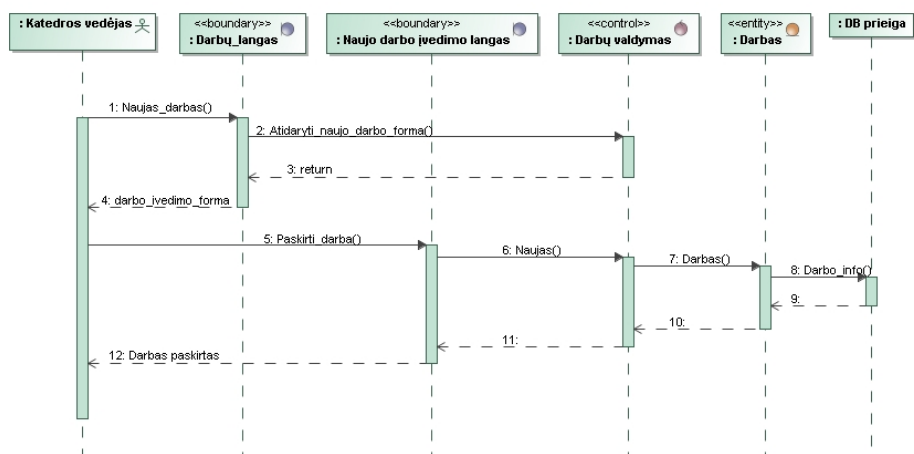
35 pav. Panaudojimo atvejo „Panaikinti darbą darbuotojui“ sekos diagrama

Panaudojimo atvejo „Suteikti teises“ sekos diagramoje (36 pav.) pavaizduotas teisių suteikimo procesas vartotojui.



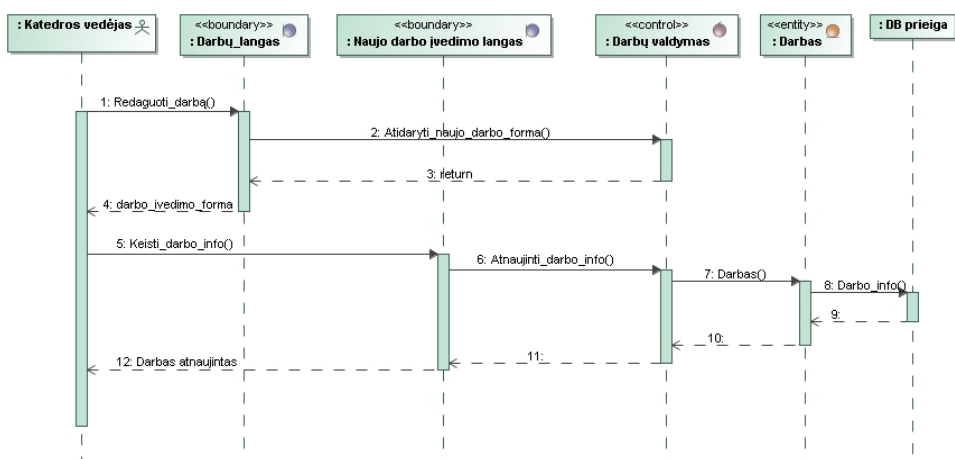
36 pav. Panaudojimo atvejo „Suteikti teises“ sekos diagrama

Panaudojimo atvejo „Paskirti darbą darbuotojui“ sekos diagramoje (37 pav.) parodyta, kaip elgiasi klasės, paskiriant darbą darbuotojui.



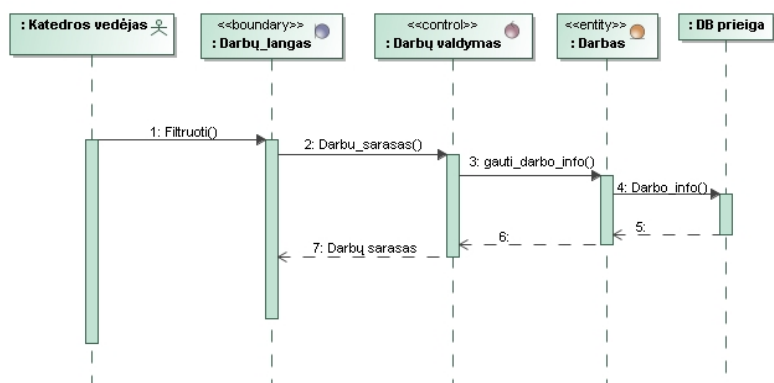
37 pav. Panaudojimo atvejo „Paskirti darbą darbuotojui“ sekos diagrama

Panaudojimo atvejo „Redaguoti darbą“ sekos diagramoje (38 pav.) matome, kaip darbuotojas išsirenką darbą, pakeičia jo duomenis ir vėl išsaugo duomenų bazėje.



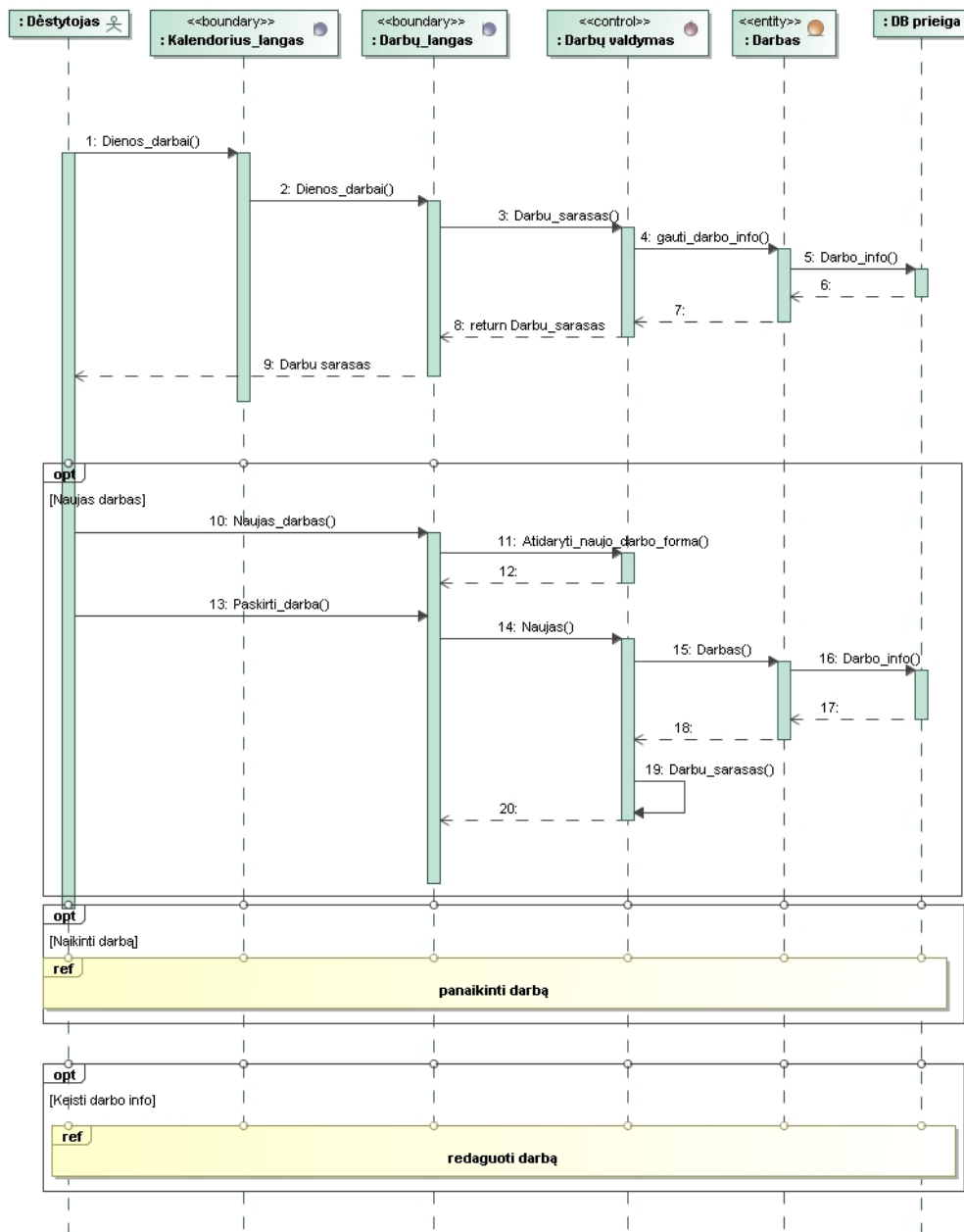
38 pav. Panaudojimo atvejo „Redaguoti darbą“ sekos diagrama

Panaudojimo atvejo „Peržiūrėti katedros darbus“ sekos diagramoje (39 pav.) matome, kaip peržiūrimi katedroje atlikti darbai ir kaip elgiasi klasės vykdant šį panaudojimo atvejį.



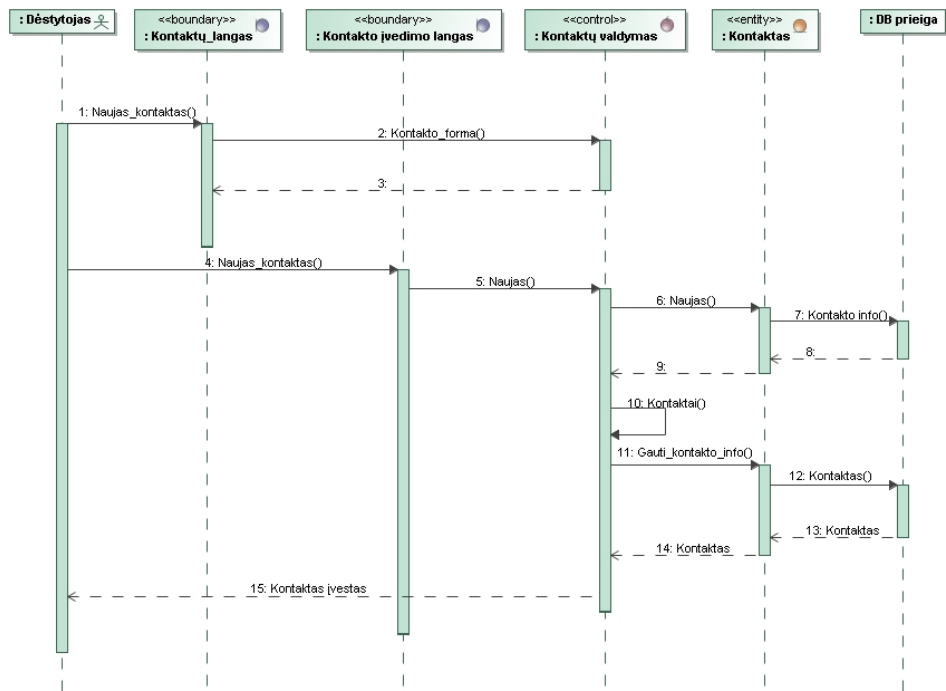
39 pav. Panaudojimo atvejo „Peržiūrėti katedros darbus“ sekos diagrama

Panaudojimo atvejo „Susidaryti savo darbotvarkę“ sekos diagramoje (40 pav.) matome, kaip vykdomas procesas, kai vartotojas susidaro savo darbotvarkę.



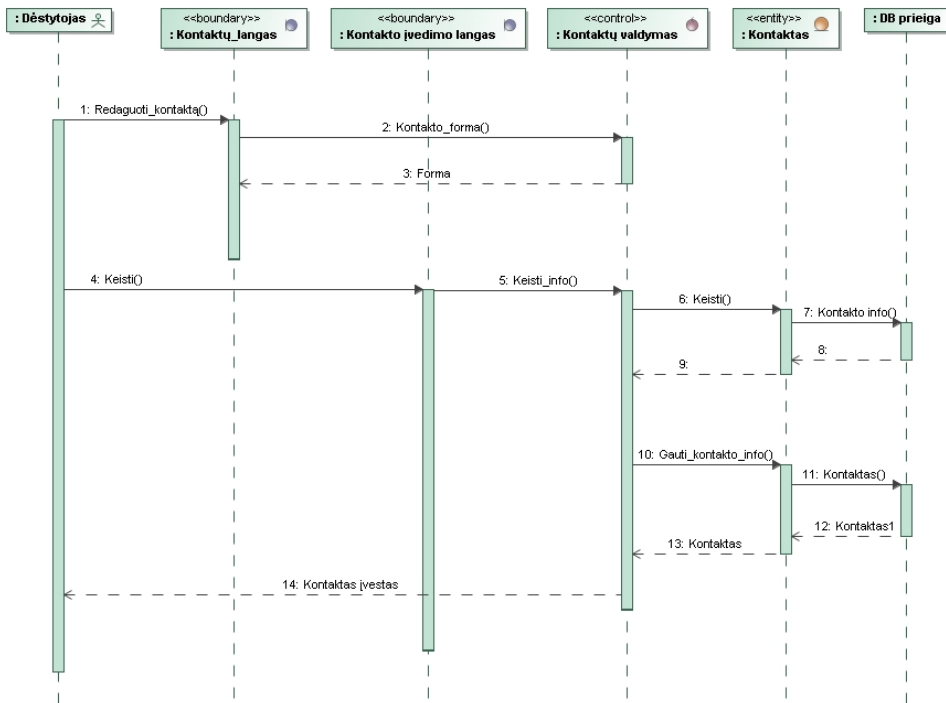
40 pav. Panaudojimo atvejo „Susidaryti savo darbotvarkę“ sekos diagrama

Panaudojimo atvejo „Įvesti kontaktą“ sekos diagramoje (41 pav.) pavaizduota, kaip įvedami naujo kontakto duomenys ir įrašomi į duomenų bazę.



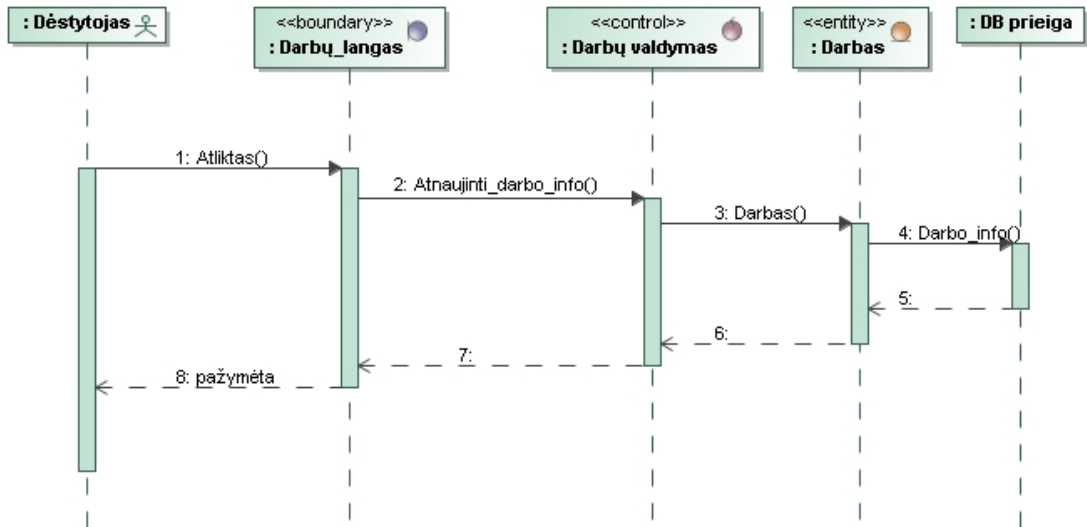
41 pav. Panaudojimo atvejo „Įvesti kontaktą“ sekos diagrama

Panaudojimo atvejo „Redaguoti kontaktą“ sekos diagramoje (42 pav.) pavaizduota, kaip įvedami pakeisti kontakto duomenys ir įrašomi į duomenų bazę.



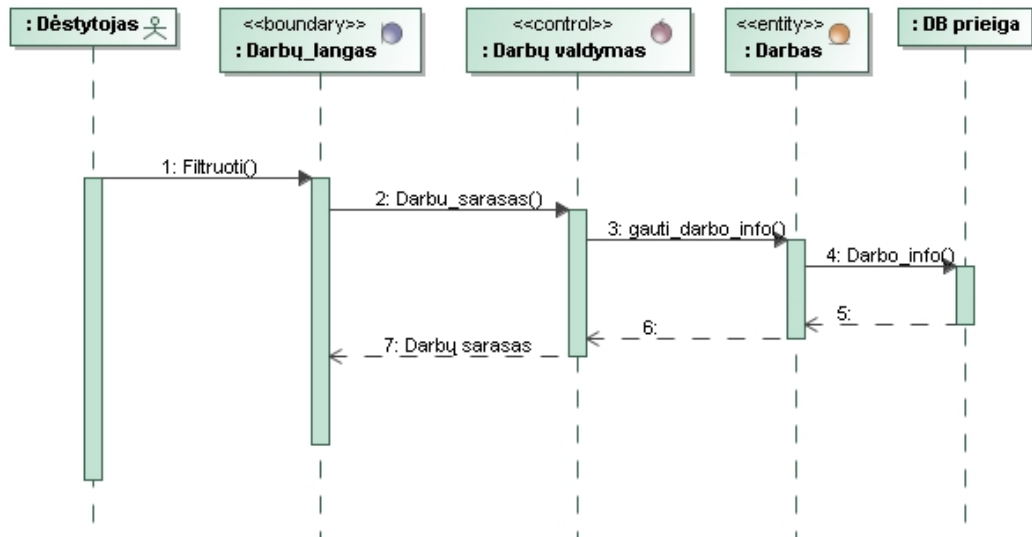
42 pav. Panaudojimo atvejo „Redaguoti kontaktą“ sekos diagrama

Panaudojimo atvejo „Pranešti apie atliktus darbus“ sekos diagramoje (43 pav.) matome, kaip atliekamas pranešimo apie atliktus darbus procesas.



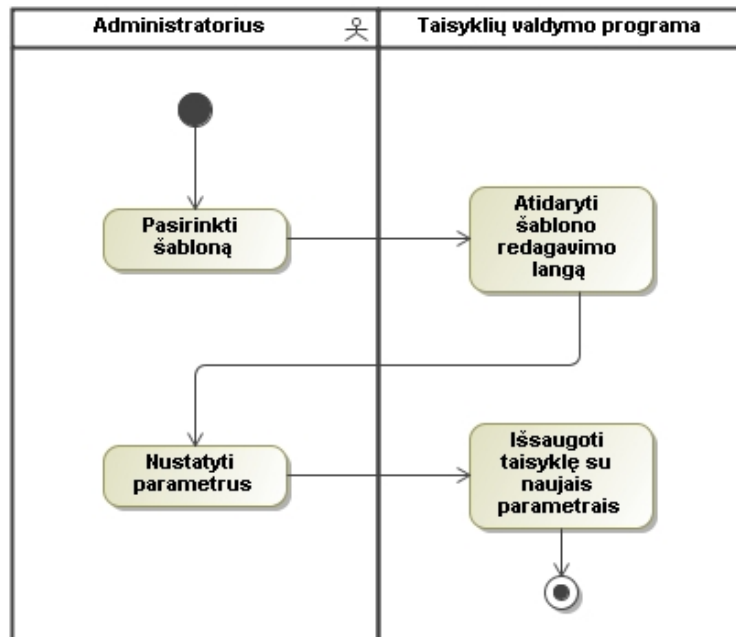
43 pav. Panaudojimo atvejo „Pranešti apie atliktus darbus“ sekos diagrama

Panaudojimo atvejo „Peržiūrėti paskirtus darbus“ sekos diagramoje (44 pav.) pavaizduota, kaip peržiūrimi paskirti darbai, kaip atliekamas informacijos paėmimas iš duomenų bazės ir atvaizdavimas į ekraną.



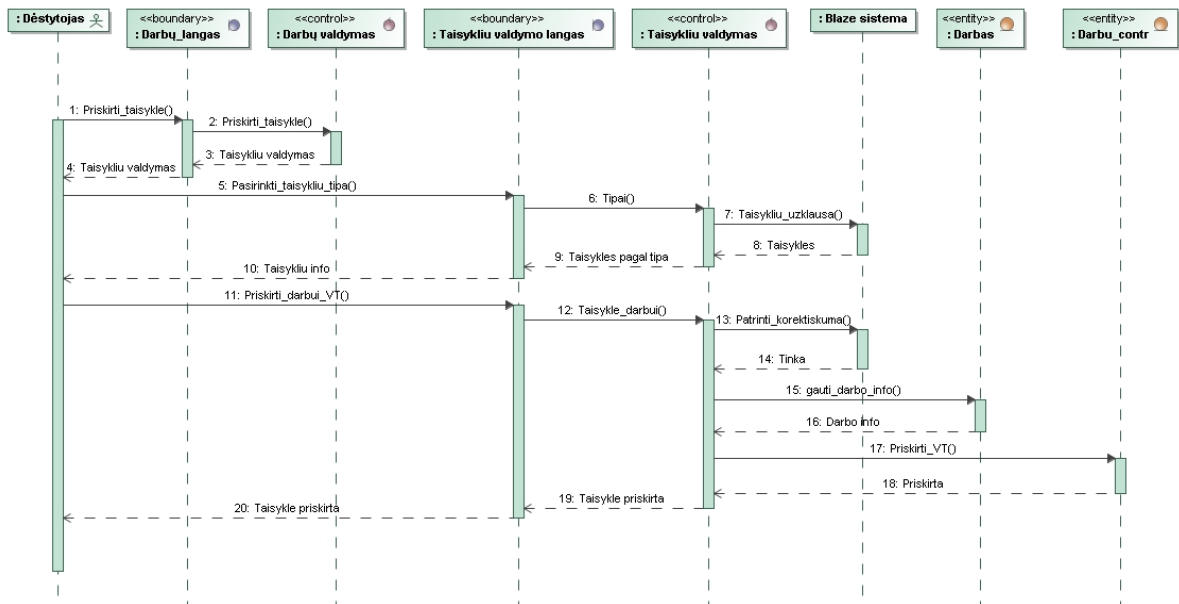
44 pav. Panaudojimo atvejo „Peržiūrėti paskirtus darbus“ sekos diagrama

Panaudojimo atvejo „Pasirinkti VT elementų reikšmes“ veiklos diagramoje (45 pav.) pavaizduota, kaip dėstytojas susikuria savo veiklos taisykles su pasirinktomis taisyklių reikšmėmis naudojant šablonus.



