

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

Vytautas Taujanskas

**Mokslo tyrimo darbų  
registravimo ir pateikimo PS**

Magistro darbas

Darbo vadovas  
doc. R. Butleris

Kaunas, 2004

KAUNO TECHNOLOGIJOS UNIVERSITETAS  
INFORMATIKOS FAKULTETAS  
INFORMACIJOS SISTEMŲ KATEDRA

TVIRTINU  
Katedros vedėjas  
doc. dr. R. Butleris  
2004 05 24

**Mokslo tyrimo darbų  
registravimo ir pateikimo PS**

Magistro darbas

Kalbos konsultantė  
Lietuvių kalbos katedros lektorė  
dr. J. Mikelionienė  
2004 05 24

Recenzentas  
doc. dr. V. Pilkauskas  
2004 05 24

Vadovas  
doc. dr. R. Butleris  
2004 05 24

Atliko  
IFM-8/2 gr. stud.  
V. Taujanskas  
2004 05 24

# Turinys

1.	Ivadas	6
2.	Turinio valdymo sistemų funkcionalumo ir technologinių sprendimų analizė	7
2.1	Kuriamo produkto rinkos analizė	7
2.2	TVS funkcijų analizė	8
2.3	Technologijų analizė	11
2.3.1	Servletai	11
2.3.2	Java ServerPages dinaminiai puslapiai	12
2.3.3	JavaBeans komponentai	13
2.3.4	Struts karkasas	14
2.4	Struts karkaso architektūra	14
2.4.1	Klasių pasiskirstymas pagal Model-View-Controller	17
2.4.2	Programuotojo kuriamų komponentų paskirtis	18
2.5	Case įrankiai Struts UML diagramoms kurti	19
2.5.1	Microsoft VISIO ir StrutsGUI	19
2.5.2	Taikomojo uždavinio realizacija	21
2.6	Programų konstravimo aplinkos Eclipse pritaikymas Struts karkasui	24
2.6.1	Struts projekto kūrimas	24
2.6.2	Automatizavimo galimybės	25
2.7	Uždavinio formuluotė	26
3.	Mokslo tyrimo darbų PS realizacija	28
3.1	Projekto kūrimo pagrindimas	28
3.2	Projekto apribojimai	28
3.3	Funkciniai reikalavimai	29
3.3.1	Veiklos sfera	29
3.3.2	Funkciniai reikalavimai	30
3.4	Nefunkciniai reikalavimai	30
3.5	Panaudojimo atvejų vaizdas	31
3.6	Loginis vaizdas	32
3.7	Proceso vaizdas	33
3.8	Paskirstymo vaizdas	34
3.9	Realizacijos vaizdas	34
3.10	Duomenų vaizdas	37
4.	Mokslo tyrimo darbų portalų tyrimas	39
4.1	Panašūs portalai	39
4.2	TVS funkcionalumas	41
5.	Sukurtos mokslo tyrimo darbų PS eksperimentinis vertinimas	44
6.	Išvados	48
7.	Literatūra	49
8.	Terminų ir santrumpų žodynas	50
9.	Priedai	51
9.1	Straipsnis Turinio valdymo sistemos mokslo tyrimo darbų registracijai.	51

9.2	Straipsnis. The portal of research and development projects for business and science integration.	57
9.3	Loginio vaizdo diagramos	67
9.3.1	Duomenų peržiūros procesas	67
9.3.2	Duomenų redagavimo procesas	68
9.3.3	Duomenų išsaugojimo procesas	69
9.3.4	Veiklos klasių diagrama	70
9.4	Projektų registravimo sistemos procesas	71
9.5	Taikomojo uždavinio struts-config.xml failas	72
9.6	Sukurtos programų sistemos vartotojo sąsajos vaizdai	73

## **SUMMARY**

Analysis of already created portals, which appeal to national research and development works presentation, are made. Lacks and advantages of existing portals are overlooked and considered for prototype development. National research and development portal functions are described and solution is presented.

Product was built as a specific content management system (CMS). The range of CMS products offers more than 10 different functions, but not all of them are necessity. The most useful functions are chosen and produced for this product.

The need of architecture sketch occurs, when creating CMS or running a CMS-like website. Separation from structure, contents and representation issue is described and solutions are provided.

## 1. Įvadas

Mokslo tyrimo darbų skelbimas ir sukurtų rezultatų pateikimas – būdas padidinti pasiektų rezultatų efektyvumą, paskleisti žinias visuomenėje, pritraukti naujų partnerių, rėmėjų ar užsakovų. Pasirinktas problemos sprendimo būdas – portalas mokslo tyrimo darbų registravimui ir pateikimui.

Buvo sukurta programinė įranga pateikė specifiniam duomenų karkasui pritaikytas standartines kūrimo, atvaizdavimo, redagavimo bei trynimo (CRUD – *Create-Retrieve-Update-Delete*) galimybes. Ši funkcionalumą pateikia standartinės turinio valdymo sistemos (TVS), tačiau dėl specifinės darbo duomenų struktūros, jų pritaikymas taptų sudėtingas. Taip pat buvo atlikta ir internetinių technologijų, pritaikomų turinio valdymo sistemų kūrimui analizė.