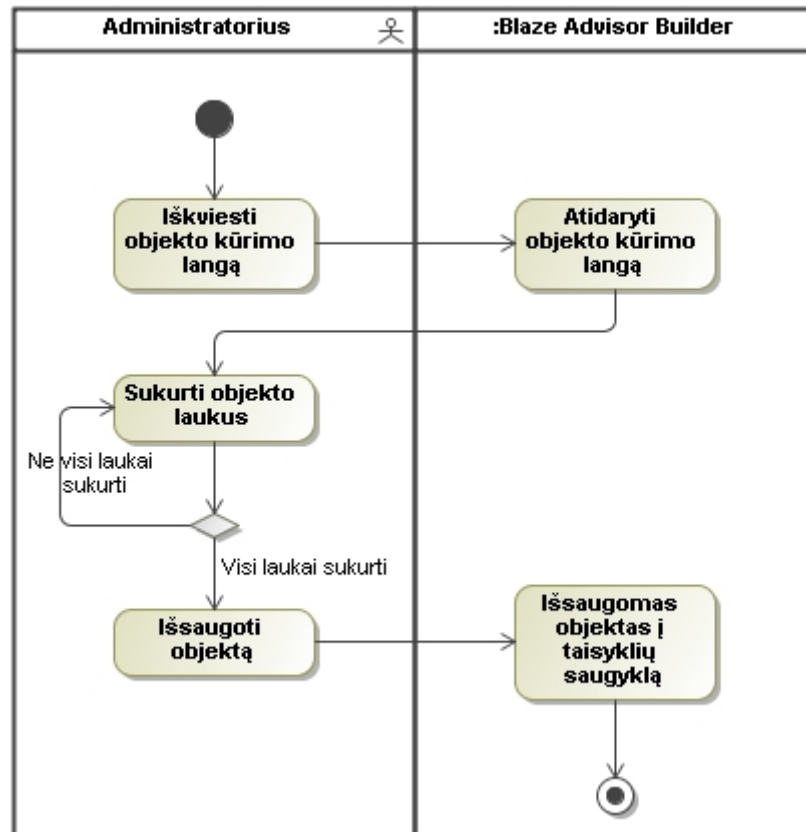
45 pav. Panaudojimo atvejo „Pasirinkti VT elementų reikšmes“ veiklos diagrama

Panaudojimo atvejo „Nustatyti VT darbotvarkei“ sekos diagramoje (46 pav.) pavaizduota, kaip dėstytojas priskiria veiklos taisykles savo darbotvarkei.



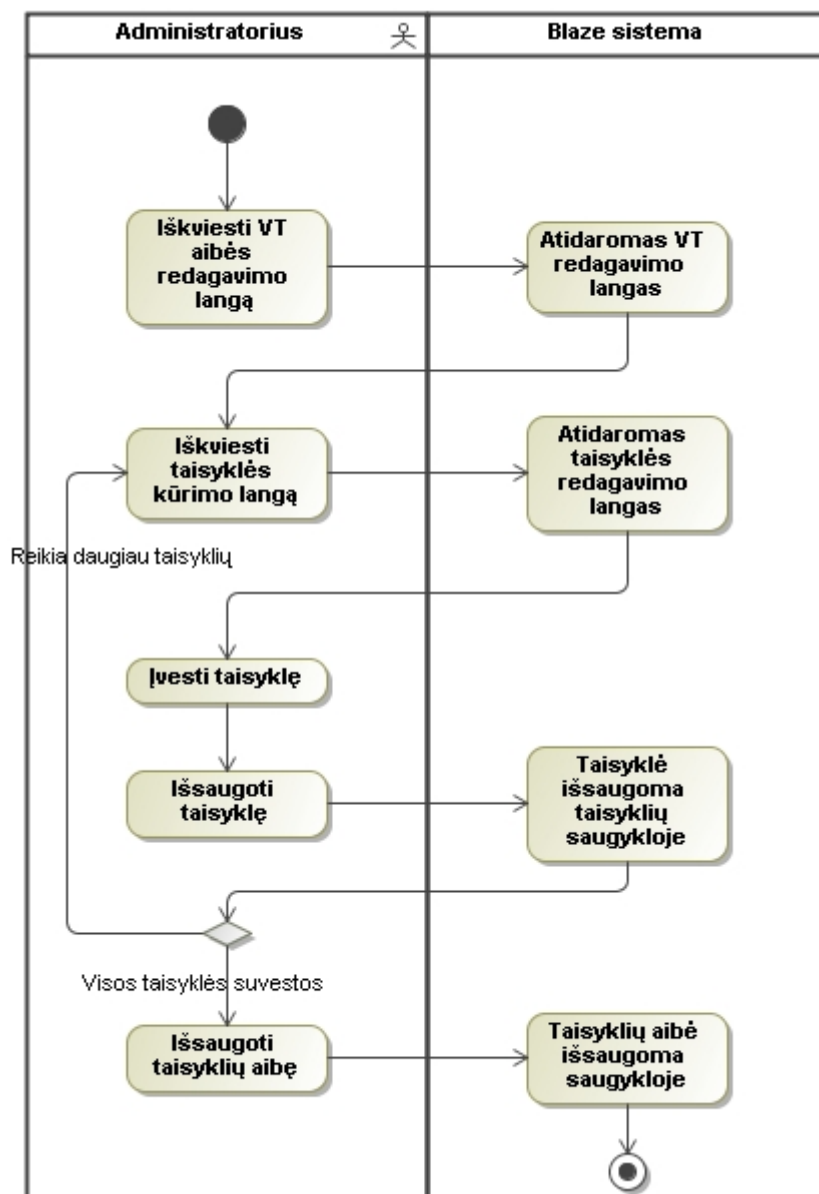
46 pav. Panaudojimo atvejo „Nustatyti VT darbotvarkei“ sekos diagrama

Panaudojimo atvejo „Administruoti objektus“ veiklos diagramoje (47 pav.) pavaizduota, kaip administratorius kuria objektus Blaze Advisor sistemoje.



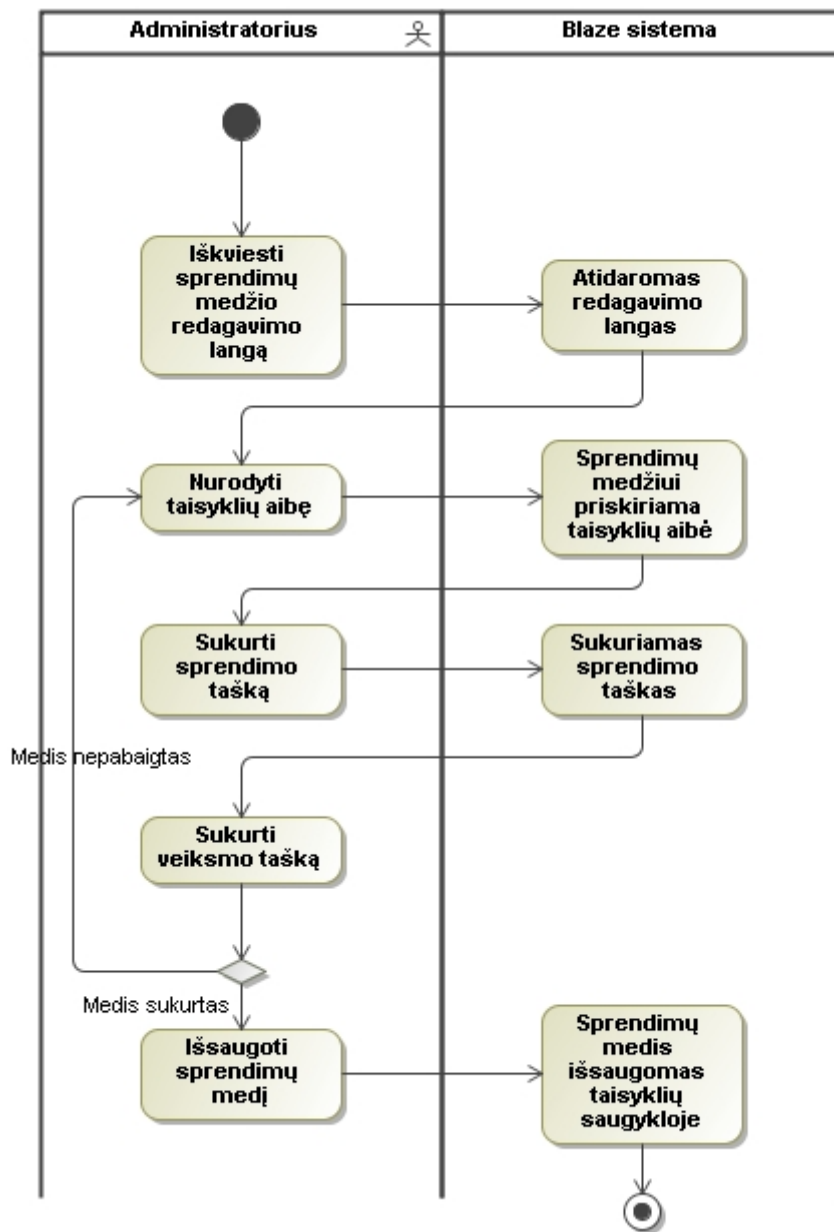
47 pav. Panaudojimo atvejo "Administruoti objektus" veiklos diagrama

Panaudojimo atvejo „Administruoti taisyklių aibes“ veiklos diagramoje (48 pav.) pavaizduota, kaip administratorius administruoja taisyklių aibes Blaze Advisor sistemoje.



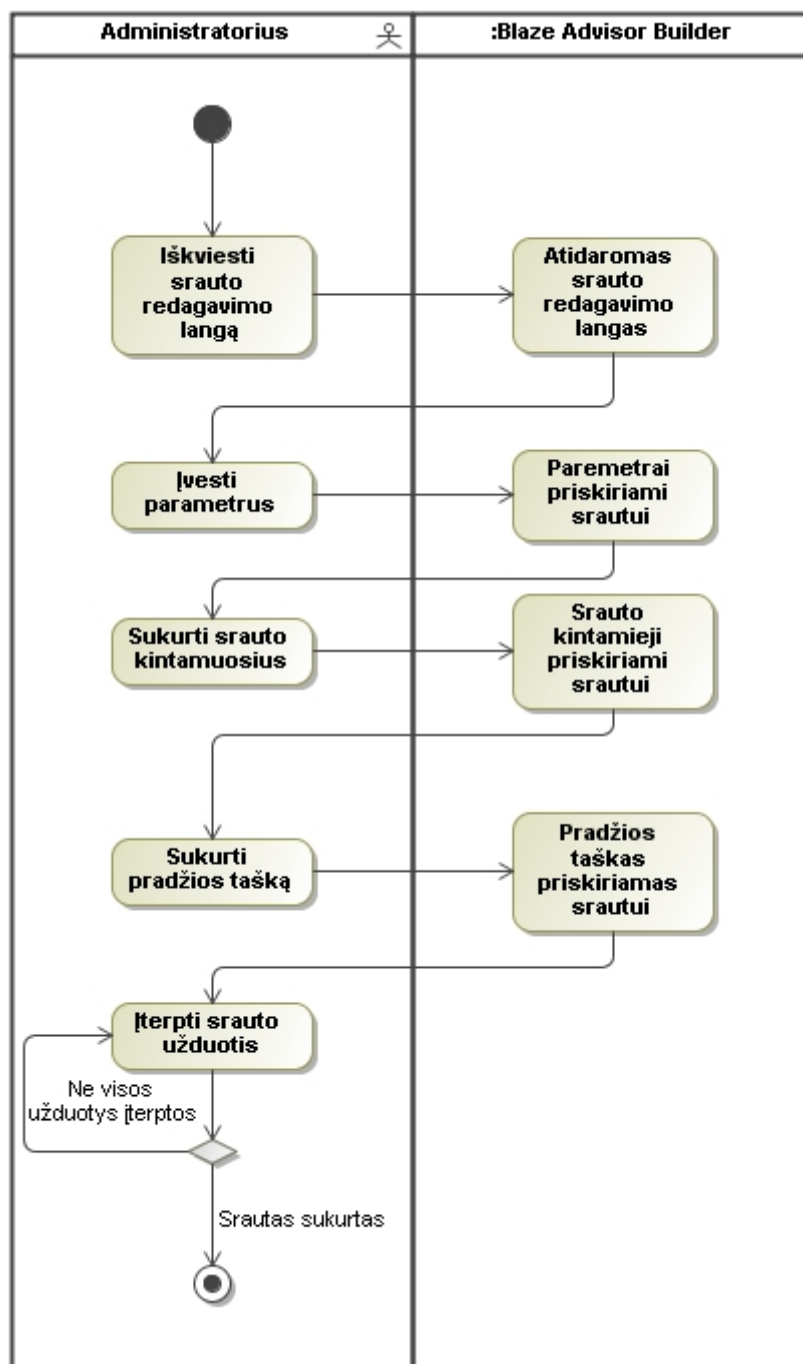
48 pav. Panaudojimo atvejo "Administruoti taisyklių aibes" veiklos diagrama

Panaudojimo atvejo „Administruoti sprendimų medžius“ veiklos diagramoje (49 pav.) pavaizduota, kaip administratorius administruoja sprendimų medžius Blaze Advisor sistemoje.



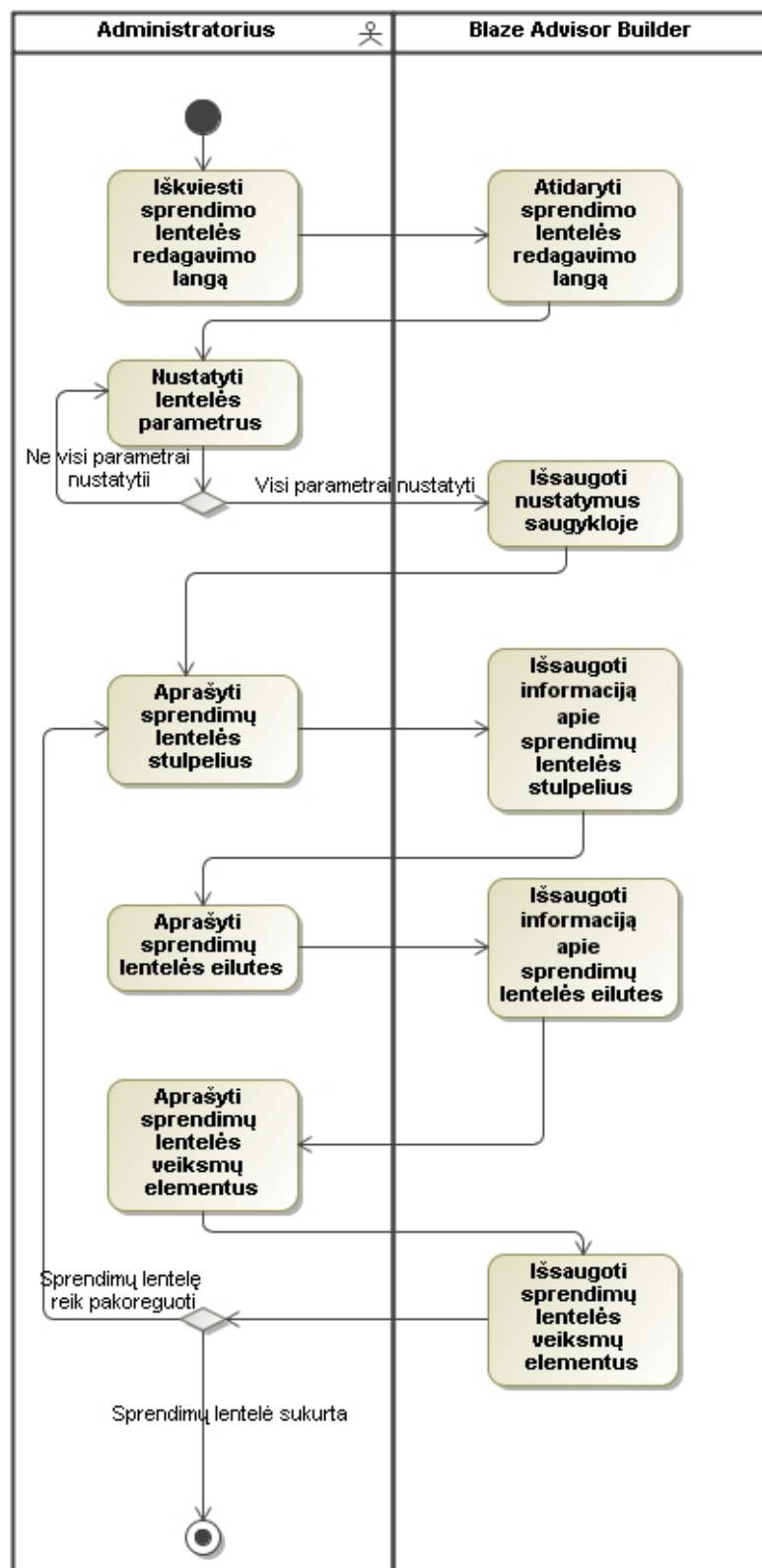
49 pav. Panaudojimo atvejo "Administruoti sprendimų medžius" veiklos diagrama

Panaudojimo atvejo „Administruoti taisyklių srautus“ veiklos diagramoje (50 pav.) pavaizduota, kaip administratorius administruoja taisyklių srautus Blaze Advisor sistemoje.



50 pav. Panaudojimo atvejo "Administruoti taisyklių srautus" veiklos diagrama

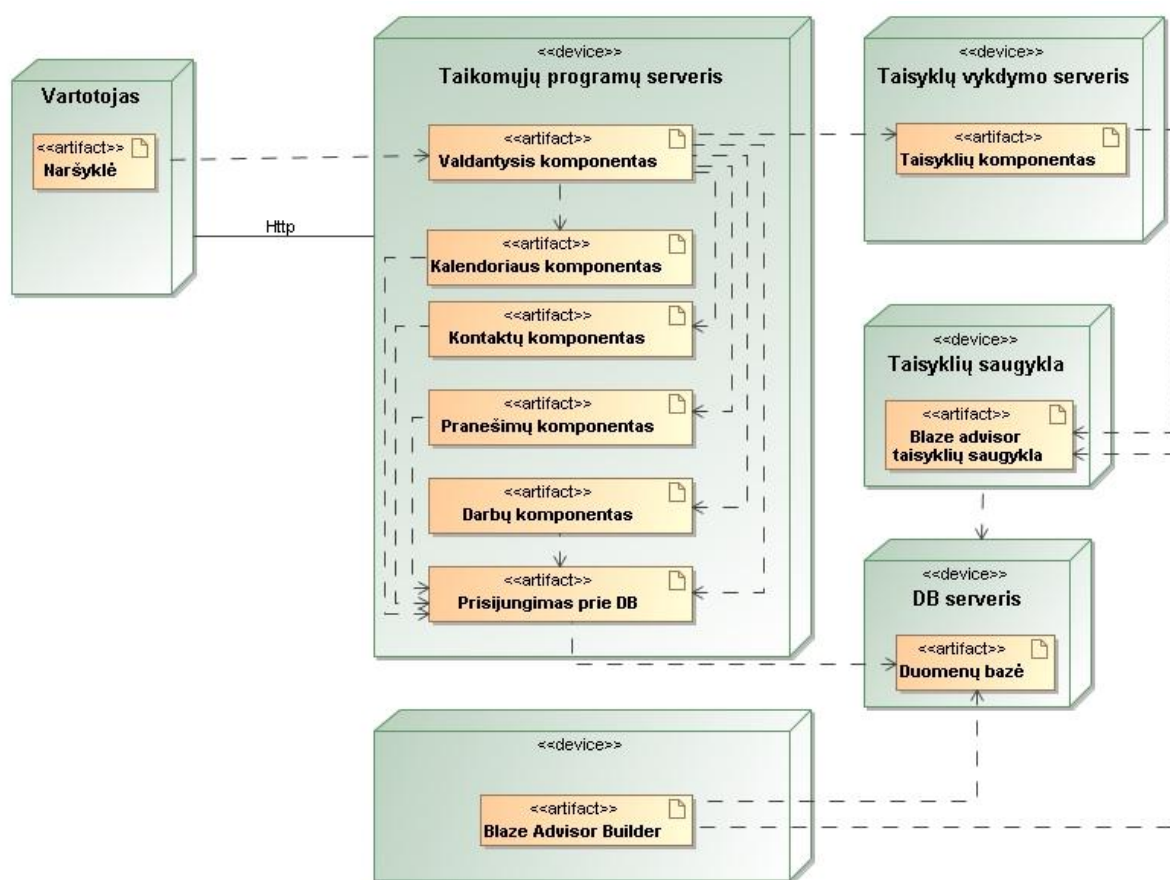
Panaudojimo atvejo „Tvarkyti sprendimų lenteles“ veiklos diagramoje (51 pav.) pavaizduota, kaip administratorius tvarko sprendimų lenteles Blaze Advisor sistemoje.



51 pav. Panaudojimo atvejo "Tvarkyti sprendimų lenteles" veiklos diagrama

4.4. Realizacijos modelis

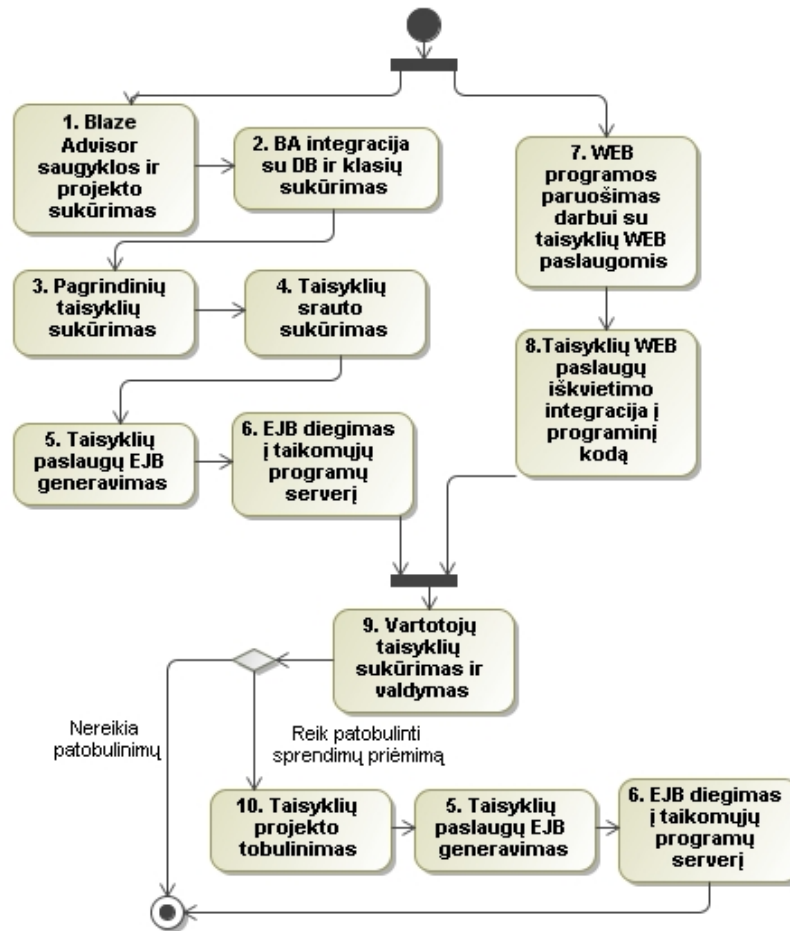
Komponentų įdiegimo diagrama (52 pav.) atvaizduoja kaip bus realizuota sistema. Vartotojas naršyklės pagalba iškviečia reikalingas funkcijas. Jei reikalingi duomenys taikomųjų programų serveris kreipiasi į duomenų bazės serverį. Taisyklių apdorojimui naudojamas taisyklių vykdymo serveris. Taisyklių kūrimui naudojamas Blaze Advisor paketas.