Darbo tikslas – aprašyti programų sistemos, kaip specifinės turinio valdymo sistemos, kūrimo procesą, pritaikant modelio, vaizdo, valdiklio (MVC – Model, View, Controller) šabloną. Aprašyti specifinės turinio valdymo sistemos kūrimą, naudojant modelio, vaizdo, valdiklio šabloną. Sukurtą sistemą įvertinti kitų projektų duomenims pateikti skirtų TVS kontekste.

Analizės dalyje aprašyti vartotojų tikslai, naudojantis iš dabartinių Europos Sąjungos narių sudarytos darbo grupės parengtais dokumentais [5]. Pasirinkus programų sistemą, kaip TVS modelį, buvo apibrėžtos šio tipo programinės įrangos funkcijos [4]. Renkantis kūrimo platformą, svarstyta keletas alternatyvų: PHP, .NET ir Java technologijos [15], [17]. Dėl plataus kūrimo, palaikymo priemonių ir komponentų pasirinkimo, lengvai pritaikomos atvirojo kodo koncepcijos, nepriklausomumo nuo platformos buvo pasirinkta Java technologija. Atlikta galimų JSP sprendimų analizė ir pasirinktas pateikties ir skaičiavimo logikos atskyrimo problemą [13] eliminuojantis sprendimas.

Projektinėje dalyje pateikiami fragmentai iš kūrimo metu sudarytų dokumentų: panaudojimo atvejų vaizdas, statinės programinės įrangos diagramos, reikšmingiausių PS procesų bendradarbiavimo diagramos, pateikiamas duomenų modelis.

Atliekant panašių portalų tyrimą, pateikiami palyginimų su panašiais produktais rezultatatai. Eksperimentinis tyrimas suteikė galimybę pagal įvestus programų sistemos duomenis atlikti analitinį sukurto produkto įvertinimą.

Prieduose pateikiami moksliniai darbo rezultatai – autoriaus straipsniai konferencijose.

## **2. Turinio valdymo sistemų funkcionalumo ir technologinių sprendimų analizė**

### **2.1 Kuriamo produkto rinkos analizė**

Pagal Europos mokslo darbų registravimo portalo ISTRTD reikalavimų nustatymo darbo grupės pateiktus dokumentus [5]:

- politikai;
- mokslo bei rinkos atstovai;
- mišri vartotojų grupė.

Tarp politikų pažymimi atsakingi už valstybės mokslo tyrimo darbų registravimą bei tvarkymą, valdžios atstovai susiję su mokslo grupėmis, atsakingi už Europos programą IST (IST – Informacinės visuomenės technologijos) bei už kitas Europos programas.

Tarp antrųjų darbo grupė išskiria atsakinguosius už mokslo institutų valdymą, universitetų bendruomenę ir individualius mokslininkus. Numatoma kad šios grupės atstovai turėtų būti susiję su rinka.

Likusiems priskiriami Europos IST programų kandidatai, bendrovių (daugiau mažų bei vidutinių) vadovai atsakingi už plėtrą, rinkodarą, bei investuotojai.

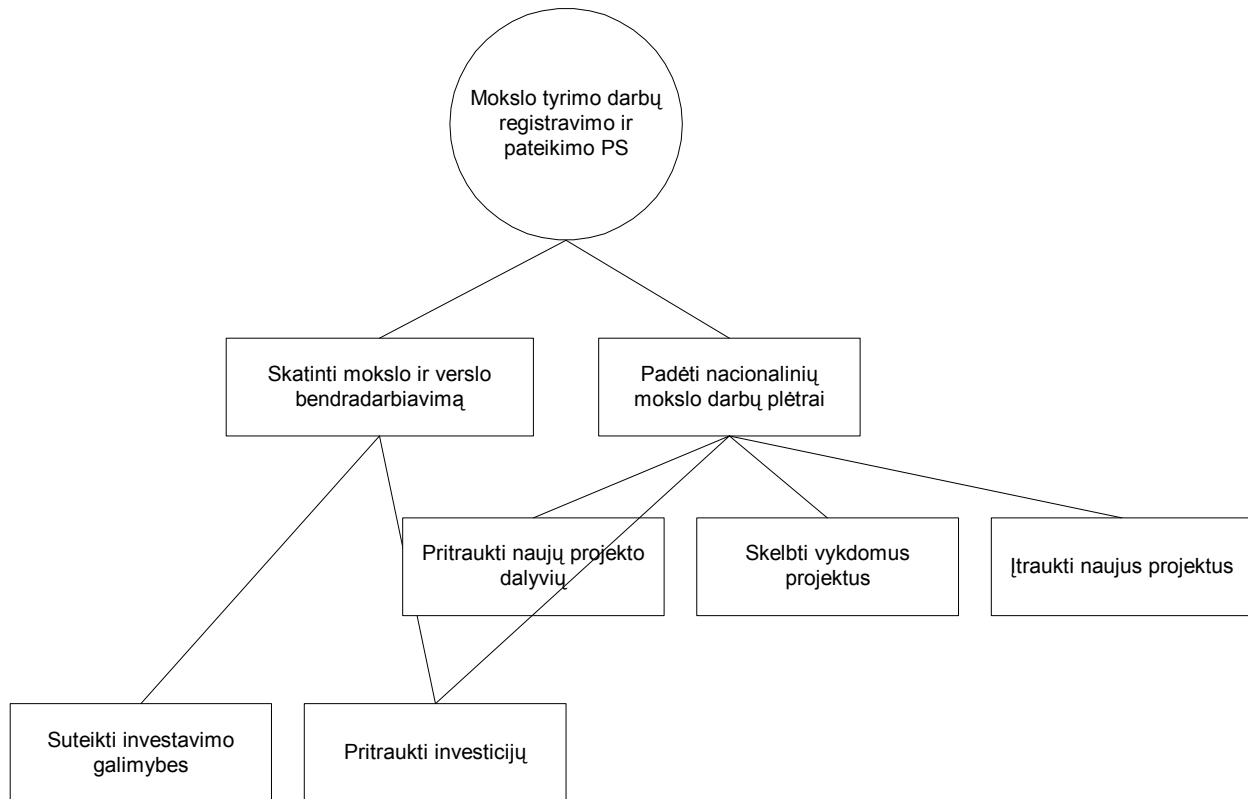
Mokslo tyrimo darbų programinė sistema skirta supažindinti su vykdomais ar planuojamais projektais, pritraukti projektų dalyviams, keistis informacija. Kiekvienai iš galimų vartotojų grupių numatomi skirtingi naudos aspektai.