52 pav. Diegimo modelis

5. Veiklos taisyklių integracijos metodika

Veiklos taisyklių (VT) integracijos į taikomųjų uždavinių sistemą procesas pavaizduotas 53 paveikslėlyje. Veiklos diagramoje integracijos etapai sužymėti skaitmenimis. Apžvelgsime kiekvieną etapą atskirai.



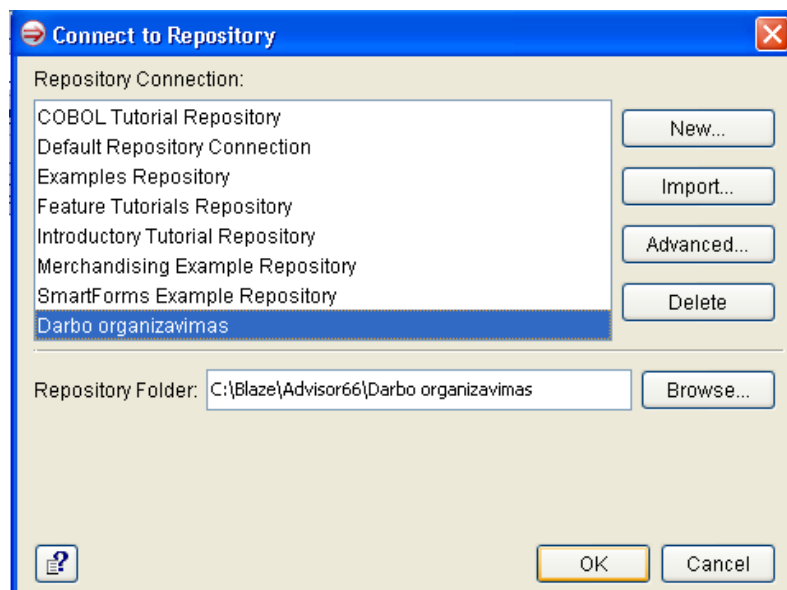
53 pav. Veiklos taisyklių integracija su WEB sistema

5.1. Veiklos taisyklių kūrimas

5.1.1. Blaze Advisor saugyklos ir projekto sukūrimas

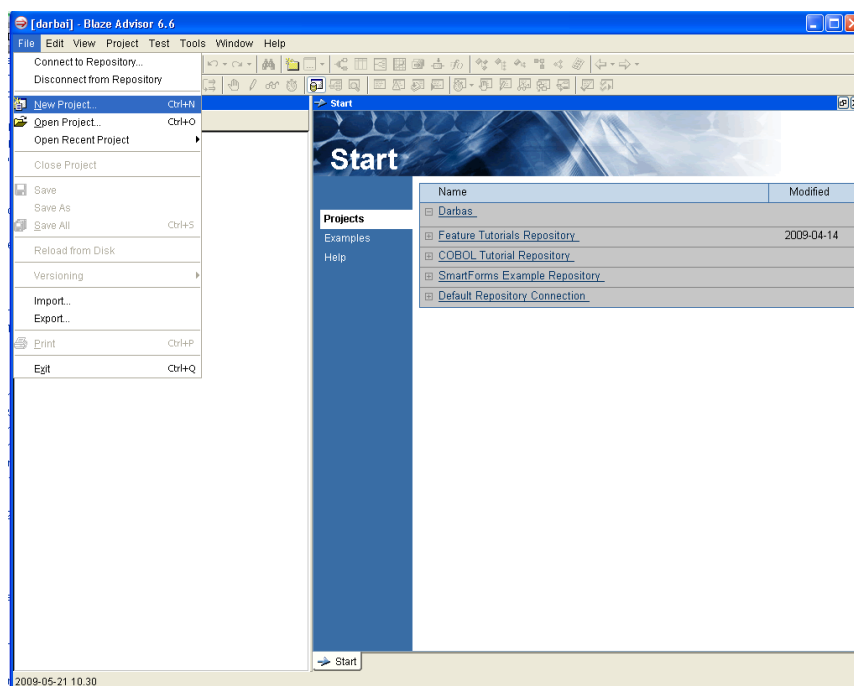
Integracija prasideda Blaze Advisor saugyklos ir projekto sukūrimu (1 etapas) (53 pav.). Blaze Advisor saugykloje saugomos visos veiklos taisyklės, klasės, taisyklių srautų informacija ir kiti taisyklių valdymo elementai. Vienoje saugykloje gali būti keli projektai.

Prisijungus prie sistemos, pasirenkame su kuria taisyklių saugykla norime dirbti (54pav.). Paspaudus mygtuką „New“ sukuriama nauja taisyklių saugykla.



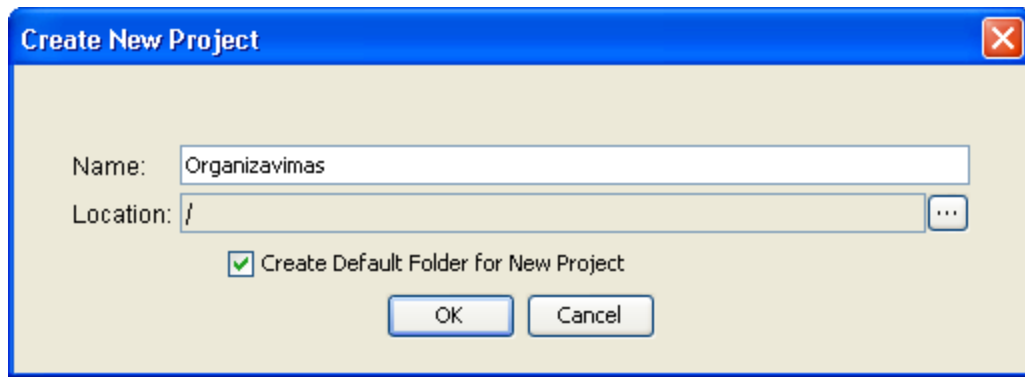
54pav. Prisijungimas prie taisyklių saugyklos

Paspaudus programoje meniu punktą „File->New Project“ sukuriamas naujas projektas. (55 pav.)



55 pav. Naujo projekto sukūrimas

Kuriant projektą, nurodomas projekto pavadinimas ir saugojimo vieta. (56 pav.)

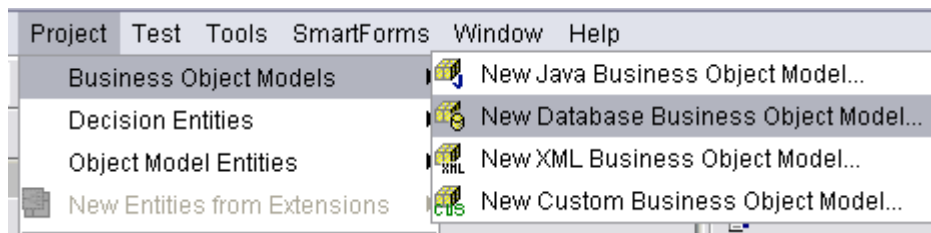


56 pav. Projekto pavadinimo priskyrimas

5.1.2. Blaze Advisor integracija su duomenų baze ir klasių sukūrimas

Antrame etape (53 pav.) su Blaze Advisor esančiu vedliu nustatomos klasės, kurios komunikuos su duomenų bazių lentelėmis per JDBC jungtį.

Priklausomai nuo duomenų bazės pirmiausiai įrankyje yra sukuriama prieiga prie duomenų bazės(57 pav.).



57 pav. Prieigos prie duomenų bazės kūrimas

Jungimuisi nurodoma kokia yra naudojama valdyklė, nuoroda į DB ir prisijungimo duomenys (58 pav.).

JDBC Driver:	sun.jdbc.odbc.JdbcOdbcDriver
Database URL:	jdbc:odbc:LocalServer
User ID:	sa
Password:	

58 pav. Prisijungimas prie DB


Sėkmingai prisijungus prie duomenų bazės, toliau seka duomenų bazės lentelių atvaizdavimas objektinėmis klasėmis. Tuomet veiklos logiką realizuojančios klasės gali

kreiptis į duomenis, esančius duomenų bazėje, ir naudoti juos sprendimams priimti. MySQL duomenų bazė taip pat saugo veiklos taisyklių parametrus, kuriuos nusistato vartotojai.

5.1.3. Pagrindinių taisyklių sukūrimas

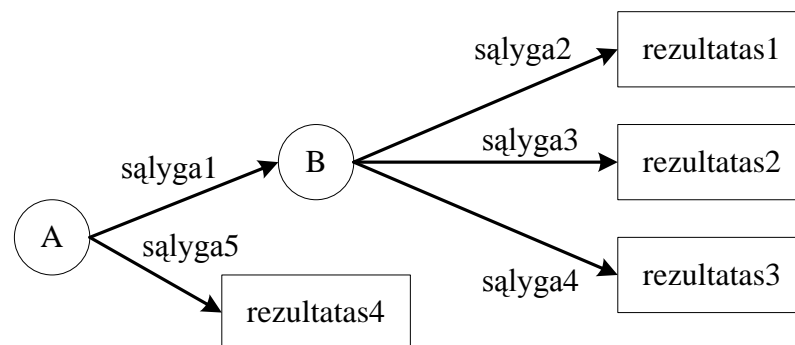
Trečiame etape (53 pav.) kuriamos reikalingos veiklos taisyklės. Tam naudojamos IF..THEN tipo taisyklės, sprendimų medžiai, sprendimų lentelės, taisyklių srantai. Taip pat sukuriamos taisyklės, kurios naudos vartotojų parametrus.

Bendra IF..THEN tipo taisyklių formavimo procedūra yra tokia:

1. Pirmiausia atidaromas taisyklių rinkinio redaktorius, projekto medyje dukart spragtelėjus ant taisyklių rinkinio.
2. Redaktoriaus lango „content“ dalyje arba įrankių juostoje spaudžiamas simbolis  naujai taisyklei sukurti. Sukurtai taisyklei reikia suteikti pavadinimą.
3. Nustatomas taisyklės prioritetas. Jeigu prioritetas nenustatytas tuomet taisyklės vykdomos iš eilės. Šis punktas nėra privalomas.
4. Nustatomas taisyklės veikimo laikas. T.y. jeigu norime, kad taisyklė būtų vykdoma tik tam tikru laiko momentu. Šis punktas nėra privalomas.
5. Žemiau taisyklės pavadinimo įvedamas taisyklės tekstas. Taisyklė turi nustatytą struktūrą. Ji turi prasidėti sąlygos žodžiu *if* ir privalo turėti raktinį žodį *then*, kuris nurodo kas turi būti įvykdyta jeigu sąlyga tenkinama. Taip pat taisyklė gali turėti ir *else* sakinį kuris įvykdomas, kai sąlyga netenkinama.
6. Kiekvienai taisyklei galima užrašyti komentarus.

Sprendimų medis formuojamas nurodant sąlygas ir grąžinamus rezultatus. Medžio mazgas gali būti sąlygos tikrinamas kintamasis arba grąžinamas rezultatas.

Tarkime, kad turime tokį sprendimų medį(59 pav.):



59 pav. Sprendimų medžio pavyzdys

Tokiu atveju veiklos taisyklės galima aprašyti taip:

```
if (A tenkina sąlyga1 and B tenkina sąlyga2) then
    return rezultatas1
if (A tenkina sąlyga1 and B tenkina sąlyga3) then
    return rezultatas2
if (A tenkina sąlyga1 and B tenkina sąlyga4) then
    return rezultatas3
if (A tenkina sąlyga5) then
    return rezultatas4
```

Sprendimų lentelėje aprašomos sąlygos ir rezultatas, grąžinamas, kai tenkinamos atitinkamos sąlygos. *Blaze Advisor* aplinkoje galima sudaryti trijų tipų lenteles:

- kai sąlygos aprašomos lentelės eilutėse;
- kai sąlygos aprašomos lentelės stulpeliuose;
- kai sąlygos aprašomos ir lentelės eilutėse, ir stulpeliuose.

Tarkime, kad turime lentelę, kurioje sąlygos aprašomos eilutėse ir stulpeliuose.

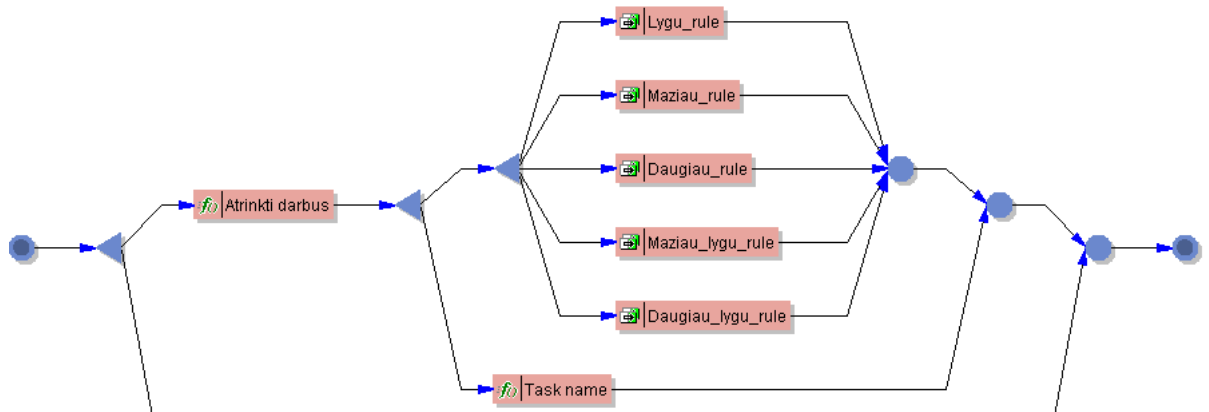
	<i>sąlyga1</i>	<i>sąlyga3</i>
	<i>sąlyga2</i>	<i>sąlyga4</i>
<i>sąlyga5</i>	rezultatas1	rezultatas3
<i>sąlyga6</i>	rezultatas2	rezultatas4

Tokiu atveju atitinkamas rezultatas gaunamas tokiu būdu:

```
if (sąlyga1 and sąlyga2 and sąlyga5) then
    return rezultatas1
if (sąlyga1 and sąlyga2 and sąlyga6) then
    return rezultatas2
if (sąlyga3 and sąlyga4 and sąlyga5) then
    return rezultatas3
if (sąlyga3 and sąlyga4 and sąlyga6) then
    return rezultatas4
```

5.1.4. Taisyklių srauto sukūrimas

Ketvirtame etape sukuriamas pagrindinis taisyklių srautas (60 pav.), kuris pagal vartotojų sukurtus parametrus ir taisyklių grąžintas reikšmes nustato, kokias funkcijas ar taisykles vykdyti toliau, kada grąžinti galutinį rezultatą. Taisyklių srautas yra grafinė veiklos proceso prezentacija. Taisyklių srautas turi prasidėti startiniu mazgu ir pasibaigti pabaigos mazgu. Tarp šių mazgų galima įdėti vykdomuosius mazgus.



60 pav. taisyklių srauto fragmentas

5.1.5. EJB diegimas į taikomųjų programų serverį

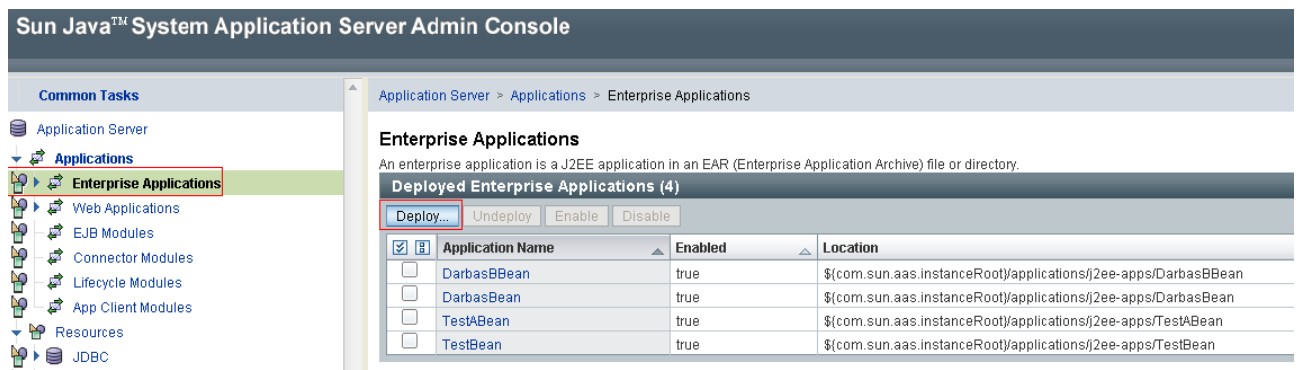
5 etapas – tai taisyklių projekto transformavimas į taisyklių paslaugas (EJB failų generavimas). Procesas susideda iš šių dalių[2]:

- Taisyklių serverio sukonfigūravimas pagal platformą į kurią diegsime taisykles.
- Taisyklių projekto nustatymas paslaugoms teikti.
- Aprašymas paslaugų įeities taškų, kurie leis klientams pasiekti paslaugas.

Įeities taškai tai yra veiklos metodai. Metodų naudojami argumentai, perduoda duomenys į taisykles, bei grąžina rezultatus.

Įvykdžius prieš tai aprašytus etapus, sugeneruojami failai, kurie diegiami į pasirinktą taikomųjų programų serverį.

Prisijungiamo prie Sun Java System Application Serverio administratoriaus konsolės. Pasirenkame „Enterprise Application“ ir spaudžiame mygtuką „Deploy...“. (61 pav.)

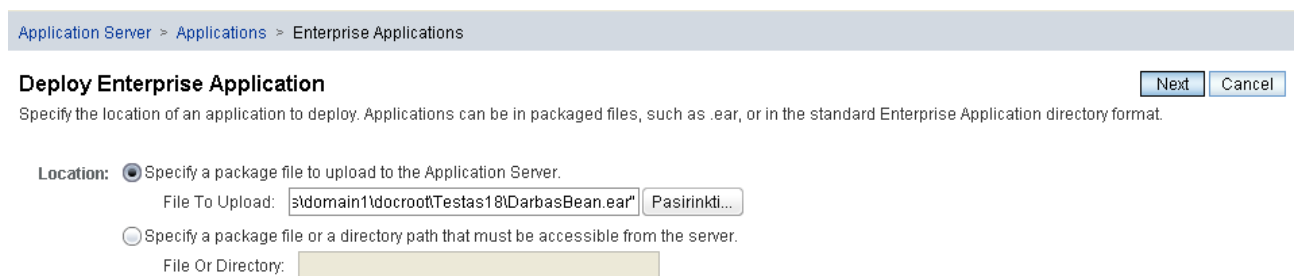


61 pav. Taisyklių paslaugų diegimas

Pasirenkame sugeneruotą EAR tipo failą. Spaudžiame „Next“ ir Sekančiame lange „OK“ (62 pav.). Norėdami patikrinti ar teisingai įkėlėme EAR failą, atveriamė naršyklėje:

http://Serveris/webservices-ejb/Paslaugu_Pavadinimas?WSDL

Jei naršyklėje matome WSDL failą, tai web paslaugos įdiegtos teisingai.



62 pav. EAR tipo failo įkėlimas į serverį

5.2. WEB programos paruošimas

Taisyklių iškvietimui naudojamas SOAP. Tai yra XML paremtas protokolas, kuris leidžia programoms keistis informacija per HTTP. Į WEB programą reiktų įtraukti SOAP reikalingas bibliotekas. 7-8 etapai (53 pav.) gali būti vykdomi lygiagrečiai su 1-5 etapais (53 pav.).

Sekančiame etape (8) (53 pav.) į programinį kodą įterpiamos eilutės, kurios iškviečia veiklos taisykles, apdoroja gražintą rezultatą ir padeda vartotojams valdyti savo taisyklių parametrus.

Sukūrus visą aplinką vartotojui, prijungus taisyklių paslaugas, vartotojas gali pradėti dirbti su sistema. Tai yra ne tik atlikinėti savo veiklą, bet ir kurti veiklos taisykles, nustatant tam tikrus parametrus.

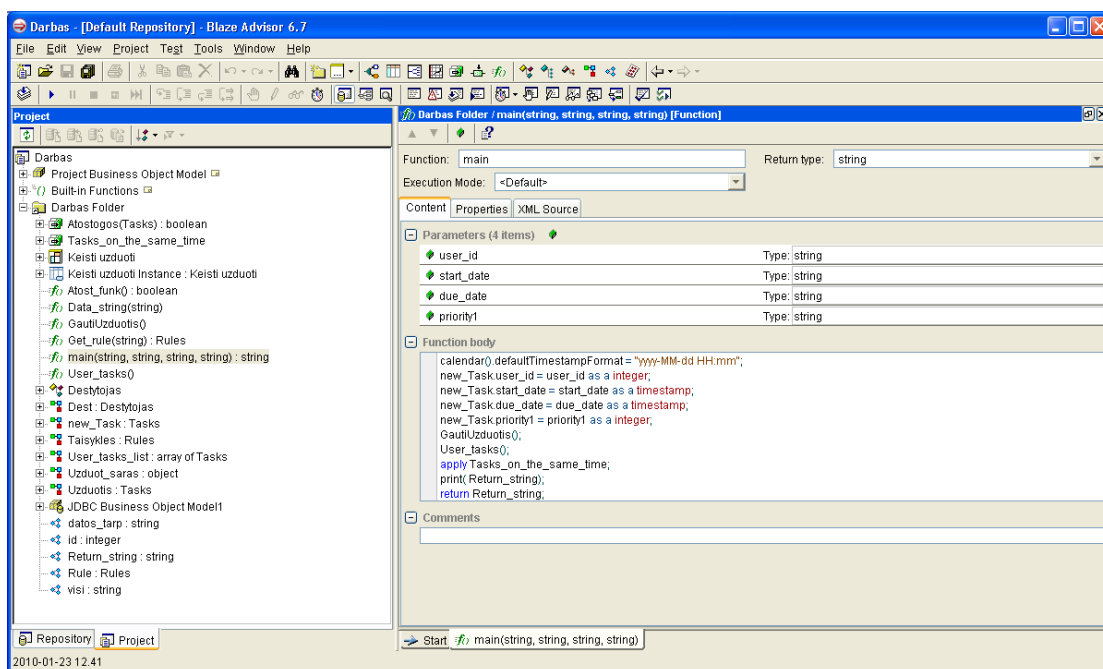
Pasikeitus tam tikroms veikloms atliekami 10,5,6 etapai, pavaizduoti 53 paveikslėlyje.

6. Sistemos realizacija

Šiame skyriuje apžvelgiama sukurta sistema: pagrindiniai vartotojo langai ir funkcijos, kurias galima atlikti sistemoje. Technologijų analizė įgalino sudaryti konkrečios sistemos projektą ir pagal jį sukurti sistemą, kurios struktūra yra labiau tinkama greitesniam prisitaikymui prie aplinkybių ir modifikavimui, nei įprastu būdu sukurtos sistemos.