Politikos atstovams sukurtas įrankis padės skirstyti mokslo tyrimo darbus pagal prioritetus, surasti geriausius darbų valdymo pavyzdžius, vykdyti nacionalinius mokslo darbų plėtojimo tikslus.

Mokslo atstovams pravers galimybė bendradarbiauti – plačiau susipažinti su panašios krypties vykdomais projektais, pabendrauti su panašios krypties projektų vykdytojais, pritraukti investicijų. Universitetams toks portalas padėtų atnaujinti mokymo planus pagal naujausius mokslo darbus panašioje kryptyje. Jauniems, dar neturintiems darbo srities, mokslininkams portalas padėtų ją pasirinkti.

Verslo atstovams mokslo projektų padėties žinojimas suteiktų žinių, kur pakreipti verslą, kad pasiekti daugiau naudos, užmegzti pelningų kontaktų.

Pagal [1] straipsnyje paskelbtus tyrimų duomenis sudarytas programų sistemos vartotojų tikslų modelis, kuris pavaizduotas šiame paveikslėlyje.



1 pav. Mokslo tyrimo darbų registravimo ir pateikimo programinės sistemos tikslų modelis[1].

## 2.2 TVS funkcijų analizė

Kadangi kuriama programų sistema teiks turinio valdymo paslaugas, buvo nuspręsta išanalizuoti turinio valdymo sistemų (TVS, angl. *Content management systems, CMS*) pritaikymo galimybes.

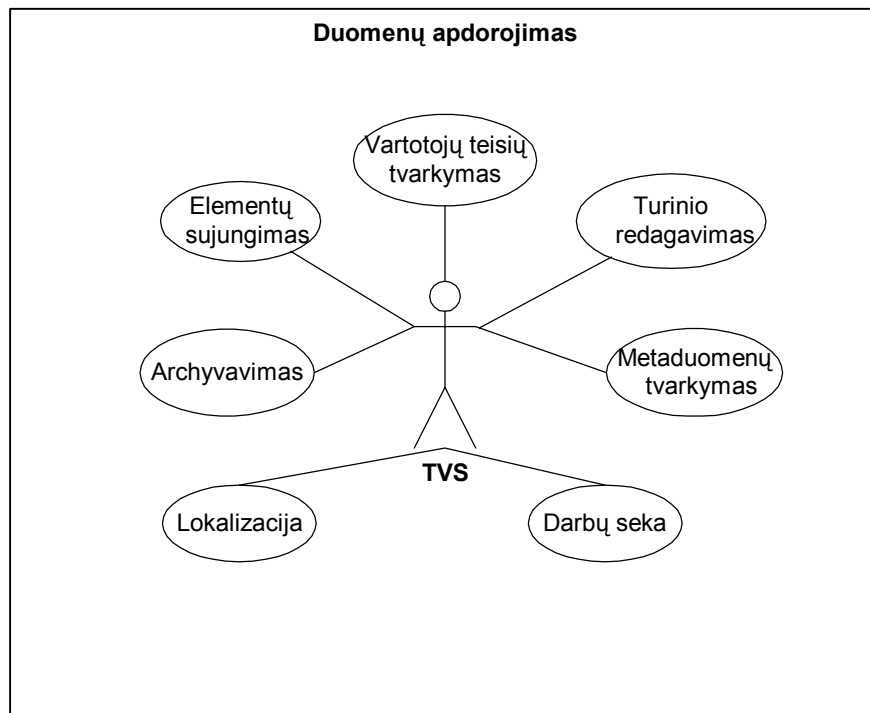
Turinio valdymo sistemų paslaugas galima išskirti į dvi aiškias kategorijas: duomenų apdorojimą ir duomenų pateikimą [4].

Apdorojimo fazėje (2 pav.) yra šios funkcijos:

- vartotojų teisių tvarkymas: kam bus prieinamos įvairios sistemos funkcijos;
- duomenų tvarkymas: turinio kėlimas į sistemą;
- metaduomenų tvarkymas;
- darbų seka, apibrėžianti turinio talpinimą, redagavimą ir patvirtinimą;
- atskirų TVS paslaugų sujungimas ir talpinimas į TVS, šablonų redagavimas;



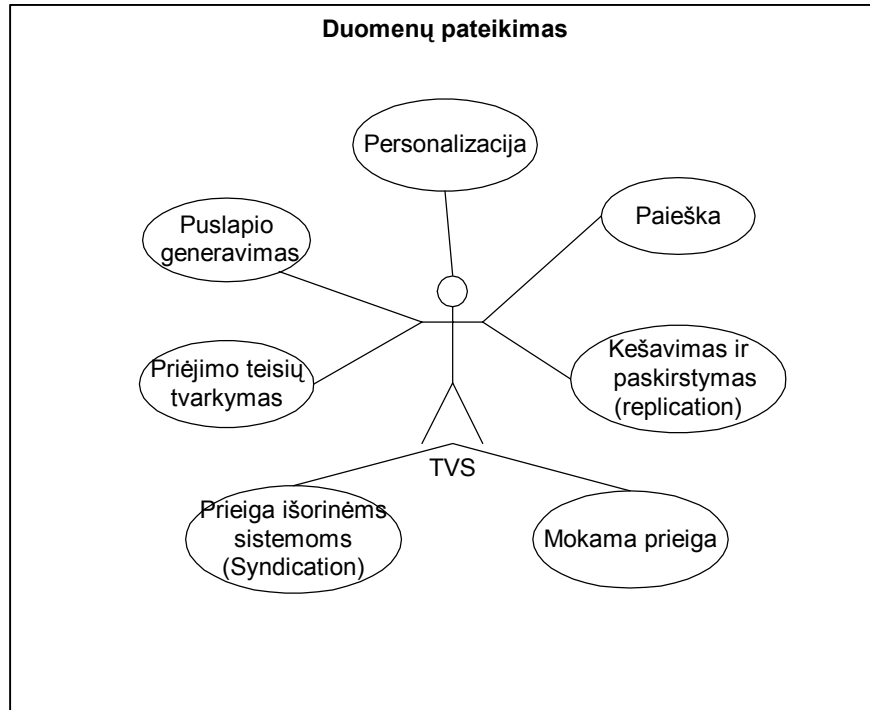
- archyvavimas: versijų kontrolė ir versijų valdymas;
- lokalizacija: programų sistemos pritaikymas atskiroms kalboms.



*2 pav. Turinio valdymo sistemos redagavimo dalies galimybės*

Duomenų pateikimas (3 pav.) apima šias TVS funkcijas:

- puslapio generavimą: pagrindinė TVS funkcija informacijos peržiūrai;
- paiešką;
- personalizaciją – kiekvienam vartotojui asmeninis TVS pritaikymas;
- priėjimo teisių tvarkymą – skirtingi vartotojai turi skirtingas informacijos peržiūros teises;
- prieigos apmokestinimas suteikia galimybes uždirbti turinio teikėjams;
- prieiga išorinėms sistemoms praplečia TVS matomumą;
- kešavimas ir paskirstymas pagreitina veikimą.



*3 pav. Turinio valdymo sistemos pateikimo dalies galimybės*

**Duomenų apdorojimo** paslaugos skirtos vidiniams sistemos vartotojui, kurio jurisdikcijoje yra duomenų įvedimo bei apdorojimo darbai. Saugaus prisijungimo užtikrinimas – vartotojo autorizacija ir teisių tikrinimas realizuojama per vartotojo grupes: kiekvienas vartotojas priklauso tam tikrai grupei, kuri apibrėžia skirtingas naudojimosi sistema teises. Autoriai pateikia medžiagą, redaktoriai ją patvirtina, o administratoriai gali redaguoti sistemos kodą, taigi keisti išvaizdą ir galimybes. Informacijos pateikimas TVS sistemai apibūdina darbų seką: pateikimas, redagavimas, patvirtinimas. Įdiegus redaguotos informacijos įvairių versijų saugojimą, pristatoma archyvavimo bei versijų valdymo galimybė. Versijas skirtingoms kalboms vadiname lokalizacija – pritaikymu atskiros lokalizacijos vartotojams.

**Duomenų pateikimas** remiasi puslapio generavimu. Naudojant stilių šablonus (stylesheets) tam pačiam duomenų šaltiniui galima gauti du rezultatus. Ši „vieno šaltinio-kelių rezultatų“ idėja yra pats svarbiausias pateikties logikos atskyrimo nuo veiklos logikos principas [14]. Informacijai pasiekti galima numatyti skirtingas priėjimo teises – įvedamas teisių tvarkymas ir išoriniams vartotojams, kurie prieina tik prie jiems skirtos informacijos. Tampa įmanomas mokamų paslaugų teikimas: priėjimas prie informacijos suteikiamas tik susimokėjusiems už peržiūrą naršytojams.

Personalizacijos paslauga didina svetainės patrauklumą – atvaizdavimas pritaikomas atsižvelgiant į vartotoją. Fiksuojami bent trys personalizacijos tipai:

- grupinė,
- asmeninė,
- savybėmis paremta.