6.1. Blaze Advisor Builder posistemio veikimo aprašymas

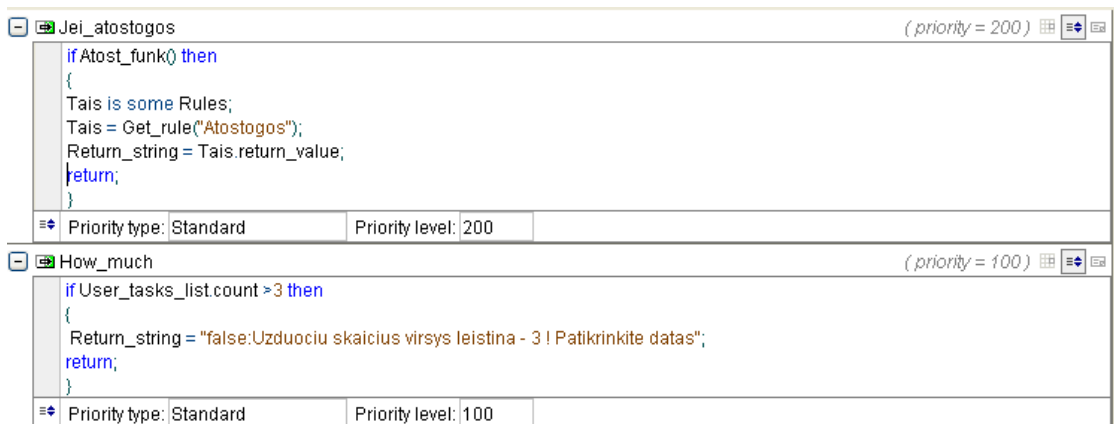
Sistemos vartotojas prisijungęs prie sistemos ir pasirinkus projektą atidaromas projekto darbo langas, kuriame kuriami nauji objektai, taisyklės, funkcijos ir kiti elementai. (63 pav.).



63 pav. Blaze Advisor pagrindinis langas

Langą sudaro trys dalys: Menių ir įrankių juostos (viršuje), projekto elementų sąrašas (kairėje) ir sritis (dešinėje), kurioje dirbama su veiklos taisyklių projekto elementais.

64 paveikslėlyje pateikiama veiklos taisyklių pavyzdys. Taisyklės galima grupuoti į taisyklių rinkinius. Taisyklėms gali būti priskiriami prioritetai. Pirmiausia vykdomos taisyklės turinčios didesnę prioritetą. Taisyklės kaip ir funkcijos turi galimybę gražinti įvairaus tipo reikšmes (string, boolean, int ir pan.). Taisyklės struktūra turi būti: „IF ... THEN ...“ arba „IF... THEN... ELSE...“.



64 pav. Blaze Advisor veiklos taisyklių pavyzdys.

Blaze Advisor turi galimybę kurti sprendimų medžius, sprendimų lenteles, taisyklių srautus, objektus ir kintamuosius. Detalesnė informacija apie šių elementų valdymą ir panaudojimą pateikiama prieduose.

Sukurtas projektas turi būti įdiegiamas taisyklių vykdymo serveryje. Greitasis diegimo vedlys iškviečiamas meniu juostoje. Sugeneruoti failai patalpinami taisyklių vykdymo serveryje ir teikia taisyklių paslaugas. Patalpinimo procesas aprašomas prieduose.

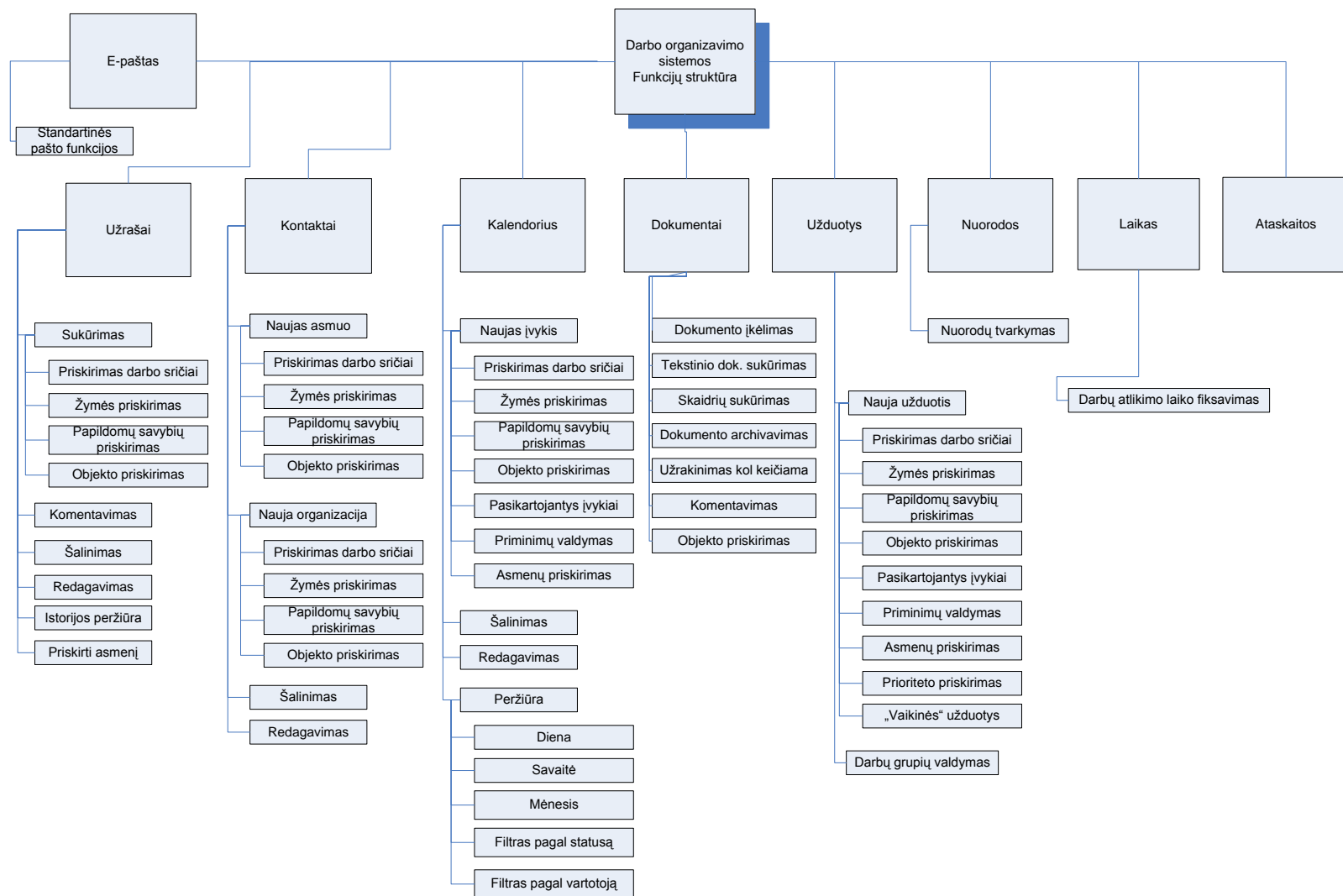
6.2. Darbo organizavimo posistemio veikimo aprašymas

Registruotas sistemos vartotojas prisijungęs prie sistemos, patenka į pagrindinį langą, kuris pavaizduotas 65 paveikslėlyje.



65 pav. Darbų organizavimo posistemio pagrindinis langas

Darbų organizavimo posistemio funkcijų struktūra pateikiama 66 pav. Kiekviena funkcijų grupė yra detalizuojama hierarchiškai, pavaizduojant smulkesnes funkcijas.



66 pav. Darbo organizavimo posistemio funkcijų struktūra.

Sistemoje funkcijos valdomos meniu punktais. Meniu struktūra pavaizduota 67 pav.



67 pav. Darbo organizavimo posistemio meniu struktūra.

Visos sistemos funkcijos aprašomos vartotojo vadove ir pateikiama prieduose. Šioje dalyje detaliau bus aprašoma tik veiklos taisyklių redagavimo sritis (68 pav.).

Taisyklės pavadinimas	Taisyklės objektas	Objekto reikšmė	Taisyklės grąžinama reikšmė
max_tasks	task.count	6	Užduočių skaičius viršys leistinai Patikrinkite datas
Atostogos	task.name	Atostogos	Darbuotojas tuo metu atostogauja
Konferencija Helsinkyje	task.name	Konferencija_Helsinkyje	Darbuotojas dirba Helsinkyje
Pranešimas	task.name	Pranešimas	Užsiėmęs
Teismas	task.name	Teismas	Darbuotojas teisiiasi teisme

68 pav. Darbo organizavimo posistemio veiklos taisyklių redagavimo sritis.

Vartotojo kuriama taisyklė susideda iš tokių parametru:

- Antraštė
- Taisyklės objektas (tai gali būti užduoties ar įvykio pavadinimas, skaičius kiek vyksta tuo metu, trukmė, atsakingas asmuo, kūrėjas, pradžios ir pabaigos datos.)
- Palyginimo simbolis (<, >, =, <=, >=)
- Objekto reikšmė (reikšmė su kuria lyginama. Pvz. Galime įrašyti datą, su kuria lygins užduočių pradžias.)
- Grąžinimo tipas. Galimos reikšmės:
 - Taisyklė neleidžia atlikti veiksmo: sukurti užduoties, pakeisti įvykio datos ir pan. Išvedamas klaidos pranešimas.
 - Taisyklė leidžia vykdyti veiksmus.
 - Taisyklė tik informuoja vartotojus ir išveda informacinį pranešimą.
- Taisyklės grąžinama reikšmė. Tai informacinis pranešimas parodomas vartotojui, kai suveikia atitinkama taisyklė.

Taisyklių redagavimo forma pateikiama 69 pav.

Redaguojama taisyklė

Antraštė: *

Taisyklės sąlygos dalis:

Taisyklės objektas:

Palyginimo simbolis:

Objekto reikšmė:

Taisyklės veiklos dalis:

Gražinimo tipas:

Taisyklės gražinama reikšmė:

69 pav. Darbo organizavimo posistemio veiklos taisyklių redagavimo forma.

Tarkime sukuriame taisyklę, kuri neleidžia priskirti užduočių, kai yra užduotis su pavadinimu „Atostogos“.

Bandant įvesti naują užduotį pasirenkame datą, kuri yra tuo metu, kai darbuotojas atostogauja. Sistema iškviečia taisyklę (70 pav.):

Antraštė: *

Atostogos

Taisyklės sąlygos dalis:

Taisyklės objektas:

Palyginimo simbolis:

Objekto reikšmė:

Taisyklės veiklos dalis:

Gražinimo tipas:

Taisyklės gražinama reikšmė:

Darbuotojas tuo metu atostogauja

70 pav. Veiklos taisyklė „Atostogos“

Sistema patikrina, ar darbuotojas tuo metu atostogauja. Gražinama eilutė „false:Darbuotojas tuo metu atostogauja“. Kadangi gražinamos eilutės pirmoje dalyje yra „false“, internetinė sistema užduoties neišsaugo. Išvedamas pranešimas, kad įvyko klaida ir paaiškinimas kodėl: „Darbuotojas tuo metu atostogauja“ (71 pav.).

Error: Demesio: Darbuotojas tuo metu atostogauja

71 pav. Internetinės sistemos išvedamas pranešimas, kuris gautas iš veiklos taisyklių.

6.3. Testavimo modelis bei duomenys, kontrolinis pavyzdys

Testavimas būtinas kiekvienos sistemos patvirtinimo procesas. Testavimo tikslas yra patikrinti ar realizuota sistema teisingai vykdo visus panaudojimo atvejus. Norint pilnai ištestuoti sistemą reikia patikrinti kiekvieną jos elementą.

Kadangi sistema kuriama nuolat ją tobulinant, tai sistema turi būti testuojama visą kūrimo etapą. Sistemos testavimui yra sudaromas testavimo planas, kuriame atsispindi, kokius veiksmus reikia atlikti testuojant sistemą, taip pat, kokia turi būti sistemos reakcija į konkrečius veiksmus. Testavimo duomenys turi atskleisti silpnąsias sistemos vietas.


Atlikus testus ir aptikus sistemos trūkumus atliekamas sistemos tobulinimas. Toks iteracinis procesas atliekamas tol, kol nebelieka klaidų sistemoje.

6.3.1. Prisijungimas

Prisijungimas turi užtikrinti vartotojo duomenų saugumą, vartotojų skirstymą pagal grupes. Prisijungimo testavimo scenarijus pateiktas 22 lentelėje.

22 lentelė Prisijungimo testavimo scenarijus

Nr.	Įėjimas	Laukiamas rezultatas
6.3.1-1	Neįvedamas vartotojo vardas ir/arba slaptažodis ir bandoma prisijungti	Atveriamas prisijungimo langas, išvedamas pranešimas, kad įvyko klaida
6.3.1-2	Įvedamas neteisingas vartotojo vardas ir/arba slaptažodis ir bandoma prisijungti	Atveriamas prisijungimo langas, išvedamas pranešimas, kad įvyko klaida (72 pav.)
6.3.1-3	Įvedamas teisingas vartotojo vardas ir/arba slaptažodis ir bandoma prisijungti	Atveriamas pagrindinis langas
6.3.1-4	Įvedamas teisingi vartotojo vardai ir/arba slaptažodžiai skirtingų grupių vartotojų ir bandoma prisijungti	Atveriamas pagrindinis langas. Vartotojai turi galimybes dirbti pagal savo grupės tipą.



The screenshot shows a login interface titled "Prisijungti". A prominent red error message box states: "Nepavyko prisijungti. Prašome patikrinti prisijungimo duomenis ir pabandyti dar kartą." Below the message, there are two input fields: "Prisijungimo vardas:" containing the letter "a" and "Slaptažodis:". A checkbox labeled "Prisiminti mane 14 dienų" is unchecked. At the bottom, there is a "Prisijungti" button, a link "(Pamiršot slaptažodį?)", and a "Pasirinktys" link.

72 pav. Prisijungimo langas, kai įvedami blogi duomenys

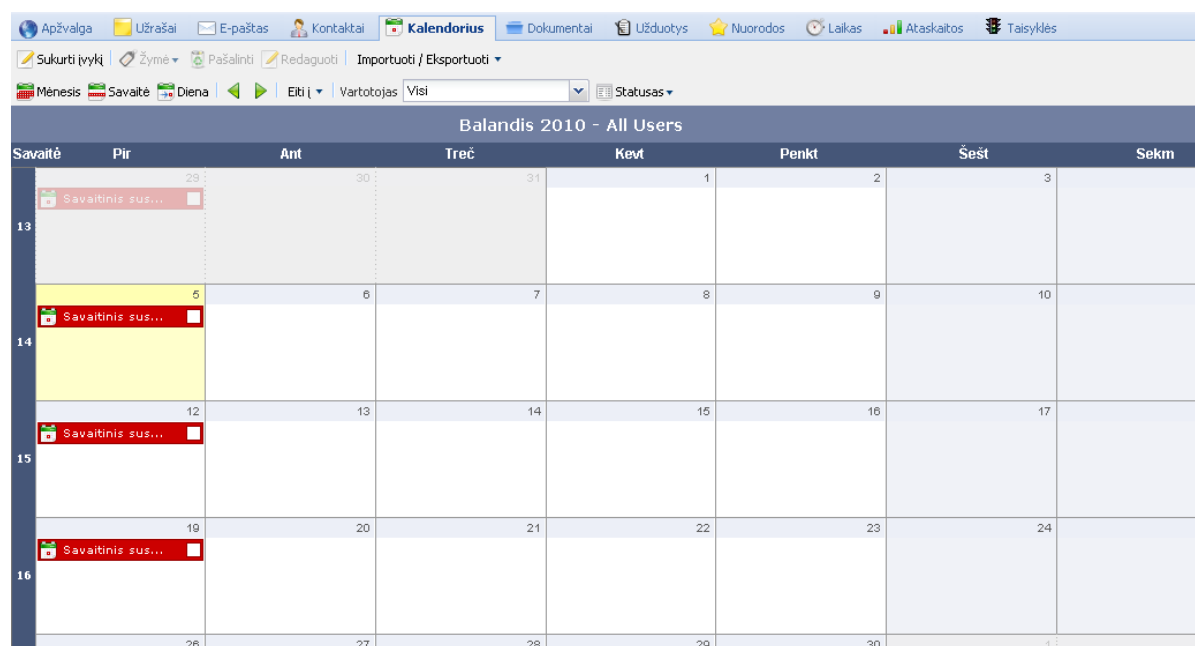
Prisijungimo testavimas įvykdytas sėkmingai.

6.3.2. Meniu

Meniu suteikia galimybę vartotojui atverti sistemos langus skirtus įvairioms funkcijoms vykdyti. Testavimas turi patikrinti, kad visi meniu punktai veikia teisingai, tai yra pasirinkus atitinkamą meniu punktą, atveriamas jį atitinkantis sistemos langas. Meniu testavimo scenarijus pateiktas 23 lentelėje.

23 lentelė Meniu testavimo scenarijus

Nr.	Įėjimas	Laukiamas rezultatas
6.3.2-1	Pasirenkamas pradinis meniu punktas	Atveriamas pagrindinis langas
6.3.2-2	Pasirenkamas kalendoriaus meniu punktas	Atveriamas kalendoriaus langas (73pav.)
6.3.2-3	Pasirenkamas užrašų meniu punktas	Atveriamas užrašų langas
6.3.2-4	Pasirenkamas e-pašto meniu punktas	Atveriamas e-pašto langas
6.3.2-5	Pasirenkamas kontaktų meniu punktas	Atveriamas kontaktų langas
6.3.2-6	Pasirenkamas dokumentų meniu punktas	Atveriamas dokumentų langas
6.3.2-7	Pasirenkamas užduočių meniu punktas	Atveriamas užduočių langas
6.3.2-8	Pasirenkamas nuorodų meniu punktas	Atveriamas nuorodų langas
6.3.2-9	Pasirenkamas laiko meniu punktas	Atveriamas laiko langas
6.3.2-10	Pasirenkamas taisyklių meniu punktas	Atveriamas taisyklių langas
6.3.2-11	Pasirenkamas valdymo meniu punktas	Atveriamas valdymo langas



73 pav. Kalendoriaus langas, kai pasirinktas meniu punktas „Kalendorius“

Meniu testavimas įvykdytas sėkmingai.

6.3.3. Taisyklių valdymas

Taisyklių valdymas turi užtikrinti vartotojo taisyklių valdymą. Taisyklių valdymo testavimo scenarijus pateiktas 24 lentelėje.

24 lentelė Taisyklių valdymo testavimo scenarijus

Nr.	Įėjimas	Laukiamas rezultatas
6.3.3-1	Neįvedamas taisyklės pavadinimas	Atveriamas taisyklės forma, išvedamas klaidos pranešimas.
6.3.3-2	Neįvedama objekto reikšmė	Atveriamas taisyklės forma, išvedamas klaidos pranešimas.
6.3.3-3	Neįvedama taisyklės gražinama reikšmė.	Atveriamas taisyklės forma, išvedamas klaidos pranešimas.
6.3.3-4	Taisyklės redagavimo/kūrimo forma užpildoma teisingai.	Taisyklės duomenys išsaugomi. Atveriamas taisyklių sąrašo langas.

Taisyklių valdymo testavimas įvykdytas sėkmingai.

6.3.4. Taisyklių veikimas

Testavimas turi patikrinti, ar vartotojų sukurtos taisyklės veikia teisingai. Testuojamos taisyklės su skirtingais parametrais. Testavimo scenarijus pateiktas 25 lentelėje.

25 lentelė Taisyklių veikimo testavimo scenarijus

Nr.	Įėjimas	Laukiamas rezultatas
6.3.4-1	Sukuriama taisyklė, kuri neleidžia kurti naujų užduočių, kai yra užduotis su pavadinimu „Testas“. Kuriama nauja užduotis, kai egzistuoja užduotis „Testas“.	Išvedamas klaidos pranešimas. Nauja užduotis nesukuriama.
6.3.4-2	Sukuriama taisyklė, kuri neleidžia kurti naujų užduočių, kai yra užduotis su pavadinimu „Testas“. Kuriama nauja užduotis, kai neegzistuoja užduotis „Testas“.	Sukuriama nauja užduotis.
6.3.4-3	Sukuriama taisyklė, kuri informuoja, kai yra užduotis su pavadinimu „Testas“. Kuriama nauja užduotis, kai egzistuoja užduotis „Testas“.	Sukuriama nauja užduotis. Išvedamas informacinis pranešimas.
6.3.4-4	Sukuriama taisyklė, kuri neleidžia kurti naujų užduočių, kai užduočių skaičius vienu metu daugiau nei 10. Kuriama nauja užduotis, kai užduočių skaičius vienu metu daugiau nei 10.	Išvedamas klaidos pranešimas. Nauja užduotis nesukuriama.
6.3.4-5	Sukuriama taisyklė, kuri informuoja, kai užduočių skaičius vienu metu daugiau nei 10. Kuriama nauja užduotis, kai užduočių skaičius vienu metu daugiau nei 10.	Sukuriama nauja užduotis. Išvedamas informacinis pranešimas.
6.3.4-6	Sukuriama taisyklė, kuri informuoja, kai užduočių skaičius vienu metu lygus 10. Kuriama nauja užduotis, kai užduočių skaičius	Sukuriama nauja užduotis. Išvedamas informacinis pranešimas.

	vienu metu daugiau lygus 10.	
6.3.4-7	Sukuriama taisyklė, kuri neleidžia kurti naujų užduočių, jei užduoties pabaigos data vėlesnė nei 2010-09-01. Kuriama nauja užduotis, kurios pabaigos data vėlesnė nei 2010-09-01.	Išvedamas klaidos pranešimas. Nauja užduotis nesukuriama.
6.3.4-8	Sukuriama taisyklė, kuri neleidžia kurti naujų užduočių, jei darbuotojo atliekančio užduotį pavardė „Testauskas“. Kuriama nauja užduotis, kurią atliks darbuotojas „Testauskas“.	Išvedamas klaidos pranešimas. Nauja užduotis nesukuriama.
6.3.4-8	Sukuriama taisyklė, kuri informuoja, jei darbuotojo atliekančio užduotį pavardė „Testauskas“. Kuriama nauja užduotis, kurią atliks darbuotojas „Testauskas“.	Sukuriama nauja užduotis. Išvedamas informacinis pranešimas.

Taisyklių veikimo testavimas įvykdytas sėkmingai.

6.4. Sukurtos sistemos realizacijos apibendrinimas

Informacinė sistema sukurta, siekiant pagerinti informacijos sistemų katedros informacijos valdymą, katedros darbų koordinavimą. Sistemos privalumai - visa reikalinga darbui informacija prieinama naudojantis internetu ir bet kurio metu. Taip pat visa veikla yra kontroliuojama pačių darbuotojų sukurtomis veiklos taisyklėmis. Darbuotojai turi galimybę tarpusavyje bendrauti pranešimais ir taip koordinuoti bendrus darbus. Sistemos vartotojas gali matyti, kokie darbai jam paskirti, bei kokius darbus jis pats yra susiplanavęs. Darbai filtruojami pagal svarbą ir laiką.

Darbų paskirstymo ir koordinavimo uždaviniams spręsti sukurta darbo organizavimo informacinė sistema praturtinta veiklos taisyklių posistemiu. Atlikta integracija su Blaze Advisor įrankiu padeda vartotojams valdyti veiklos taisykles. Tai suteikia galimybę priimti efektyvesnius sprendimus, vykdant katedros veiklą.