Pirmuoju atveju pateikiamos skirtingos personalizacijos galimybės kelių tipų vartotojų grupėms. Asmeninės personalizacijos atveju kiekvienas naršytojas gali sau asmeniškai nusistatyti svetainės nuostatas. Įdiegus savybėmis paremtą personalizaciją, internetiniai agentai stebi vartotojo elgseną ir pagal tai pritaiko nuostatas[2].

Kitos funkcijos pagerina TVS patogumą, greitaveiką bei pasiekiamumą. Daug duomenų apimančiuose tinklalapiuose, informacijos pasiekimo patogumui praverčia paieška. Norint padidinti sistemos veikimo greitį naudojamas kešavimas. Sisteminei apkrovai išskaidyti naudojama keletas tarnybinių stočių, todėl galimas duomenų surinkimas iš skirtingų šaltinių. Informaciją taip pat galima perduoti kitiems portalams, pvz., naujienų santrauką XML formatu.

Išskirtos turinio valdymo sistemų funkcijos taikomos kaip bazinės savybės programų sistemos kūrimui.

## **2.3 Technologijų analizė**

Renkantis kūrimo platformą, svarstyta keletas alternatyvų: PHP, .NET ir Java technologijos. Galutinis pasirinkimas buvo Java technologijos dėl plataus kūrimo, palaikymo priemonių ir komponentų pasirinkimo, lengvai pritaikomos atvirojo kodo koncepcijos, nepriklausomumo nuo platformos. Tai patvirtino tiek subjektyvūs [17], tiek nepriklausomi tyrimų rezultatai [15]. PHP užleido pirmaujančias pozicijas dėl silpnesnio objektinio programavimo koncepcijos, didesnio primityvumo. .NET technologija pralaimėjo dėl kūrimo, palaikymo priemonių monopolio priklausančio „Microsoft“ ir dėl mažesnio nemokamų komponentų pasirinkimo.

Tačiau problema iškilo renkantis Java technologiją, kuri patogiausiai išspręstų specifinės turinio valdymo sistemos pritaikymą projektui.

### **2.3.1 Servletai**

Java technologija, pritaikoma internetui, yra servletai –programų serverį praplečiantys komponentai.

Prieš atsirandant JSP technologijai, serverio komponentai atlikdavo skaičiavimus, o vartotojo informaciją generuodavo spausdindami rezultatus ir jų atvaizdavimą formuojančius HTML elementus. Žemiau pavaizduotas servleto fragmentas, kuris atspausdina lentelės elementus.

```
MyTableData tableData = MyDAO.queryData();
PrintWriter writer = response.getWriter();
writer.println("<table border=\"1\">");
for (int i = 0; i < tableData.getData().length; i++) {
    writer.println("<tr>");
    writer.println("<td>");
    writer.println(tableData.getData()[i]);
    writer.println("</td>");
    writer.println("</tr>");
}
writer.println("</table>");
```

Gaunamas sugeneruotas rezultatas:

Item 0
Item 1
Item 2
Item 3

4 pav. Servleto sugeneruotas HTML

Nepaisant to, kad skaičiavimai visiškai susimaišę su duomenų išvedimu, programos kodas turi būti perkompiliuojamas ir įdedamas (*deploy*) apdorojimui.

Šio mažo pavyzdžio atveju tai neatrodo didelė problema, tačiau kai informacinę sistemą sudaro didelė apimtis panašaus kodo, pakeitimų atlikimas sukelia daug darbo sąnaudų.

### 2.3.2 Java ServerPages dinaminiai puslapiai

Su JSP, atsirado galimybė skaičiavimus įterpti tarp HTML elementų, taip sumažinant vaizdavimo ir skaičiavimų tarpusavio ryšius. Taip pat nebereikia kiekvieną kartą kompiliuoti ir įkelti kodo, atlikus pakeitimus.

Java veiklos logika, naudojant skriptletus puikiai išikomponuoja HTML dokumente, aprašančiame duomenų atvaizdavimą, tačiau išlieka nepatrauklus programos kodo įsiterpimas tarp tagais aprašomų vaizdavimo elementų, kaip kad matyti sekančiame fragmente, taip pat atspausdinančiame lentelės elementus.

```

<%
  MyTableData tableData = MyDAO.queryData();
%>
<table border="1">
  <%
    for (int i = 0; i < tableData.getData().length; i++) {
  %>
  <tr>
    <td>
      <%= tableData.getData()[i] %>
    </td>
  </tr>
  <% }%>
</table>

```

5 pav. JSP: kodas ir vaizdavimo elementai

Didelio projekto atveju, atsiranda daugiau veiklos logikos JSP puslapiuose, todėl parašytas kodas sunkiau skaitomas ir pasigendama didesnio veiklos logikos ir vaizdavimo logikos atskyrimo.

Norint išspręsti skaičiavimų ir atvaizdavimo problemą, reikia JSP naudoti kartu su architektūriniais šablonais.