6.5. Taikymo rekomendacijos

Šiuolaikinėse informacinėse sistemose veiklos taisyklių valdymo įrankiai tampa būtini. Jie leidžia kaupti ir modifikuoti taisykles. Veiklos taisyklės, kurių valdymui naudojami taisyklių valdymo įrankiai, turi būti:

- formalizuojamos;
- užrašomos įrankyje naudojama kalba;

- saugojamos ir palaikomos.

Darbe pagrindinis akcentas yra nukreiptas į VTVS tarpusavio sąveiką su taikomąja sistema. Pateikiama strategija, kaip organizuojama integracija, kuriamos ir vykdomos veiklos taisyklės. Aprašomi keli veiklos taisyklių paruošimo būdai išvystomi VTVS pagalba. Tai pat pateikiama kaip eilinis vartotojas gali susikurti savo taisykles naudojant šablonus taikomojoje programoje.

Pasirinktoje srityje veiklos taisyklės keičiasi dažnai, darbuotojų suformuluotos taisyklės turi būti kuo greičiau įtrauktos į informacinę sistemą ir vykdomos veikloje. Naudojant siūlomą metodiką, natūralia kalba suformuluotos veiklos taisyklės transformuojamos į vykdomąsias taisykles. Sistemoje realizuotos veiklos taisyklės daro įtaką organizacijos darbui ir veiklos koordinavimui.

Informacinę sistemą galima taip pat pritaikyti kitose srityse, kuriuose reikalingas darbų koordinavimas. Veiklos taisyklių naudojimas palengvina sistemos pritaikymą įvairiuose srityse.

Sistema planuojama toliau tobulinti. Bus siekiama išplėsti veiklos taisyklių veikimą. Naudoti jas ne tik užduočių ir įvykių valdymui, bet ir valdant dokumentus, bei pritaikyti veiklos taisykles diskusijų modulyje.

Tolesniuose tyrimuose ir darbuose bus siekiama suteikti galimybę eiliniams vartotojams kurti savo sprendimų lenteles ir įverčių modelius.

7. Išvados

1. Atlikus veiklos taisyklių (VT) valdymo sistemų (VTVS) tyrimą, nustatyta, kad šių sistemų palaikomi VT valdymo mechanizmai užtikrina efektyvų veiklos taisyklių, realizuojančių veiklos sprendimų priėmimo logiką, atskyrimą nuo programinio kodo. Veiklos taisyklių valdymo posistemio integravimas informacinėje sistemoje padeda užtikrinti lankstų ir greitą šios sistemos pritaikymą prie pasikeitusių veiklos poreikių.
2. Detaliai išanalizavus VTVS (Blaze Advisor) architektūrą, šios sistemos naudojamą specifinę VT užrašymo kalbą bei VT kūrimo komponentus, nustatyta, kad įrankyje yra pilnai įgyvendintos visos VTVS sistemų palaikomos VT struktūros (taisyklių rinkiniai, sprendimų medžiai, lentelės, srautai ir kt.) užtikrinančios galimybę kurti įvairaus sudėtingumo taisykles, kontroliuojančias organizacijos veiklos procesus.
3. Darbo metu atliktos probleminės srities analizės (analizuota KTU Informacijos sistemų katedros veikla) pagrindu nustatytas poreikis kurti katedros darbo organizavimo informacinę sistemą, praturtiną VT valdymo posistemiu; suformuoti tokio tipo sistemai keliami reikalavimai.
4. Įgytų žinių apie VT valdymo sistemas ir nustatytų vartotojo reikalavimų pagrindu buvo suprojektuota, realizuota ir ištestuota KTU Informacijos sistemų katedros darbo organizavimo sistema, praturtinta veiklos taisyklių posistemiu.
5. Siekiant pritaikyti veiklos taisyklių posistemiu praturtintų informacinių sistemų naudojimą ir eiliniams vartotojams, buvo pasiūlyta veiklos taisykles kurti, naudojant apibrėžtos struktūros šablonus.
6. Veiklos taisyklių posistemiu praturtintos sistemos kūrimo procesas buvo dokumentuotas, ir šios dokumentacijos pagrindu buvo sukurta ir aprašyta metodika veiklos taisyklių valdymo sistemos Blaze Advisor integracijai su internetine informacine sistema atlikti. Taip pat pateikti nurodymai, veiklos taisyklių valdymo sąsajos kūrimui – ši sąsaja įgalina vartotoją kurti naujas veiklos taisykles ir valdyti šių taisyklių parametrus.
7. Atlikto darbo pagrindu suformuluoti tokie pagrindiniai teiginiai:
 - Veiklos taisyklių sluoksnio atskyrimas informacinėje sistemoje suteikia galimybę vartotojams, neturintiems techninių žinių, peržiūrėti, kurti, keisti, šalinti veiklos taisykles (t.y. valdyti kompiuterizuotą veiklos logiką) be įsikišimo į programinį kodą;

- Veiklos taisyklėmis grindžiama sistema pasižymi didesniu lankstumu, plečiamumu ir pakartotiniu panaudojimu;
 - Sistemos realizacija parodė, kad pasiūlytą metodiką tikslinga taikyti, kuriant veiklos taisyklėmis grindžiamas informacines sistemas.
8. Sukurtos sistemos testavimas patvirtino, kad programinė įranga veikia gerai, ir buvo realizuoti visi vartotojo reikalavimų specifikavimo etape nustatyti funkciniai reikalavimai.

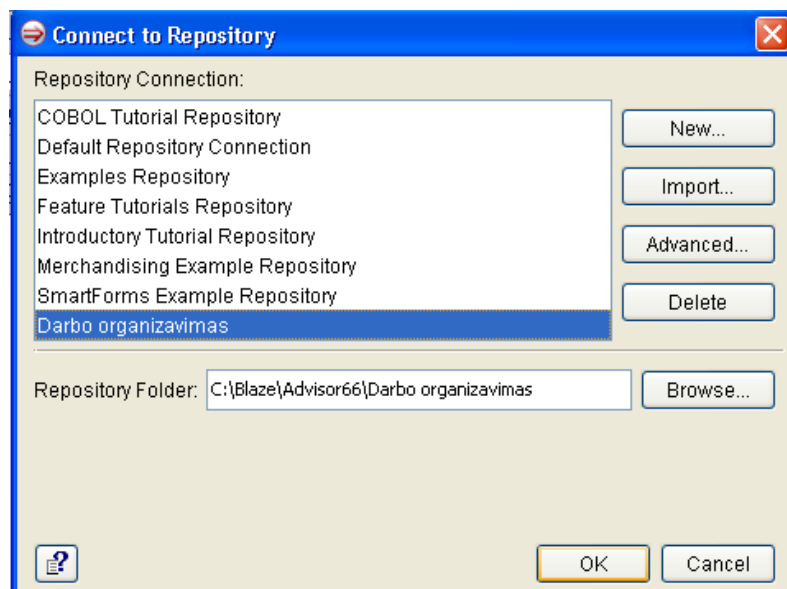
8. Literatūra

1. Babis Theodoulis, Anthony Youdeowey, 2000 Business Rules: Towards Effective Information Systems Development 313-321
2. Blaze Advisor documentation. <http://www.fairisaac.com>.
3. Chisholm, M. Business Rules Evangelist: Business Rules and black boxes. DMReview.com, Prieiga per internetą:
<http://www.dmreview.com/article_sub.cfm?articleId=8163>.
4. David C. Hay. Managing Business by Rules Prieiga per internetą:
<http://www.essentialstrategies.com/documents/bruleseco99.pdf>
5. Defining Business Rules: What Are They Really? (3rd ed.). Business Rules Group, 2000, URL: <http://www.BusinesRulesGroup.org>.
6. <http://www.paragoncorporation.com/ArticleDetail.aspx?ArticleID=15>
7. ILOG Rules. <http://www.ilog.co.uk/products/rules/>
8. Infrex. <http://www.tcs.com>
9. Yasu Technologies. <http://www.yasutech.com/products/index.htm>
10. John Medike, Feng-Wei Chen, Margie Mago. Creating an Intelligent and Flexible Solution with BPM, Business Rules, and Business Intelligence.
<http://www.ibm.com/developerworks/db2/library/techarticle/0310medicke/0310medicke.html>
11. Liudas Motiejūnas, Rimantas Butleris. Veiklos taisyklių manipuliavimo mechanizmų analizė. Informacinės technologijos 2003, KTU, ISBN 9955-09-335-8.
12. R. G. Ross. Principles of the Business Rules Approach. Addison-Wesley, 2003.
13. Ronald G. Ross, „Principles of the business rule approach“, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2003, ISBN 0201788934
14. Ronald G. Ross, "The Point of Knowledge," Business Rules Journal, Vol. 11, No.1 (Jan. 2010), Prieiga per internetą:
<<http://www.BRCommunity.com/a2010/b515.html> >
15. Sun Java System Application Server Platform Edition 8.2 Administration Guide
<http://docs.sun.com/app/docs/doc/819-4712>
16. Versta Inc.
http://www.versata.com/index2.php?option=com_content&task=view&id=124

9. Priedai

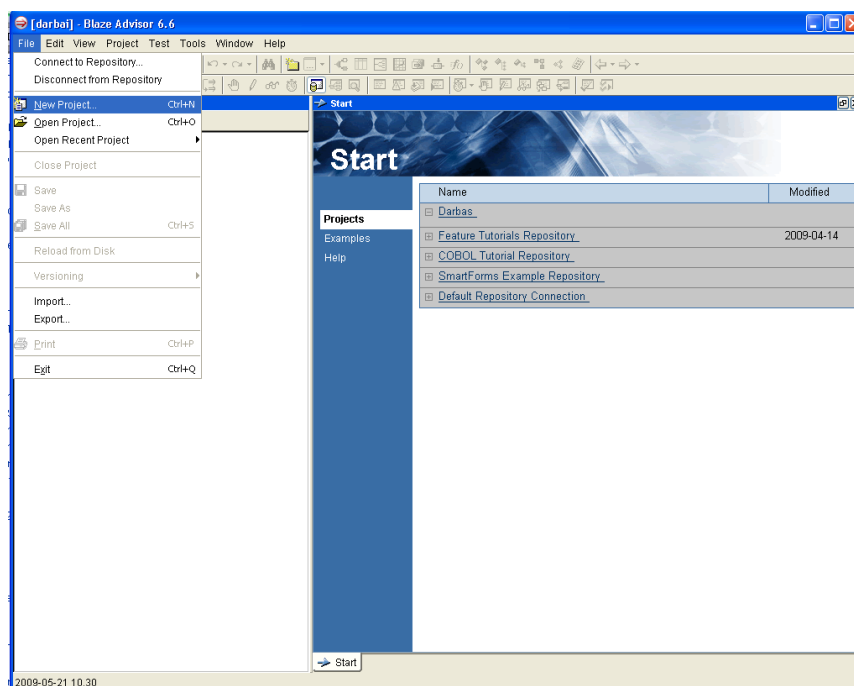
9.1. Blaze Advisor taisyklių kūrimo vadovas

Prisijungus prie sistemos, pasirenkame su kuria taisyklių saugykla norime dirbti (63 pav.). Paspaudus mygtuką „New“ sukuriama nauja taisyklių saugykla.



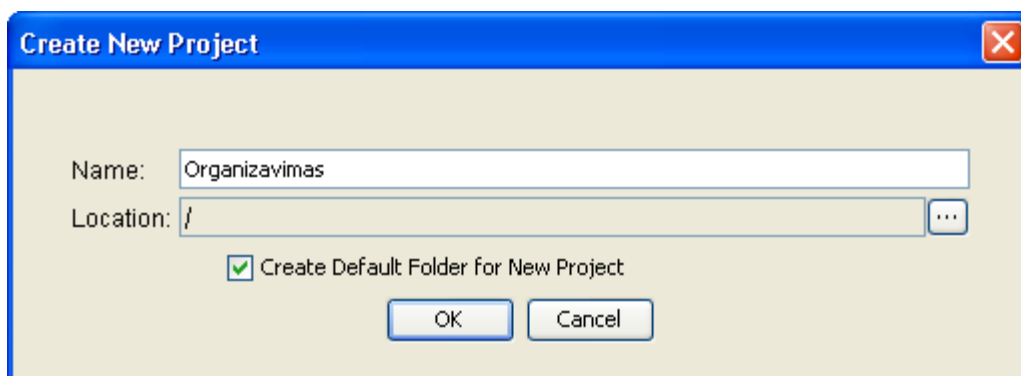
74pav. Prisijungimas prie taisyklių saugyklos

Paspaudus programoje meniu punktą „File->New Project“ sukuriamas naujas projektas. (64 pav.)



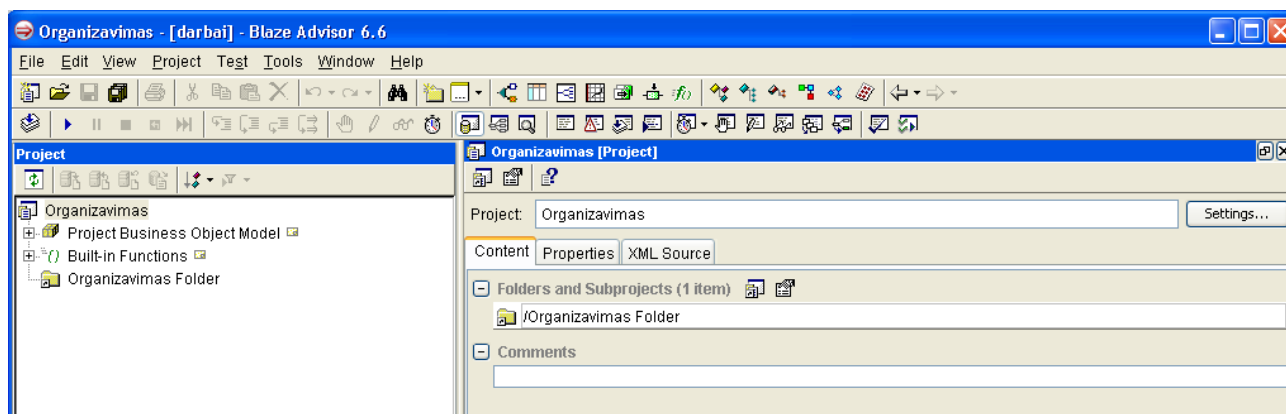
75 pav. Naujo projekto sukūrimas

Kuriant projektą, nurodomas projekto pavadinimas ir saugojimo vieta. (65 pav.)



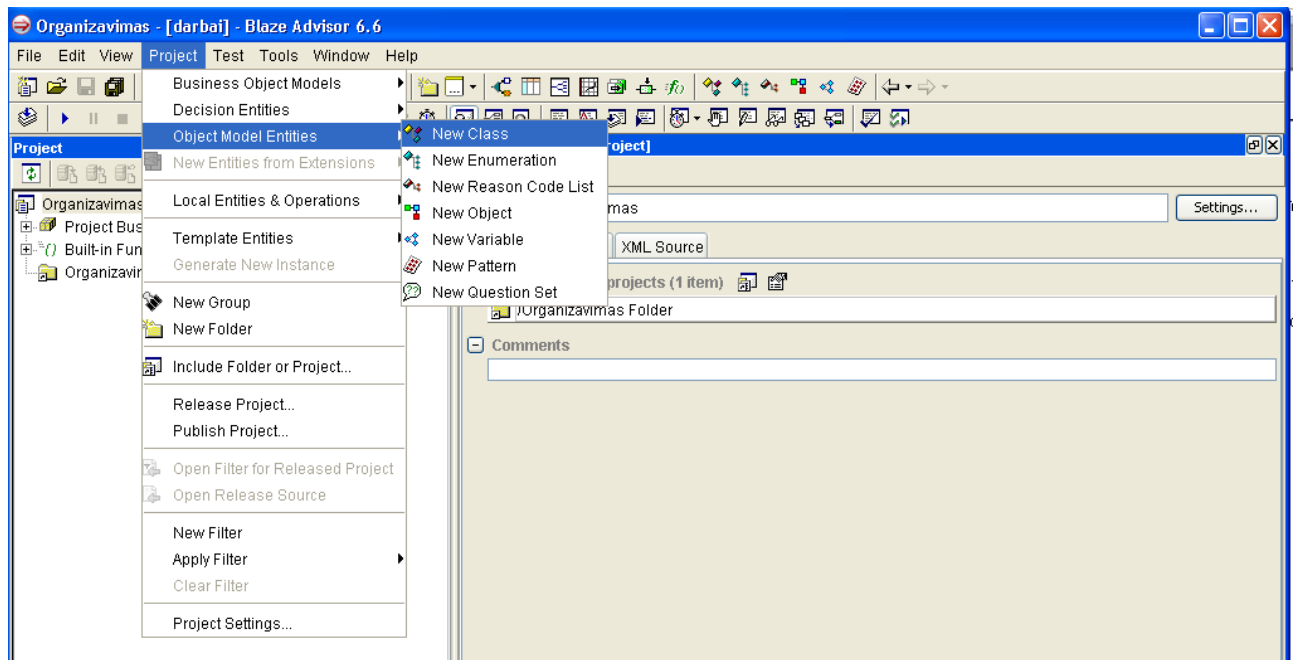
76 pav. Projekto pavadinimo priskyrimas

Sukūrus projektą atidaromas projekto darbo langas, kuriame kuriami nauji objektai, taisyklės, funkcijos ir kiti elementai. (66 pav.)



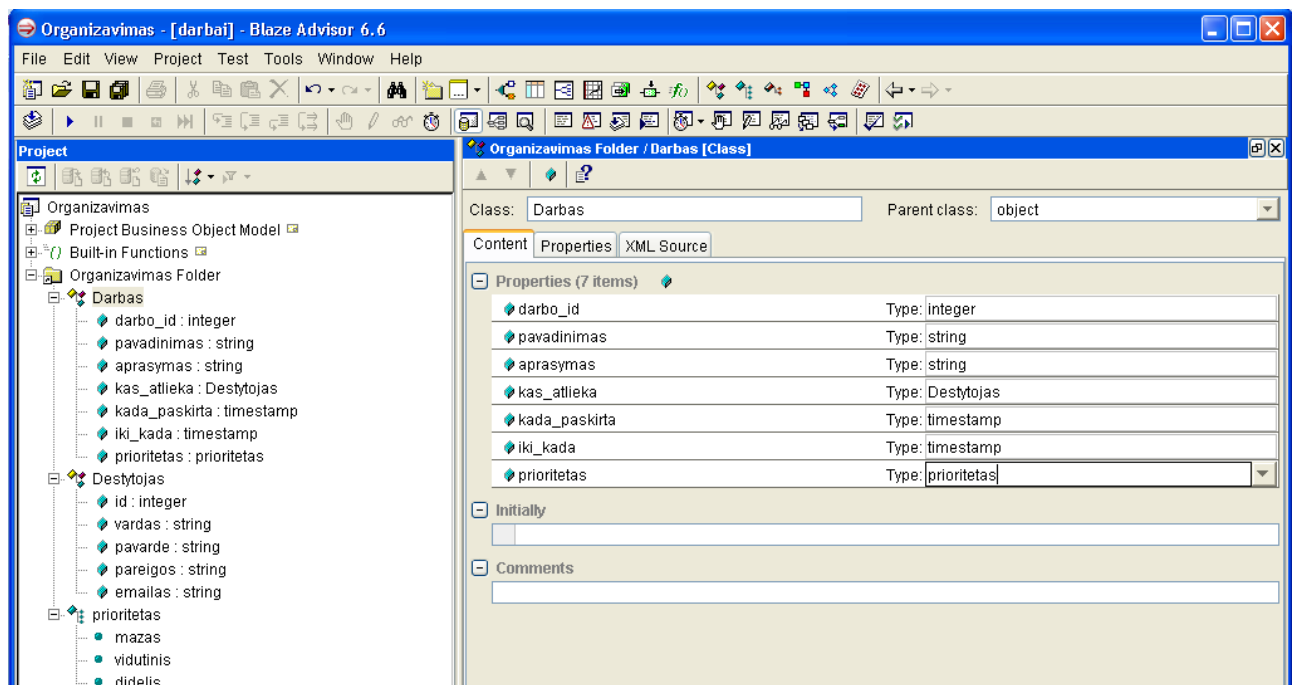
77 pav. Projekto darbo langas

Paspaudus meniu punktą „Project->Object Model Entities->New Class“ sukuriama nauja klasė.(67 pav.)



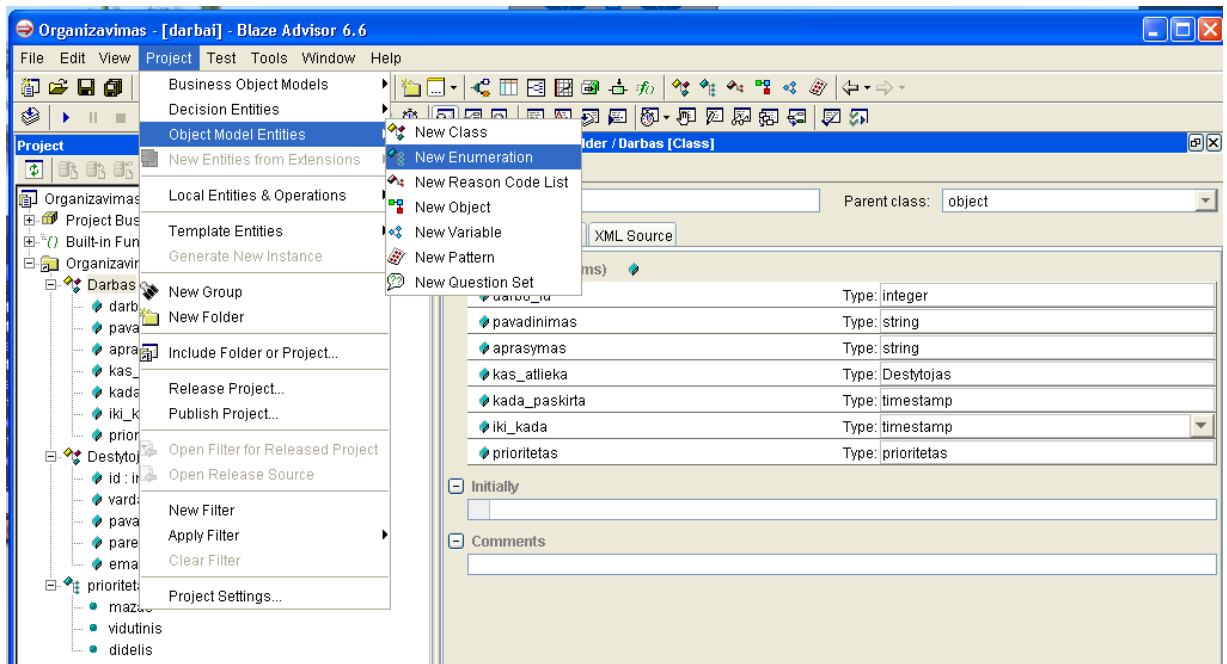
78 pav. Naujos klasės sudarymas

Sukūrus naują, nurodome jos atributus ir atributų tipus. Atributai gali būti: Objektas, Integer, String, Date, Timestamp ir kitų duomenų tipų. Tai pat galima priskirti sąrašo (Enumeration) tipo kintamąjį, kuriam yra priskiriamas reikšmių sąrašas. (68 pav.)



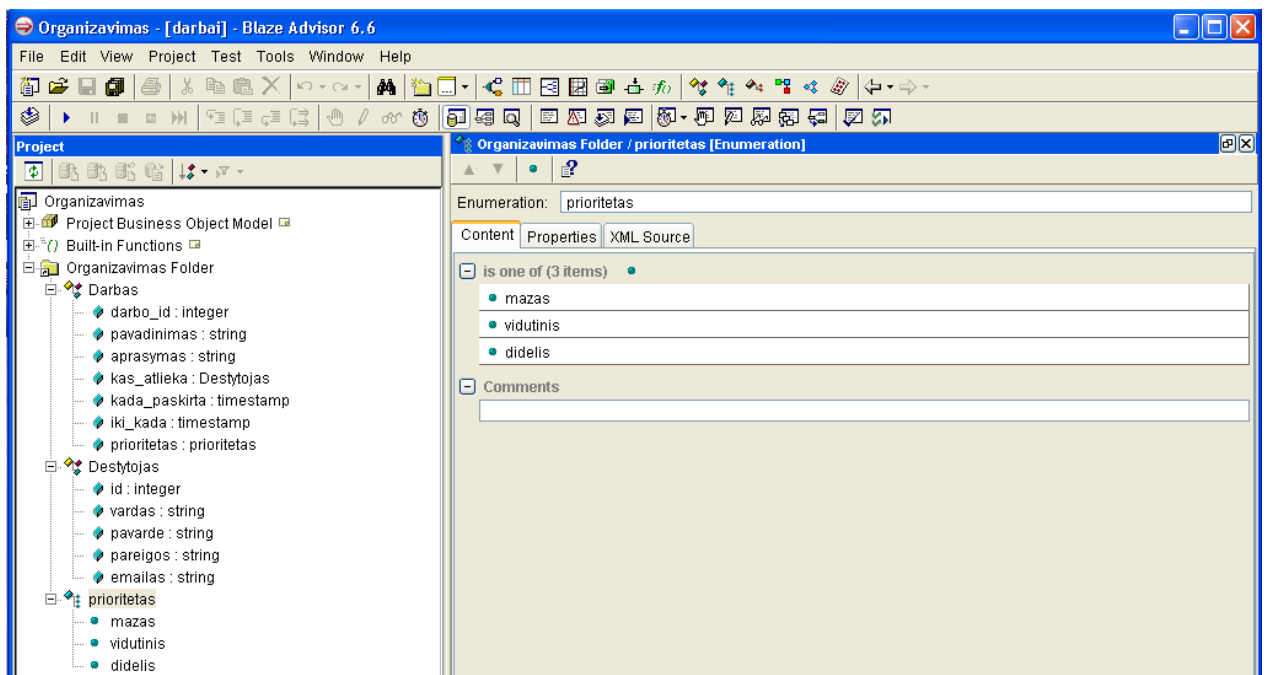
79 pav. Klasės atributų sukūrimas

Paspaudus meniu punktą „Project->Object Model Entities->New Enumeration“ sukuriamas reikšmių sąrašas. (69 pav.)



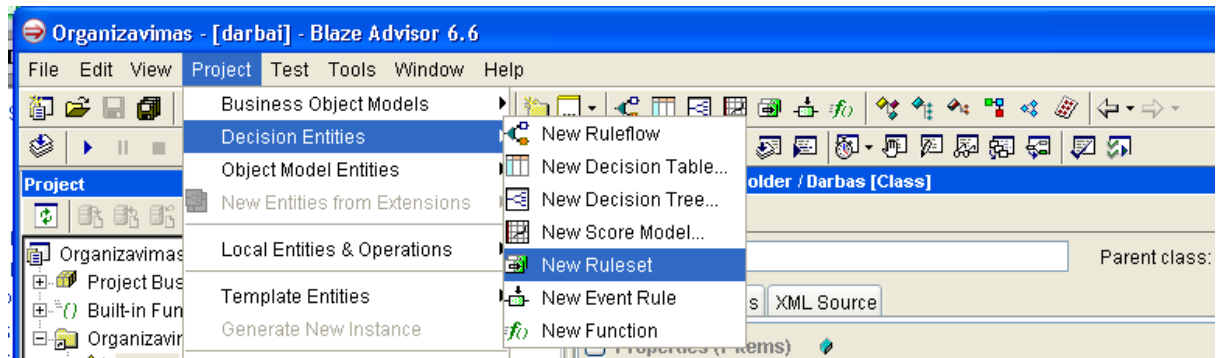
80 pav. Sąrašo sukūrimas

Atsidarius sąrašo tvarkymo langui, spaudžiame mygtuką „New Enumeration Item“. Sukuriama nauja sąrašo reikšmė. (70 pav.)



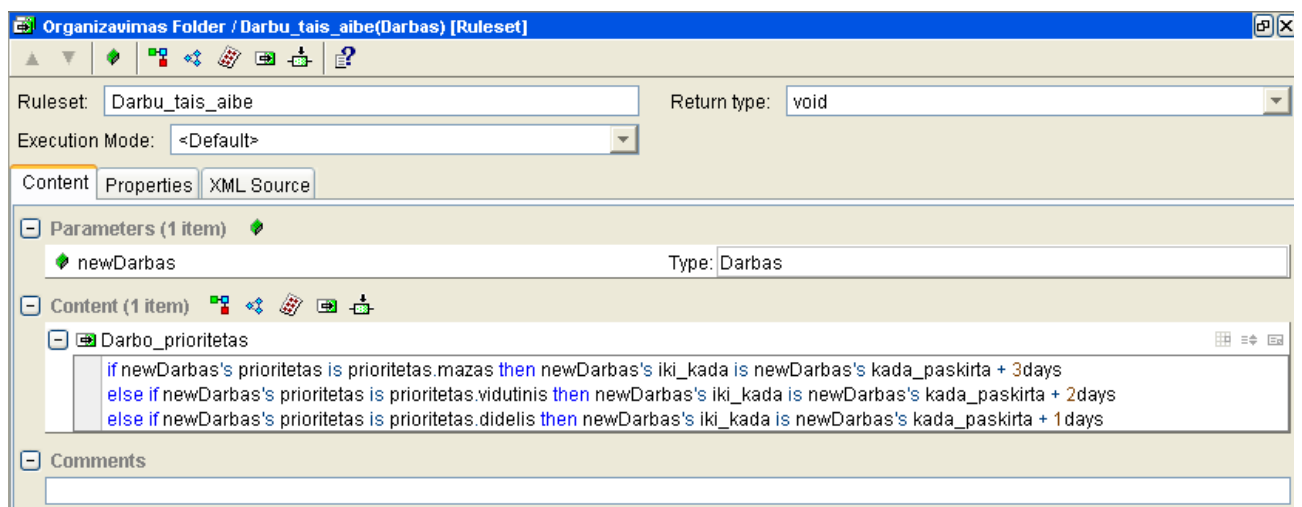
81 pav. Sąrašo reikšmių priskyrimas

Paspaudus meniu punktą „Project->Decision Entities->New Ruleset“ sukuriamas nauja taisyklių aibė.(71 pav.)



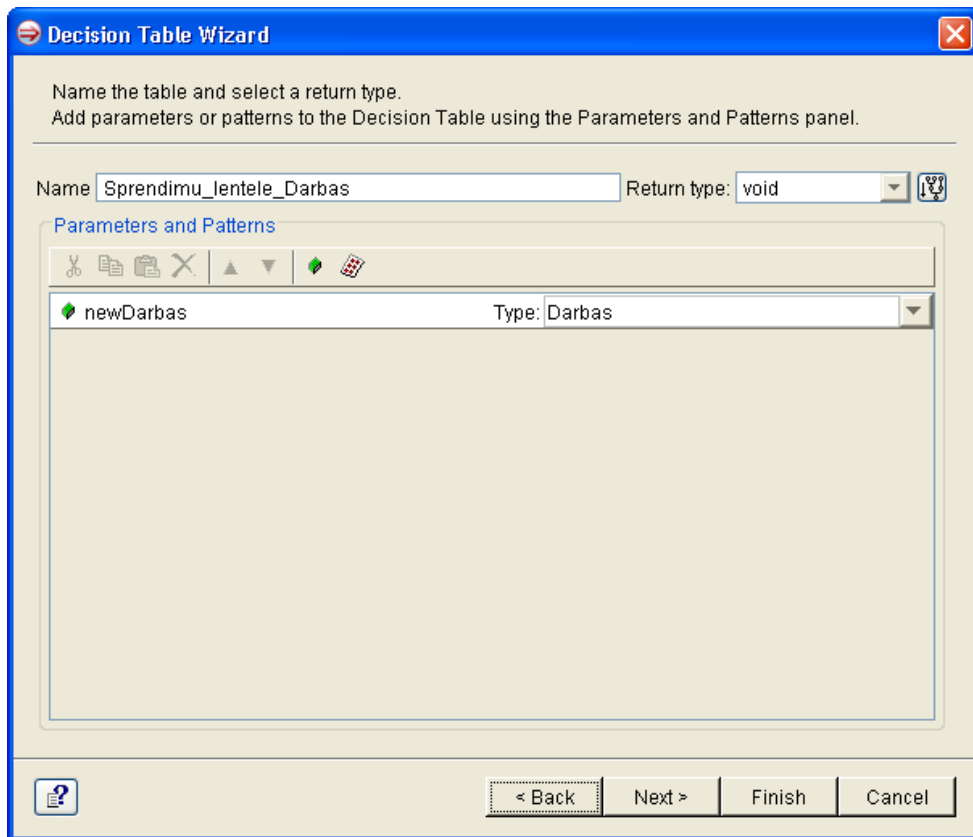
82 pav. Naujos taisyklių aibės sukūrimas

Taisyklių aibėje kuriamos kitos taisyklės (72 pav.). Taisyklių aibėje galima taip pat sukurti objektą, kintamąjį, šabloną, įvykio taisyklę. Taisyklių aibei gali būti perduodami parametrai.



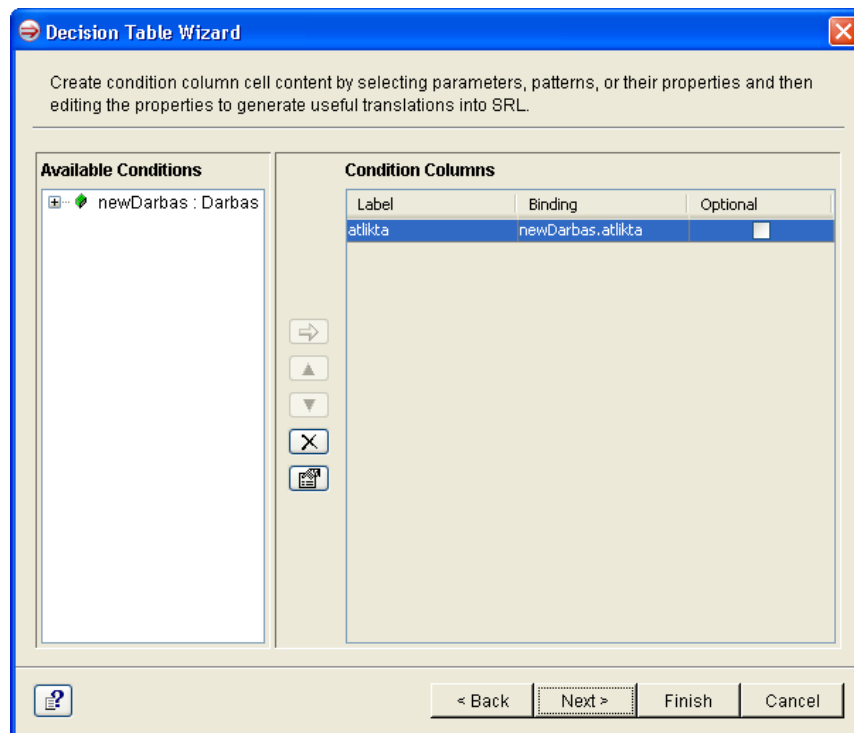
83 pav. Naujos taisyklės sukūrimas

Kuriant naują sprendimų lentelę nurodome lentelės pavadinimą, gražinamą tipą. Taip pat nurodome naudojamus objektus ir šablonus. (73 pav.)

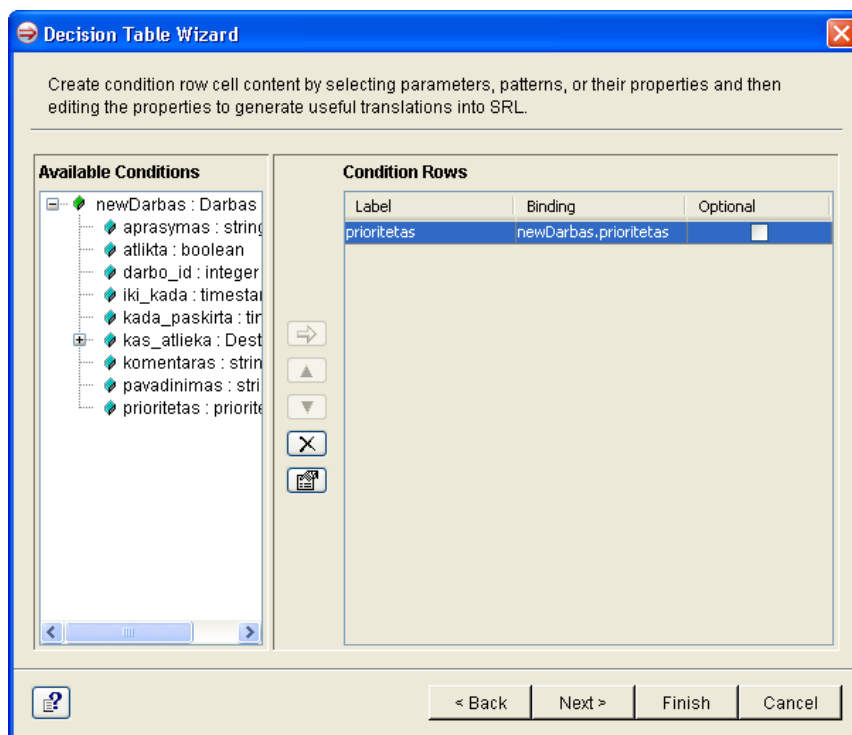


84 pav. Sprendimų lentelės sukūrimas

Toliau kuriant lentelę nurodoma stulpelių ir eilučių informacija. (74,75 pav.)

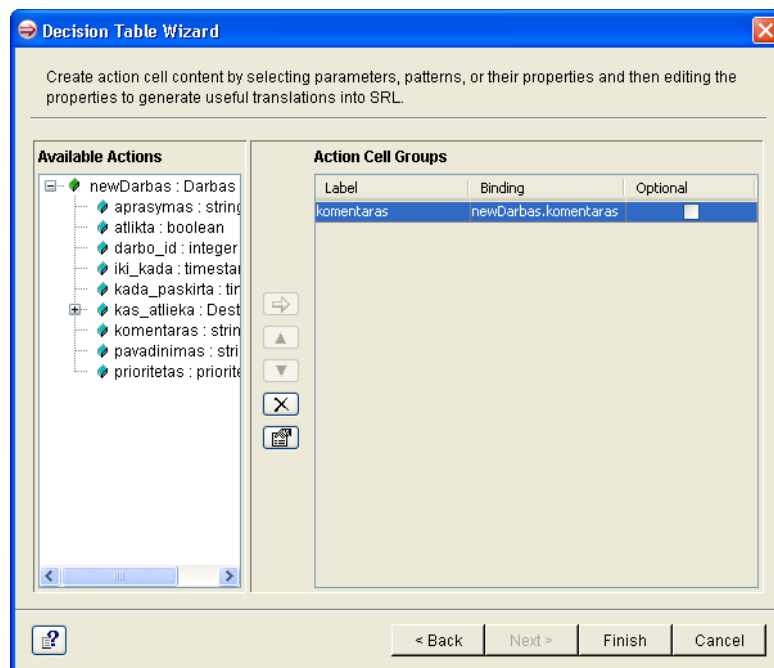


85 pav. Stulpelių sukūrimas sprendimų lentelėje



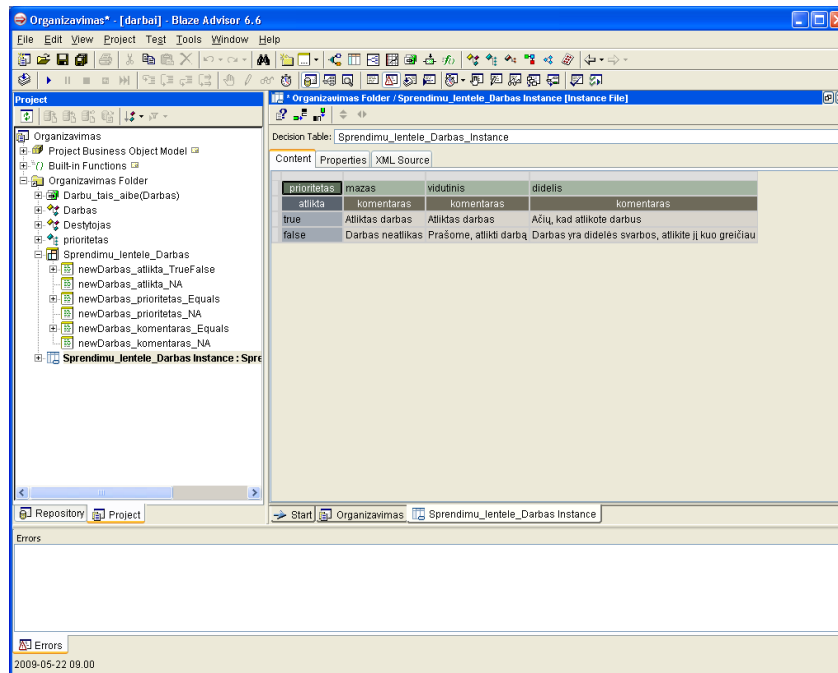
86 pav. Eilučių sukūrimas sprendimų lentelėje

Taip pat lentelėje reikia nurodyti, rezultatų elementus, kurie įgaus atitinkamas reikšmes, pagal eilučių ir stulpelių reikšmes.(76 pav.)



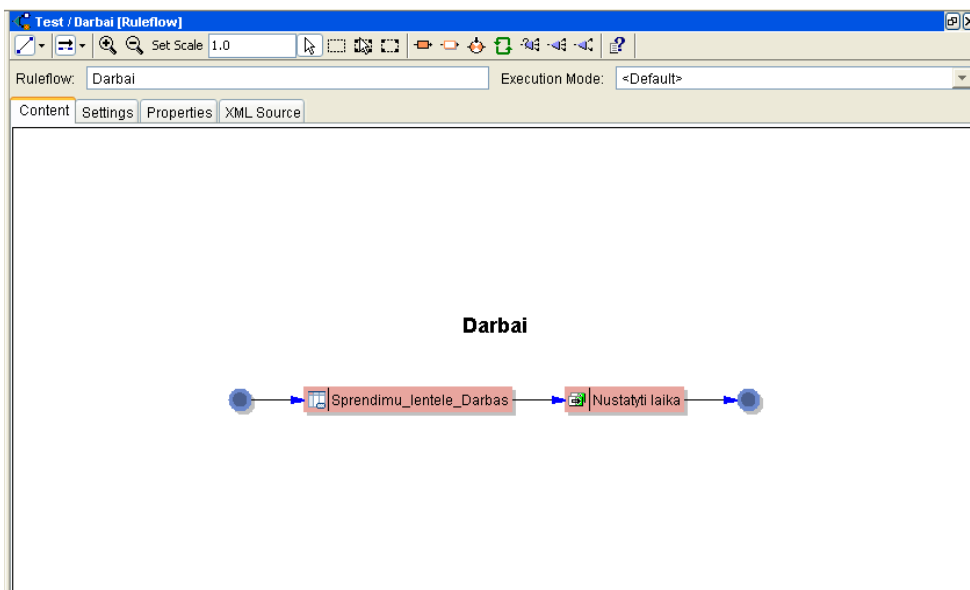
87 pav. Rezultatų elementų informacijos nurodymas

Turint sukurta sprendimų lentelę, jai nurodome prie kokių stulpelių ir eilučių reikšmių gausime atitinkamus rezultatus. (77 pav.)



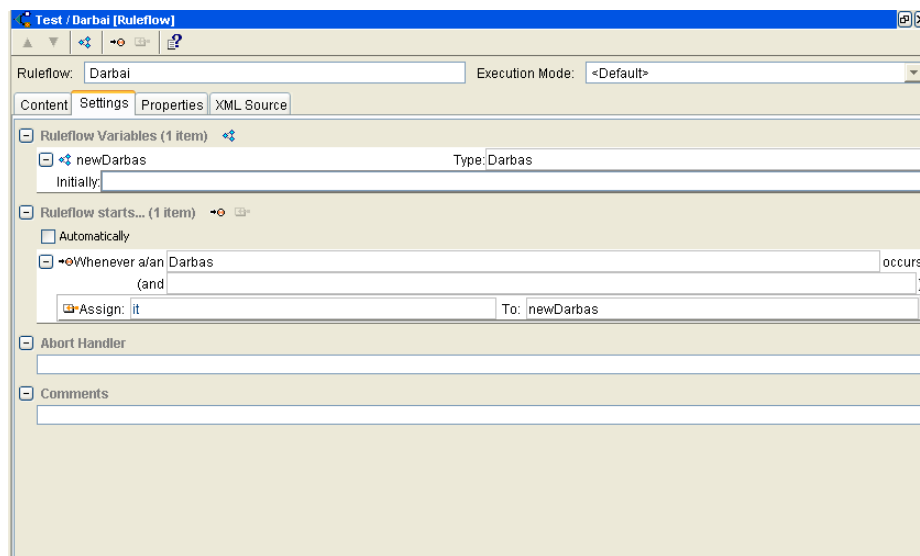
88 pav. Sprendimų lentelės reikšmių priskyrimas

Taisykles galima apdoroti vieną po kitos. Tai yra priėmus vieną sprendimą, atitinkamai parenkami kiti sprendimai. Tam naudojami taisyklių srautai. (78 pav.) Sraute nurodomi pradžios ir pabaigos taškai. Taip pat veiklos ir jų eiliškumas. Viskas atvaizduojama grafu.