### 2.3.3 JavaBeans komponentai

Galima dalį kodo iš JSP perkelti į Java komponentus. Vieni iš tokių komponentų yra JavaBeans klasės. Tai standartiniai objektai kurie susideda iš:

- klasės kintamųjų
- metodų *setXxx()* ir *getXxx()* kiekvieno kintamojo reikšmei gauti ir nustatyti
- konstruktorius be parametrų

Naudojant šias klases, sumažėja veiklos logikos perduodant duomenis tarp JSP puslapių, kurių vienas yra HTML forma, o kitas tos formos duomenis apdorojantis komponentas. JavaBeans komponentai gali būti naudojami ir su Struts karkasu.

Sudėtingesnės architektūros Enterprise Java Beans objektų panaudojimas aprašomas [3]. Veiklos objektai atlieka operacijas su duomenų objektais ir pateikia duomenis atvaizdavimui – redaguoti ir išvesti vartotojui. EJB pagalba pasiekiami duomenys, užuot programos tekste rašius SQL užklausas [11]. Taip sukuriamas turtingas programos veiklos sluoksnis.

Tačiau panaudojant šį architektūrinį modelį, vis tiek išlieka skaičiavimų ir duomenų atskyrimo problema.

### 2.3.4 Struts karkasas

Tai vienas iš pirmųjų karkasų pritaikytų JSP, ir:

- pateikia tagų bibliotekas, atliekančias logines, iteracijų operacijas, bei darbą su duomenimis, kas išsprendžia JSP įskaitomumą. *Bean* ir *logic* tagų bibliotekos naudojamos šioms problemoms išspręsti;
- naudoja MVC architektūrą, kuri atskiria skaičiavimo ir pateikimo logiką.

Vaizdavimo ir skaičiavimų atskyrimo efektyvumas pavaizduotas spausdinti lentelės elementus naudojant Struts:

```
<table border="1">
  <logic:iterate id="data" name="tableData"
property="data">
  <tr>
    <td>
      <bean:write name="data"/>
    </td>
  </tr>
</logic:iterate>
</table>
```

6 pav. Duomenų išvedimas į HTML lentelę, naudojant Struts.

Čia ir ankstesniuose pavyzdžiuose nėra pateiktas JavaBeans komponento *MyTableData* kodas. Šios klasės kintamasis *tableData* naudojamas duomenims išvesti.

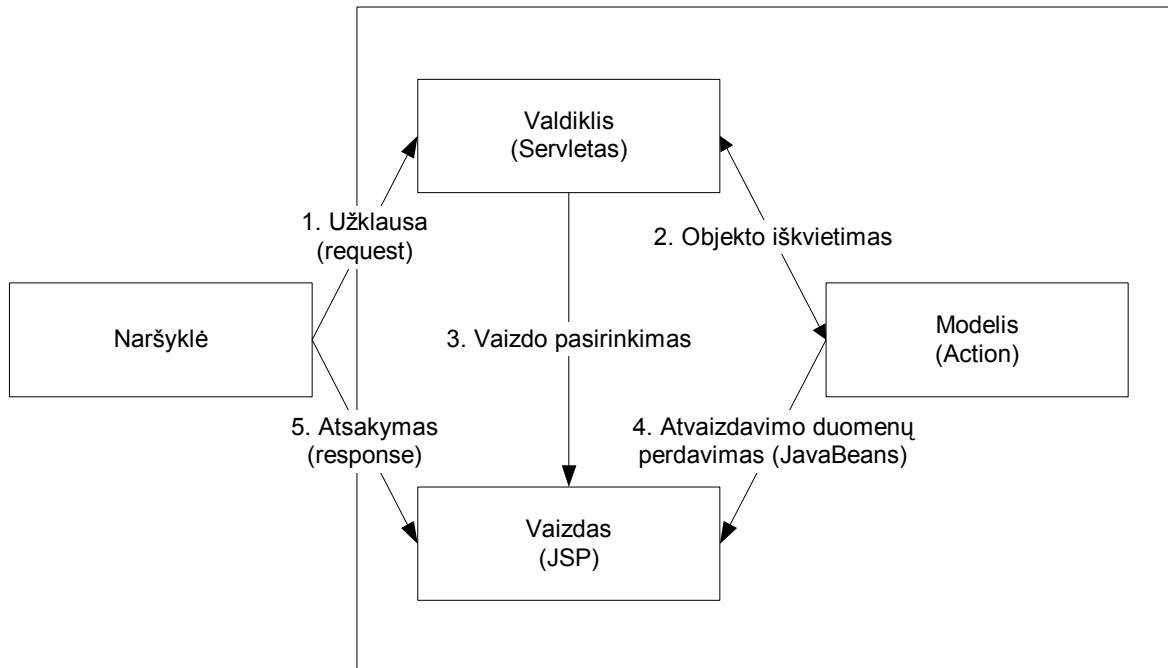
JSP puslapyje, nenaudojant Struts, tiesiogiai kuriamas šio tipo kintamasis ir kreipiamasi į jo metodus, kad išvesti duomenis, o naudojant Struts tagų bibliotekas, pakanka kaip tago parametą perduoti kintamojo vardą, taip visiškai išvengiant Java kodo JSP puslapyje.

Žinoma, ir šiame architektūriniame modelyje galima naudoti skriptletus įterpiančios veiklos logiką į JSP puslapius, tačiau tokia praktika nėra rekomenduojama.

## 2.4 Struts karkaso architektūra

Programų sistemos kūrimui pasirinktas „Struts“ karkasas, todėl darbui su juo buvo atlikta detali šio paketo analizė.