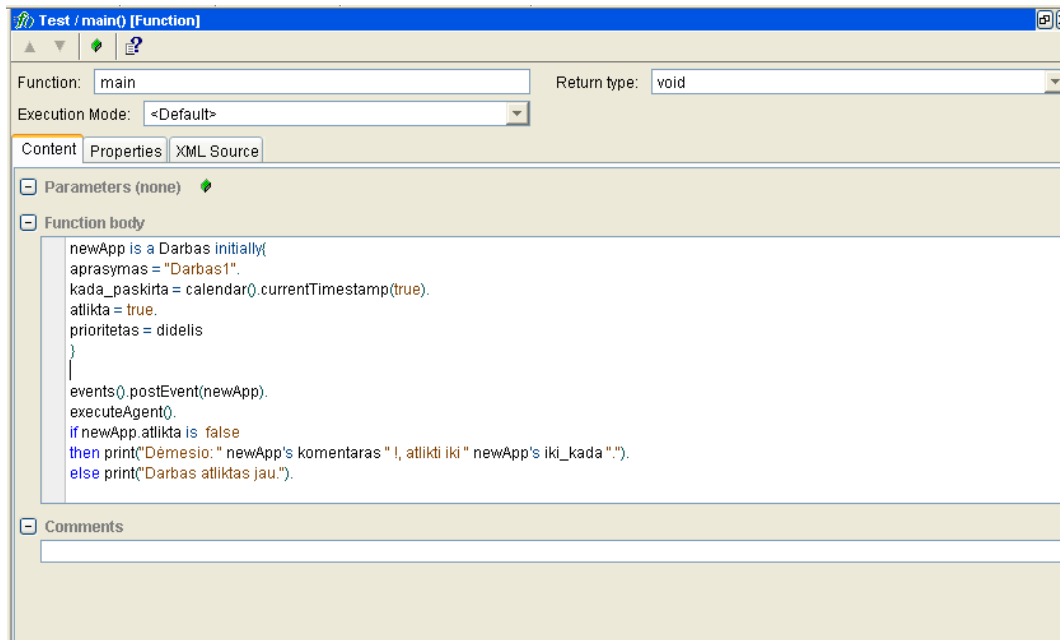
89 pav. Taisyklių srauto valdymas

Taisyklių srautui nurodome kokius parametrus paduodame ir kokią informaciją reiks apdorti. (79 pav.)



90 pav. Taisyklių srauto parametrai

Norint gauti tam tikrą sprendimą, reikia paleisti pagrindinę funkciją, nurodant pradinis duomenis (80 pav.). Blaze Advisor pagal aprašytas taisykles priima sprendimą ir jį pateikia vartotojui.



91 pav. Pagrindinė funkcija

Sistemos pateikti rezultatai patikrinus kokios svarbos darbas, bei ar dar jis neatliktas:

Dėmesio: Darbas yra didelės svarbos, atlikite jį kuo greičiau !, atlikti iki 2009-05-26 19.46.51

Execution complete.

9.2. Blaze Advisor taisyklių diegimo į taisyklių serverį vadovas

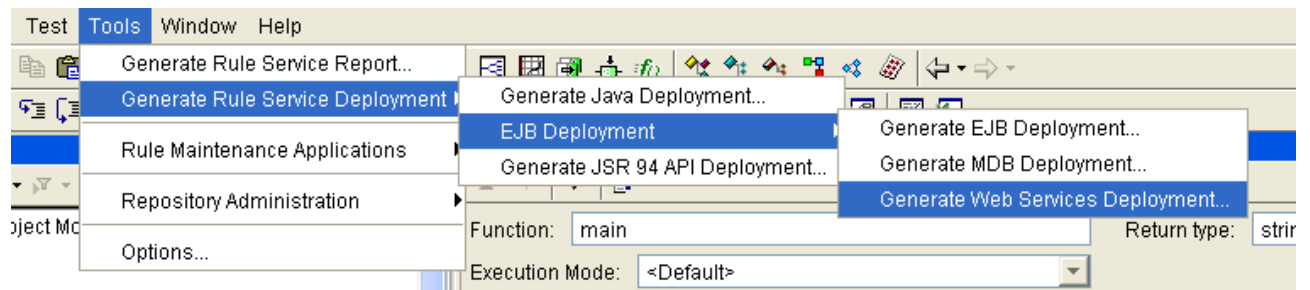
Greitas diegimo vedlys iškviečiamas meniu juostoje spaudžiant (81 pav.):

Tools->Generate Rule Service Deployment

Tuomet galime rinktis iš trijų skirtingų tipų:

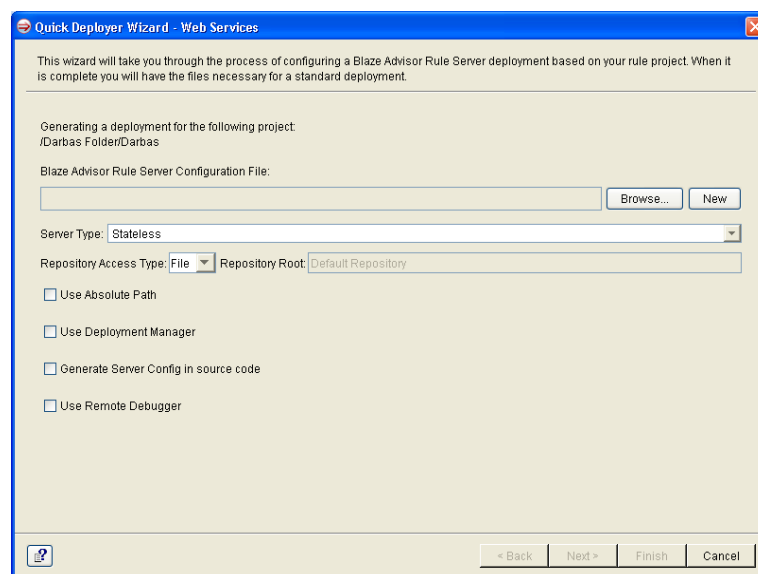
- Generate Java Deployment
- EJB Deployment
- Generate JSR 94 API deployment

Mūsų atveju renkamės EJB Deployment->Generate Web Services Deployment, nes taisyklės turi būti iškviečiamos iš programos sukurtos PHP kalba.



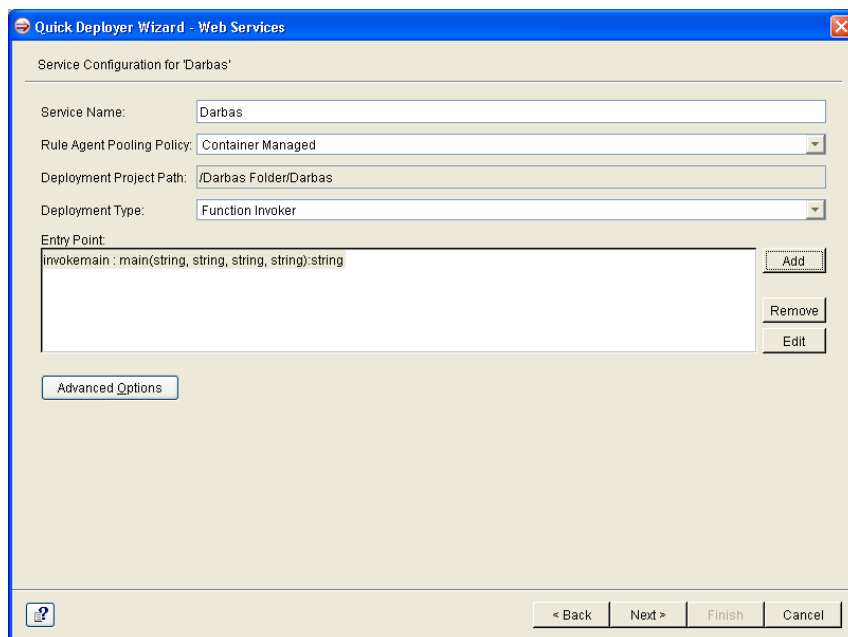
92 pav. Greitojo diegimo vedlio iškvietimas

- Atsidariusiame lange prie „Blaze Advisor Rule Server Configuration File“ spaudžiame „New“. (82 pav.)
- Dialogo lange paliekame numatytąjį pavadinimą ir spaudžiame OK.
- Kitus laukelius paliekame tuščius arba nustatytas numatytąsias reikšmes.



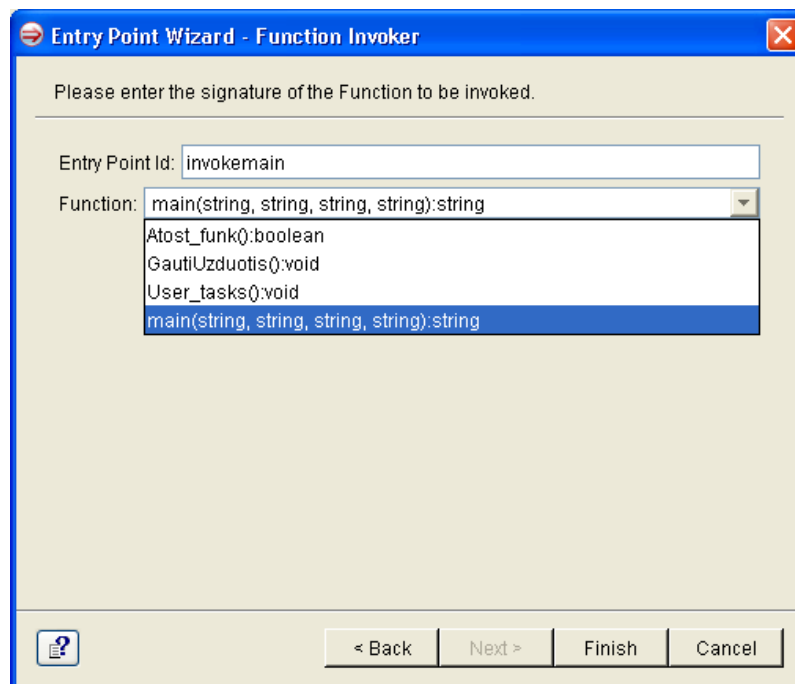
93 pav. Naujo konfiguracionio failo sukūrimas

- Spaudžiame „Next“
- Deployment type – pasirenkame „Function invoker“ (83 pav.)



94 pav. Įeities taškų nurodymas

- Spaudžiame „Add“, Atsidariusiame lange „Next“.
- Pasirenkame funkciją, kuri valdo visas taisykles ir spaudžiame „Finish“ (84 pav.)
- Grįžus prie pirmojo lango spaudžiame „Next“ (83 pav.)



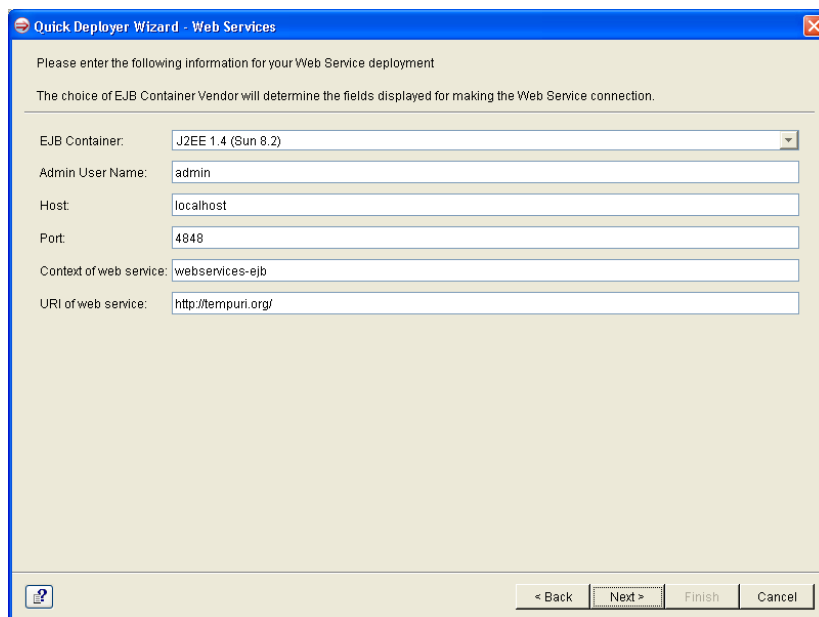
95 pav. Taisykles valdančios funkcijos pasirinkimas

- Nurodome katalogą, į kurį bus sugeneruoti failai (85 pav.)

- HTTP URL for Server Config – Nurodome kur bus padėtas taisyklių serverio konfigūracinis failas.
- Spaudžiame „Next“

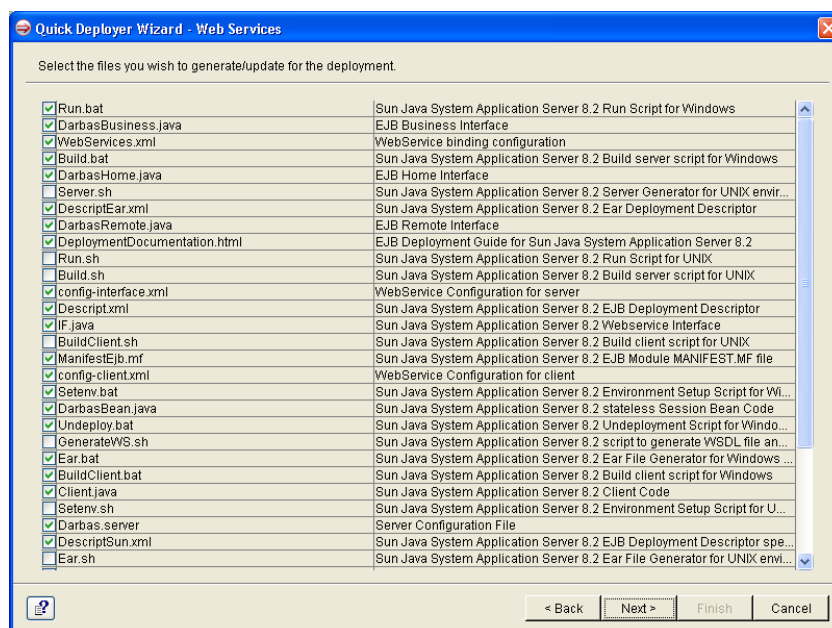
96 pav. Katalogo, į kurį bus generuojami failai, nurodymas

- Pasirenkame EJB konteinerio tipą, kuriame talpinsime. (86 pav.)
- Įvedame administratoriaus vartotojo vardą
- Serverio vardą
- Porto numerį
- Web paslaugų kontekstą
- Web paslaugų URI adresą
- Spaudžiame „Next“



97 pav. EJB konteinerio pasirinkimas

- Atsivėrusiame lange galima pasirinkti failus, kuriuos reik sugeneruoti. (87 pav.)
- Spaudžiame „Next“ ir „Finish“.



98 pav. Generuojamų failų pasirinkimas

a. Sugeneruotų failų aprašymai

- **Run.bat** - Paleidimo skriptas skirtas Windows.
- **DarbasBusiness.java** - EJB veiklos sąsaja.
- **WebServices.xml** – Web paslaugų jungties konfigūracija.
- **Build.bat** - Java failų kompiliavimo skriptas.
- **DarbasHome.java** - EJB Home sąsaja.
- **DescriptEar.xml** - Ear failo diegimo aprašas.

- **DarbasRemote.java** - EJB Remote sąsaja.
- **DeploymentDocumentation.html** - EJB diegimo aprašas skirtas Sun Java System Application Server 8.2
- **config-interface.xml** – Web paslaugų serverio konfigūracijos aprašas.
- **Descript.xml** - EJB diegimo aprašymas.
- **IF.java** - Web paslaugų sąsaja.
- **ManifestEjb.mf** - EJB modulio MANIFEST.MF failas.
- **config-client.xml** – Web paslaugų kliento konfigūracija.
- **DarbasBean.java** - Būsenos nepalaikančio Session Bean elemento kodas.
- **Undeploy.bat** - Išinstaliavimo Windows skriptas.
- **Ear.bat** - Ear failo generavimo skriptas.
- **BuildClient.bat** - Kliento sukūrimo skriptas.
- **Client.java** - Kliento kodas.
- **Darbas.server** – Serverio konfigūracinis failas.
- **DescriptSun.xml** - EJB diegimo aprašymas skirtas Sun.
- **Deploy.bat** - Diegimo skriptas.
- **Server.bat** - Paslaugų generavimo skriptas.
- **Runner.java** - Paleidimo kodas.
- **GenerateProxy.bat** - Kliento proxy objekto generavimo skriptas.
- **GenerateWS.bat** - Skriptas generuojantis WSDL failą ir mapping.xml failą.
- **DescriptEarSun.xml** - Ear diegimo aprašymas.

b. EAR failo sukūrimas ir patalpinimas į serverį

Pirmiausia turime nustatyti kintamuosius faile Setenv.bat. Kintamieji nurodo, kelius iki Java ir Blaze advisor katalogų.

Jei keliai nėra nurodyti tinkamai, pakoreguojame pirmas 4 neužkomentuotas eilutes:

```
@set J2EE_HOME=C:\Sun\AppServer
@set JAVA_HOME="C:\Progra~1\Java\jdk1.6.0_12"
@set ADVISOR_SERVER_HOME=C:\Blaze\Advisor67
@set ADVISOR_LICENSE=C:\Blaze\Advisor67\..\license
```

Komandinėje eilutėje paleidžiame Setenv skriptą.

Komandinėje eilutėje paleidžiame Build skriptą, kuris sukompiluoja Java failus.

Patikriname ar teisingai užpildytas „HTTP Url for Server Config“ laukas faile Descript.xml.

Komandinėje eilutėje paleidžiame GenerateWS skriptą, kuris sugeneruoja DarbasService.wsdl and mapping.xml failus.

Komandinėje eilutėje paleidžiame Server skriptą, kuris sugeneruoja JAR tipo failą.

Komandinėje eilutėje paleidžiame Ear skriptą, kuris sukelia JAR tipo failus į EAR tipo. EAR tipo failą jau galime diegti į serverį.

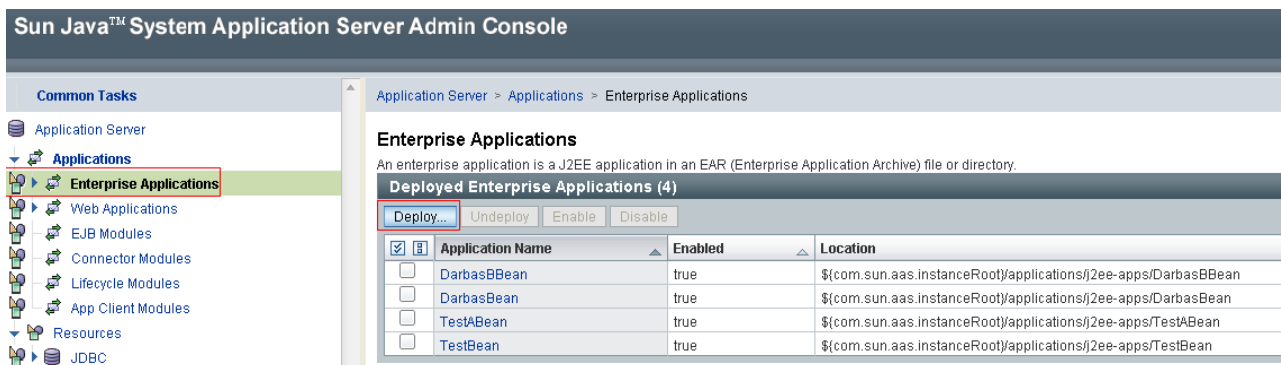

```

C:\>cd Sun\AppServer\domains\domain1\docroot\Testas19
C:\Sun\AppServer\domains\domain1\docroot\Testas19>Setenv
C:\Sun\AppServer\domains\domain1\docroot\Testas19>Build
C:\Sun\AppServer\domains\domain1\docroot\Testas19>GenerateWS
C:\Sun\AppServer\domains\domain1\docroot\Testas19>Server
Create the Darbas19 session bean jar
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
C:\Sun\AppServer\domains\domain1\docroot\Testas19>Ear
Create the Enterprise Application jar
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.
    1 file(s) copied.

```

99 pav. EAR tipo failo sukūrimas

Prisijungiamė prie Sun Java System Application Serverio administratoriaus konsolės. Pasirenkame „Enterprise Application“ ir spaudžiame mygtuką „Deploy...“. (89 pav.)



100 pav. Taisyklių paslaugų diegimas

Pasirenkame sugeneruotą EAR tipo failą. Spaudžiame „Next“ ir Sekančiame lange „OK“. (90 pav.)

Deploy Enterprise Application

Next Cancel

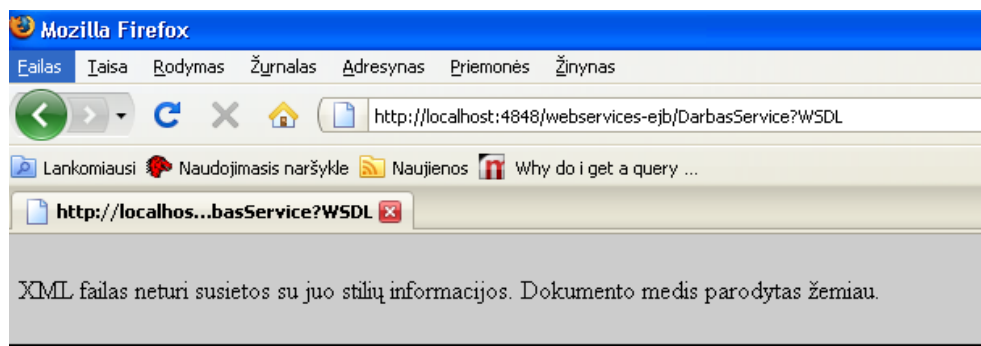
Specify the location of an application to deploy. Applications can be in packaged files, such as .ear, or in the standard Enterprise Application directory format.

Location: Specify a package file to upload to the Application Server.File To Upload: Pasirinkti... Specify a package file or a directory path that must be accessible from the server.File Or Directory: **101 pav. EAR tipo failo įkėlimas į serverį**

Norėdami patikrinti ar teisingai įkėlėme EAR failą, atveriamė naršyklėje:

<http://localhost:4848/webservices-ejb/DarbasService?WSDL>

Jei naršyklėje matome WSDL failą, tai web paslaugos įdiegtos teisingai. (91 pav.)



```

- <definitions name="DarbasService" targetNamespace="http://tempuri.org/">
  <types/>
  - <message name="IF_invokemain">
    <part name="String_1" type="xsd:string"/>
    <part name="String_2" type="xsd:string"/>
    <part name="String_3" type="xsd:string"/>
    <part name="String_4" type="xsd:string"/>
  </message>
  - <message name="IF_invokemainResponse">
    <part name="result" type="xsd:string"/>
  </message>
  - <portType name="IF">
    - <operation name="invokemain" parameterOrder="String_1 String_2 String_3 String_4">
      <input message="tns:IF_invokemain"/>
      <output message="tns:IF_invokemainResponse"/>
    </operation>
  </portType>
</definitions>

```

102 pav. WSDL failas**c. Taisyklių paslaugų iškvietimas iš internetinės sistemos**

Taisyklių iškvietimui naudojamas SOAP. Tai yra XML paremtas protokolas, kuris leidžia programoms keistis informacija per HTTP. Toliau pateikiamos kelios PHP programinio kodo eilutės su komentarais.