Žemiau pavaizduota komponentų sąveikos diagrama paaiškina tarpusavio sąveikas.



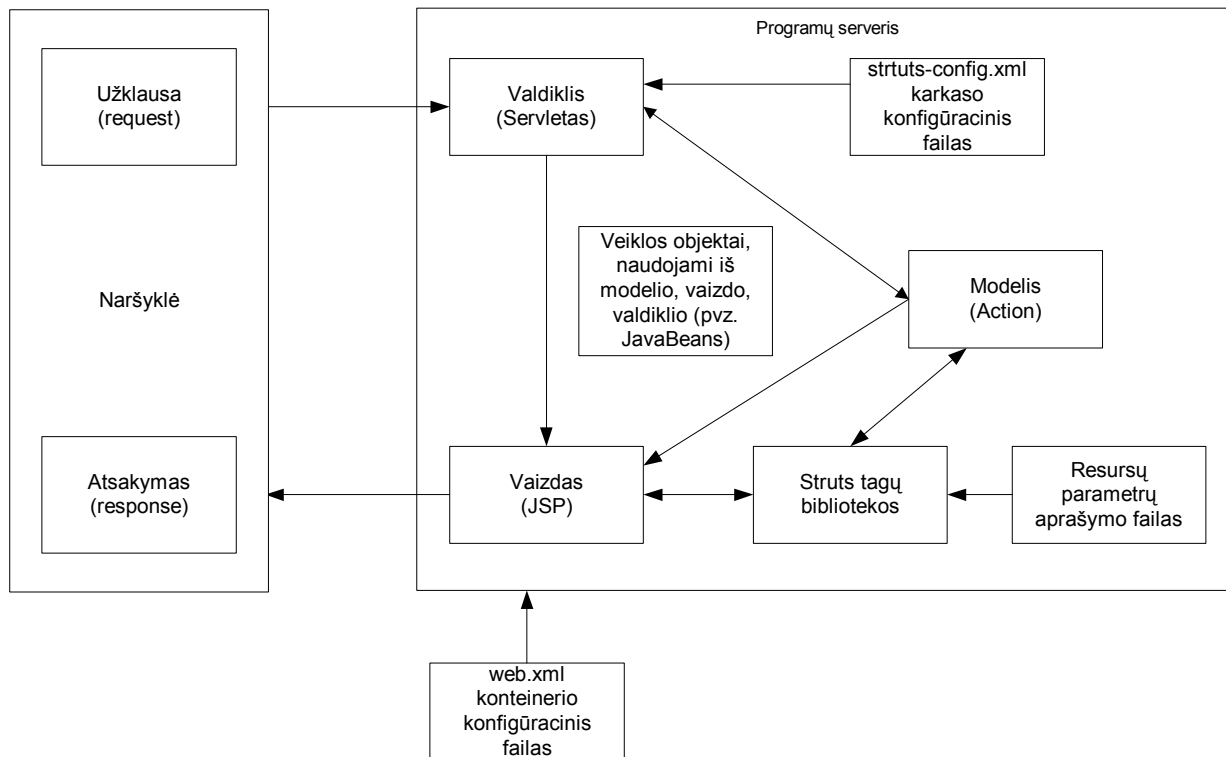
7 pav. Struts komponentų sąveika

Kliento programa (naršyklė), kreipiantis į valdiklio funkcijas atliekantį servletą, šis iškviečia reikiamą veiklos komponentą (pvz. JavaBean) ir parenka atvaizdavimo komponentą JSP, į kurią perduodami atvaizdavimui skirti duomenys iš JavaBean. Suformuoti rezultatai atvaizduojami vartotojui.

Dar smulkesnė komponentų schema žemiau pateiktame paveikslėlyje parodo tas pačias sąveikas, tik čia matyti kiti reikšmingi programų sistemos elementai:

- naršyklei kreipiantis į programų sistemą, užklausa (*request*) kreipiasi į kontrolierį, kuris naudojami struts-config.xml konfigūraciniu failu veiklos komponentų (*Model*) iškvietimui ir atvaizdavimo (*View*) parinkimui;
- programų sistemos būseną išsaugoma veiklos (*Model*) komponentuose;
- atvaizdavimo komponentai naudoja tagų bibliotekas (*tag libraries*), kurios tekstinę informaciją gali pasiimti iš resursų failų (tokia schema naudojama dirbant su I18N);
- konteinerio konfigūracija ir servletų sankirtos (*mappings*) aprašomos web.xml faile.