Soap bibliotekoms iškviešti naudojama eilutė:

```
require_once('C:/xampp/htdocs/opengoo/library/soap/nusoap.php');
```

Sukuriamas naujas web paslaugų klientas nurodant WSDL failo adresą:

```
$client = new nusoap_client('http://DARBO:3059/webservices-  
ejb/DarbasService?WSDL');
```

Sukuriamas parametrų masyvas. Parametrai turi atikti WSDL dokumente aprašytus parametrus:

```
$param = array('String_1' => $user_id , 'String_2' => $st, 'String_3' => $end,  
'String_4' => $task_data['priority']);
```

Call metodu kreipiamasi į web paslaugas. Pirmasis parametras nurodo funkciją, į kurią norime kreiptis. Antrasis – parametrų masyvas perduodamas funkcijai, į kurią kreipiamės.

```
$result = $client->call('invokemain', $param, 'http://tempuri.org',  
'http://tempuri.org/invokemain');
```

9.3. Straipsnis „Veiklos taisyklių taikymas internetinių informacinių sistemų kūrime“

Vytenis Darulis
Informacijos sistemų katedra
Kauno technologijos universitetas
Kaunas
vytenisd@gmail.com

Vadovas Tomas Skersys
Informacijos sistemų katedra
Kauno technologijos universitetas
Kaunas
tomas.skersys@ktu.lt

Santrauka — Straipsnyje nagrinėjamos veiklos taisyklių panaudojimo informacinių sistemų kūrimo procese galimybės. Aptariami veiklos taisyklių valdymo būdai. Apžvelgiamos veiklos taisyklių valdymo paketo Blaze Advisor galimybės. Veiklos taisyklių įvedimui siūloma naudoti parametrizuotais šablonais. Aprašoma veiklos taisyklių integracija į PHP kalbą sukurtą internetinę aplinką.
Raktiniai žodžiai – Veiklos taisyklės; Blaze Advisor; Informacinės sistemos

I. ĮVADAS

Organizacijos naudoja įvairias informacines ir programų sistemas, kurios yra jų veiklos dalis. Svarbu, kad visos veiklos operacijos būtų atliekamos ne tik teisingai, bet ir nustatytu laiku, arba griežtai apibrėžtu nuoseklumu, prisilaikant numatytos tvarkos, veiksmų scenarijaus, kur kiekvienam naudotojui griežtai apibrėžtos jo teisės ir galimybės atsižvelgiant į pareigas. Taip pat labai svarbu, kad visa valdymui reikalinga informacija būtų prieinama ne tik per lokalų kompiuterių tinklą, bet ir naudojantis internetu. Atsakingas darbuotojas, nesvarbu kur jis bebūtų – namuose, komandiruotėje, seminare ar kitur – visą reikalingą informaciją privalo turėti po ranka. Tam naudojamos įvairios darbo organizavimo sistemos.

Veikla vyksta nuolat besikeičiančioje aplinkoje, kartu su aplinka keičiasi pati veikla ir jos taisyklės. Pakeitimai taip pat turi atsispindėti informacinėse sistemose. Išskyla problema, kaip veiklos taisyklės, suformuluotas veiklos lygmenyje, kuo greičiau ir efektyviau perkelti į informacinę sistemą.

Tradicinės informacinės sistemos nebetenkina šiuolaikinių organizacijų poreikių. Tokios sistemos, kuriose veiklos logika yra integruojama į programinį kodą, yra keičiamos į sistemas, kurios palaiko veiklos taisyklės atskirai nuo programinio kodo. Taip užtikrinamas greitesnis organizacijos veiklos valdymas.

Straipsnyje analizuojamas veiklos taisyklių bei veiklos taisyklių valdymo sistemų taikymas organizacijų informacinėse, internetinėse sistemose. Nagrinėjamas informacinės sistemos kūrimo procesas paremtas veiklos taisyklėmis. Pateikiama metodika, kaip integruojamas veiklos taisyklių valdymo įrankis, taisyklių saugykla ir darbo organizavimo sistema.

II. VEIKLOS TAISYKLĖS

Veiklos taisyklė (VT) – tai [1]:

- teiginys, kuris apibrėžia arba sąlygoja tam tikrą veiklos aspektą. VT skirta apibrėžti veiklos struktūrai arba valdyti, įtakoti veiklos eigseną;
- atominis, deklaratyviai specifikuotas, daugkartinio panaudojimo veiklos logikos fragmentas.

VT sakiny – tai deklaratyvus sakiny, nusakantis veiklai nustatytą struktūrą arba apribojimą. Kiekvienas VT sakiny gali būti susijęs su vienu ar daugiau kitų VT sakinių.

VT sakiny vartojamas užrašant vieną ar daugiau nedalomų veiklos taisyklių. Veiklos taisyklė apibrėžia arba apriboja tam tikrą veiklos aspektą, tačiau ji negali būti toliau skaidoma į detalesnes VT. Tolesnis dekomponavimas lemtų svarbios veiklą liečiančios informacijos praradimą.

Formaliame taisyklės sakinyje vartojama specifinė formali gramatika. Formalus taisyklės sakiny privalo būti užrašytas pagal formalaus reiškinio tipo sudarymo taisyklės.

Viena taisyklė gali būti tokia:

- Jei pirkėjas yra iš Vilniaus, tada pasiūlyti paslaugą nemokamai.

Kita taisyklė:

- Jei pirkėjo amžius > 70, tada pasiūlyti 3 paslaugas nemokamai.

Iš pirmo žvilgsnio atrodo, kad šios taisyklės turi mažai bendro, tačiau detaliau išanalizavus matome, kad abi taisyklės atitinka tą pačią struktūrą:

Jei pirkėjas \diamond savybė \diamond palyginimas \diamond reikšmė,

Tada pasiūlyti nemokamai \diamond skaičius \diamond prekė.

Veiklos taisyklės (VT) yra būdas struktūrizuoti veiklos žinias. Veiklos taisyklėms nuolat keičiantis veiklos lygmenyje išskyla poreikis išlaikyti visus pakeitimus ir informacinės sistemos (IS) lygmenyje. M. Chisholm [3] teigia, kad programuotojai 40 % laiko skiria programų atgalinės inžinerijos atlikimui, tam, kad ją suprastų, galėtų padaryti pakeitimus, taisyti klaidas ar integruoti su kitomis aplikacijomis. Viso šio užtrukimo priežastimi dažnai yra ta, kad dokumentacija yra beveik visada nepilna, pasenusi ir nepatikima. Efektyvus veiklos taisyklių pakartotinis panaudojimas sistemų gyvavimo cikle leidžia

paspartinti patį informacinių sistemų reorganizavimo procesą. Tam išskiriamas veiklos taisyklių sluoksnis. Pagrindiniai faktoriai lemiantys poreikį išskirti veiklos taisykles:

- Poreikis atskirti VT ir jas traktuoti kaip atskiras informacines vertybes.
- Poreikis stebėti VT kaitą.
- Poreikis pateikti VT fiksuota forma.
- Poreikis taisykles ruošti taip, kad jas būtų galima nesunkiai pakeisti.

Atskirtas veiklos taisyklių sluoksnis suteikia galimybę verslo žmonėms, neturintiems techninių žinių, peržiūrėti, kurti, keisti, šalinti veiklos taisykles be įsikišimo į programinį kodą.

Ronald G. Ross tai vadina žinių tašku [5] (angl. the point of knowledge (POK)). Tai vieta, kur vystomos, diegiamos, taikomos ir galiausiai pašalinamos veiklos taisyklės.

Taikomosios programos, tinkamos veiklos taisyklių technologijai taikyti, sutinkamos visur, kur egzistuoja poreikis greitai ir lengvai realizuoti pokyčius deklaratyviose organizacijos žiniuose. Pasaulyje yra keletas organizacijų, kurios užsiima sistemų, pagrįstų veiklos taisyklėmis, realizacija.

Buvo išnagrinėtos tokios veiklos taisyklių valdymo sistemos:

- Quick rules
- Infrex
- Versata Logic Suite
- ILOG Rules
- Blaze Advisor

Sistemos buvo lyginamos pagal tokius kriterijus: Taisyklių aprašymo kalba, palaikomos platformos, ar turi galimybę kurti sprendimų medžius, lenteles, taisyklių srautus, ar yra galimybė naudoti šablonus, ar palaikoma versijų kontrolė.

Atlikus veiklos taisyklių valdymo sistemų analizę, nuspręsta darbo organizavimo sistemą papildyti Blaze Advisor taisyklių valdymo sistema. Pagal atliktą analizę galima teigti, kad Blaze Advisor turi daugiausiai galimybių. Tai yra šis produktas palaiko sprendimų lenteles, sprendimų medžius, taisyklių srautus. Taip pat galima naudoti šablonus.

III. BLAZE ADVISOR – TAISYKLIŲ VALDYMO SISTEMA

Viena iš lyderiaujančių pasaulinėje rinkoje kompanijų Fair Isaac sukūrė ir toliau tobulina veiklos taisyklių valdymo sistemą Blaze Advisor.

Blaze Advisor – veiklos taisyklių valdymo sistema, kurios pagalba galima kurti, diegti ir prižiūrėti į veiklos taisykles orientuotas sistemas. Veiklos taisyklių aprašymo kalba leidžia ne techniniam verslo personalui valdyti veiklos taisyklės.

Ši įranki sudaro pagrindiniai elementai, reikalingi veiklos taisyklėms įgyvendinti: sprendimų variklis, taisyklių saugykla ir jos valdymas.

Pagrindinis naudojamas veiklos taisyklių atvaizdavimo metodas – struktūrinė taisyklių kalba (Structured Rule Language SRL). SRL sintaksė yra formali ir arti natūralios kalbos (arti anglų kalbos reikšminių žodžių). Dažnai galima naudoti taisyklių šablonus, tokiu atveju taisyklių rašymas tampa labiau intuityvus, ir taip sukurtos taisyklės yra lengviau skaitomos.

Su Blaze Advisor Builder galima kurti sudėtingas taisykles. Yra galimybė naudoti besišakančias taisykles, sudarytas iš kelių žingsnių, taip sukuriant taisyklių srautus. Ši sistema palaiko sprendimų lenteles, kurios naudojamos konstruoti ir atnaujinti sąlygas ir veiksmus veiklos taisyklėse. Šių komponentų pagalba galima sudaryti įvairaus sudėtingumo veiklos taisykles, kontroliuojančias organizacijos veiklos procesus.

IV. VEIKLOS TAISYKLĖS IR WEB SISTEMOS

Internetinė informacinė sistema – tai informacinė sistema, kuri naudoja internetines technologijas, tam kad teiktų informaciją ar paslaugas vartotojams. Vartotojų sąsajai dažniausiai naudojama internetinė naršyklė.

Blaze Advisor turi galimybes, leidžiančias supaprastinti dinamišką interaktyvių taikomųjų programų informacijos atnaujinimą ir valdymą interneto puslapiuose. Taisyklės gali apibrėžti, kokia informacija turėtų būti pateikiama į interneto puslapius. Technologiniai sprendimai, nusakantys kaip fiziškai atvaizduoti interneto puslapius, yra palikti išorinėms sistemoms, todėl yra palaikomas sprendimų atskyrimas nuo infrastruktūros kodo. Lengvai apibrėžiamos ir palaikomos sąsajos su išorinėmis sistemomis teikia greitį ir efektyvumą kūrimo ir vykdymo metu.

Blaze Advisor taisyklių serveris (Rule Server) yra Blaze Advisor veiklos taisyklių valdymo sistemos komponentas. Naudojant taisyklių serverį galima įdiegti su BA Builder sukurtus taisyklių projektus kaip taisyklių paslaugas. Taisyklių paslaugos veikia kaip programos komponentas, kuris atlieka taisyklių projekto aprašytus veiklos procesus ir yra prieinamas kitiems sistemos komponentams. Naudojant EJB tipo įdiegimą, taisyklių serveris teikia paslaugas, kai EJB konteineris apdoroja klientų užklausas.

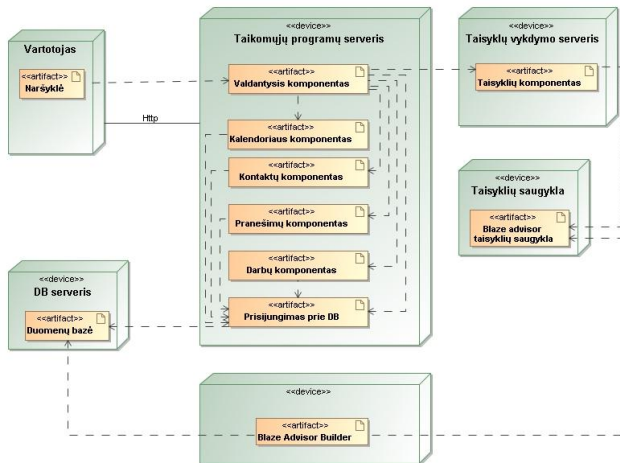
Galima sukonfigūruoti Blaze Advisor taisyklių serverį taip, kad jis apdorotų vieną arba daugiau taisyklių paslaugų. Taisyklių paslaugos gali būti prieinamos vieno arba kelių nutolusių klientų naudojant paketinį apdorojimą.

EJB greitis diegimo vedlys gali sugeneruoti failus šiems EJB konteinerių tipams[1]:

- J2EE 1.4 (Sun 8.2)
- JavaEE5
- JBoss 4.0.4
- OC4J (10.1.3)

- Oracle 9iAS Release 2 (9.0.3)
- WebLogic 8.1, 9.1, and 9.2
- WebSphere 6.0, and 6.1

Komponentų įdiegimo diagrama (Pav. 1) atvaizduoja bendrą sistemos architektūrą. Vartotojas gali naršyklės pagalba iškviešti reikalingas funkcijas. Jei reikalingi duomenys - taikomųjų programų serveris kreipiasi į duomenų bazės serverį. Taisyklių apdorojimui naudojamas taisyklių vykdymo serveris.



Paveikslas 1 Veiklos taisyklių integracijos proceso veiklos diagrama

V. VEIKLOS TAISYKLIŲ INTEGRACIJOS METODIKA

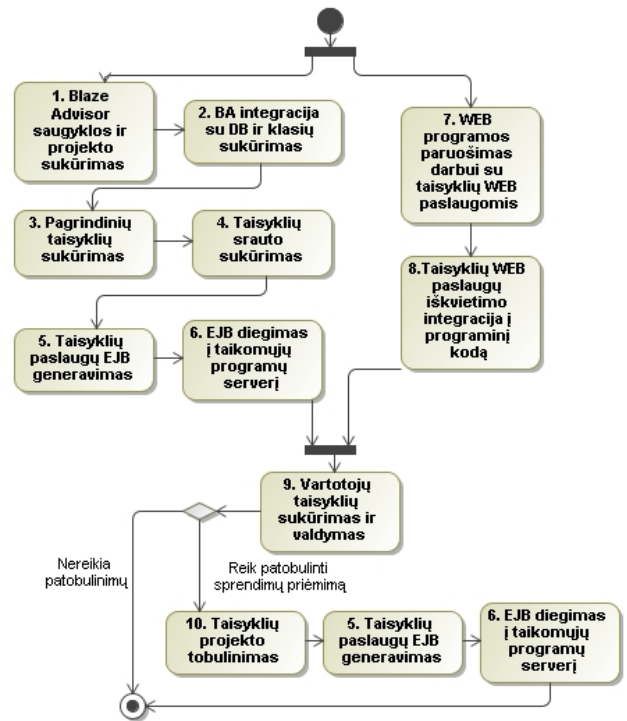
Veiklos taisyklių (VT) integracijos į taikomųjų uždavinių sistemą procesas pavaizduotas (Pav. 2). Veiklos diagramoje integracijos etapai sužymėti skaitmenimis. Apžvelgsime kiekvieną etapą atskirai.

Integracija prasideda Blaze Advisor saugyklos ir projekto sukūrimu (1 etapas). Blaze Advisor saugykloje saugomos visos veiklos taisyklės, klasės, taisyklių srautų informacija ir kiti taisyklių valdymo elementai. Vienoje saugykloje gali būti keli projektai.

Antrame etape su Blaze Advisor esančiu vedliu nustatomos klasės, kurios komunikuos su duomenų bazių lentelėmis per JDBC jungtį. Veiklos logiką realizuojančios klasės kreipiasi į duomenis, esančius duomenų bazėje, ir naudoja juos sprendimams priimti. MySQL duomenų bazė taip pat saugo veiklos taisyklių parametrus, kuriuos nusistato vartotojai.

Trečiame etape kuriamos reikalingos veiklos taisyklės. Tam naudojamos IF..THEN tipo taisyklės, sprendimų medžiai, sprendimų lentelės, taisyklių srautai. Taip pat sukuriama taisyklė, kurios naudos vartotojų parametrus.

Ketvirtame etape sukuriama pagrindinis taisyklių srautas, kuris pagal vartotojų sukurtus parametrus ir taisyklių grąžintas reikšmes nustato, kokias funkcijas ar taisykles vykdyti toliau, kada grąžinti galutinį rezultatą.



Paveikslas 2 Veiklos taisyklių integracijos proceso veiklos diagrama

5 etapas – tai taisyklių projekto transformavimas į taisyklių paslaugas (EJB failų generavimas). Procesas susideda iš šių dalių[1]:

- Taisyklių serverio sukonfigūravimas pagal platformą į kurią diegsime taisykles.
- Taisyklių projekto nustatymas paslaugoms teikti.
- Aprašymas paslaugų įeities taškų, kurie leis klientams pasiekti paslaugas.

Įeities taškai tai yra veiklos metodai. Metodų naudojami argumentai, perduoda duomenys į taisykles, bei grąžina rezultatus.

Įvykdžius prieš tai aprašytus etapus, sugeneruojami failai, kurie diegiami į pasirinktą taikomųjų programų serverį.

Taisyklių iškvietimui naudojamas SOAP. Tai yra XML paremtas protokolas, kuris leidžia programoms keistis informacija per HTTP. Į WEB programą reiktų įtraukti SOAP reikalingas bibliotekas. 7-8 etapai gali būti vykdomi lygiagrečiai su 1-5 etapais.

Sekančiame etape (8) į programinį kodą įterpiamos eilutės, kurios iškviečia veiklos taisykles, apdoroja grąžintą rezultatą ir padeda vartotojams valdyti savo taisyklių parametrus.

Sukūrus visą aplinką vartotojui, prijungus taisyklių paslaugas, vartotojas gali pradėti dirbti su sistema. Tai yra ne tik atlikti savo veiklą, bet ir kurti veiklos taisykles, nustatant tam tikrus parametrus.

Pasikeitus tam tikroms veikloms atliekami 10,5,6 etapai, pavaizduoti paveikslėlyje.

VI. VEIKLOS TAISYKLIŲ INTEGRACIJOS PAVYZDYS

Veiklos taisyklių valdymo sistemos Blaze Advisor integracija buvo įgyvendinta, kuriant pavyzdinę darbo organizavimo sistemą. Apžvelgsime vienos iš taisyklių veikimą ir jos pateikiamą rezultatą.

Veiklos taisyklės teiginio šablonas yra bazinė teiginio struktūra ar kalbos išraiška, kuri naudojama taisyklei išreikšti[4].

Šablono pagalba vartotojo kuriama taisyklė susideda iš tokių parametrų:

- Antraštė
- Taisyklės objektas (tai gali būti užduoties ar įvykio pavadinimas, užduočių skaičius - kiek vyksta tuo metu, trukmė, atsakingas asmuo, kūrėjas, pradžios ir pabaigos datos.)
- Palyginimo simbolis (<, >, =, <=, >=)
- Objekto reikšmė (reikšmė su kuria lyginama. Pvz. Galime įrašyti datą, su kuria lygins užduočių pradžias.)
- Gražinimo tipas. Galimos reikšmės:
 - Taisyklė neleidžia atlikti veiksmo: sukurti užduoties, pakeisti įvykio datos ir pan. Išvedamas klaidos pranešimas.
 - Taisyklė leidžia vykdyti veiksmus.
 - Taisyklė tik informuoja vartotojus ir išveda informacinį pranešimą.
- Taisyklės gražinama reikšmė. Tai informacinis pranešimas parodomas vartotojui, kai suveikia atitinkama taisyklė.

Antraštė: *

Atostogos

Taisyklės sąlygos dalis:

Taisyklės objektas:

Pavadinimas

Palyginimo simbolis:

=

Objekto reikšmė:

Atostogos

Taisyklės veiklos dalis:

Gražinimo tipas:

Neleisti vykdyti veiklos

Taisyklės gražinama reikšmė:

Darbuotojas tuo metu atostogauja

Paveikslas 4 Veiklos taisyklės parametrų valdymas

Tarkime sukuriame taisyklę (Pav. 3), kuri neleidžia priskirti užduočių, kai yra užduotis su pavadinimu „Atostogos“.

Bandant įvesti naują užduotį pasirenkame datą, kuri yra tuo metu, kai darbuotojas atostogauja. Sistema iškviečia taisyklę. Sistema patikrina, ar darbuotojas tuo metu atostogauja. Jei taip, tuomet gražinama eilutė „Darbuotojas tuo metu atostogauja“. (Pav. 4)

Error: Demesio: Darbuotojas tuo metu atostogauja

Apžvalga Užrašai E-paštas Kontaktai Kalendorius Dokumentai Užduotys

Atgal

Nauja užduotis Sukurti užduotį

Pavadinimas: *
Nauju metu planai

Darbo sritis - Žymės - Užduoties informacija - Pasikartojanti užduotis - Priminimai - Papildomos savvbe:

Užduoties informacija

Darbu etapas: (Užduotis gali būti priskirta egzistuojančiam darbų etapui)
-- Niekas --

Pagrindinė užduotis:

Pradžios data: (yyyy/mm/dd)

Galutinė data: (yyyy/mm/dd)

Paveikslas 3 Internetinės sistemos išvedamas pranešimas, kuris gautas iš veiklos taisyklių.

VII. IŠVADOS

Veiklos taisyklių sluoksnio atskyrimas nuo programinio kodo yra pakankamai naujas techninis dalykas. Manoma, kad perspektyvoje vis dažniau tradicinės sistemos bus keičiamos į sistemas, kurios palaiko veiklos taisykles atskirai nuo programinio kodo.

Efektyvus veiklos taisyklių pakartotinis panaudojimas sistemų gyvavimo cikle leidžia paspartinti patį informacinių sistemų reorganizavimo procesą.

Atskirtas veiklos taisyklių sluoksnis ir atitinkami įrankiai suteikia galimybę verslo žmonėms, neturintiems techninių žinių, peržiūrėti, kurti, keisti, šalinti veiklos taisykles be įsikišimo į programinį kodą.

Veiklos taisyklių valdymo sistemos Blaze Advisor integracija buvo įgyvendinta, kuriant pavyzdinę darbo organizavimo sistemą. Ši realizacija parodė, kaip galima sukurti sąsają, kuri padėtų kurti taisykles, bei valdyti jų parametrus.

Literatūra

- [1] Blaze Advisor documentation. <http://www.fairisaac.com>.
- [2] Business Rules Group, Defining Business Rules: What Are They Really? (3rd ed.). 2000, URL: <http://www.BusinessRulesGroup.org>.
- [3] Chisholm, M. Business Rules Evangelist: Business Rules and black boxes. DMReview.com, Prieiga per internetą: <http://www.dmreview.com/article_sub.cfm?articleId=8163>.
- [4] Ronald G. Ross, „Principles of the business rule approach“, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2003, ISBN 0201788934
- [5] Ronald G. Ross, "The Point of Knowledge," Business Rules Journal, Vol. 11, No. 1 (Jan. 2010), Prieiga per internetą: <<http://www.BRCommunity.com/a2010/b515.html> >