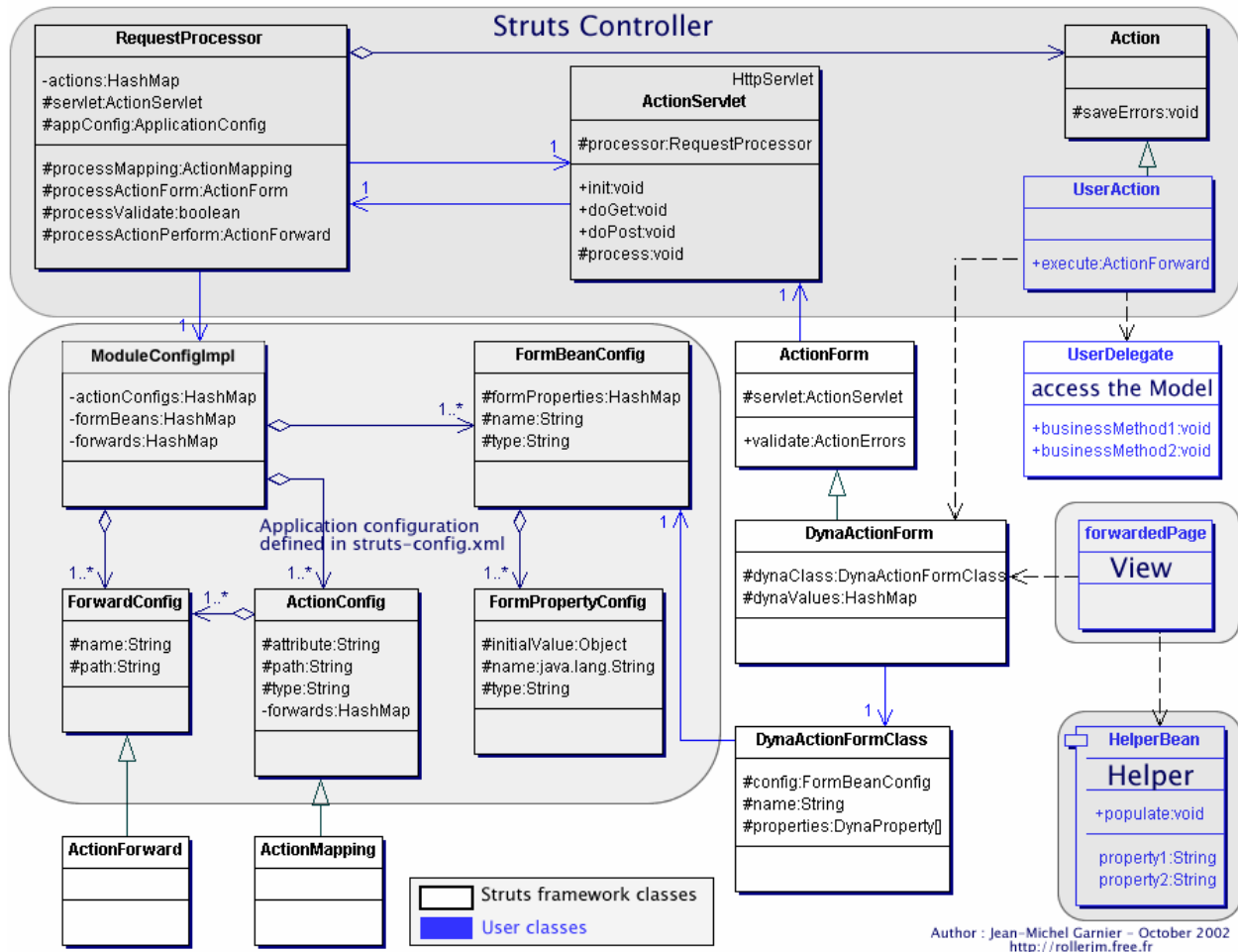
Patalpinus programų sistemą servletų konteineryje pirmiausia nuskaitomas šis failas ir Struts servletų apdorojimui priskiriamos tam tikro tipo užklauskos. Dažniausiai užklauskos, kurios baigiasi *.do* priskiriamos Struts apdorojimui.



8 pav. *Detalesnė Struts komponentų sąveikos diagrama [9]*

Dabar galima pereiti prie Struts karkaso architektūrinės schemos, kuri pavaizduota sekančiame paveikslėlyje.





9 pav. Struts paketo architektūros schema [16]

## 2.4.1 Klasių pasiskirstymas pagal Model-View-Controller

Šioje schemoje aiškiai pavaizduota Struts karkaso kontrolerio dalis: *RequestProcessor*, *ActionServlet*, *Action* ir *UserAction* klasės.

Taip pat aiškiai išskirtos klasės, kurių konfigūracija aprašoma aukščiau paminėtame *struts-config.xml* faile: *ModuleConfigImpl*, *FormBeanConfig*, *ForwardConfig*, *ActionConfig* ir *FormPropertyConfig*. Pagal veiklos prasmę – konfigūracijos nuskaitymas ir vartotojo užklausų apdorojimas – jas galima priskirti kontroleriui. Tačiau to nereikėtų griežtai daryti.

Modeliui priskiriamos veiklos logiką atliekancios *ActionForm*, *DynaActionForm*, *DynaActionFormClass*, *UserDelegate*, *HelperBean* klasės. Jos naudojamos duomenims nuskaityti, perduoti, skaičiavimo veiksmams atlikti. Dvi pastarosios yra programuotojo kuriami komponentai.

Programuotojas taip pat sukuria ir *UserAction* klases, paveldinčias karkaso *Action* klasę, priskiriamą kontrolieriui.

Atvaizdavimo dalį (View) sudaro *forwardedPage* arba JSP komponentai, kuriuos taip pat kuria programuotojas. Programuotojo kuriami komponentai schemeje pažymėti mėlynai.

## 2.4.2 Programuotojo kuriamų komponentų paskirtis

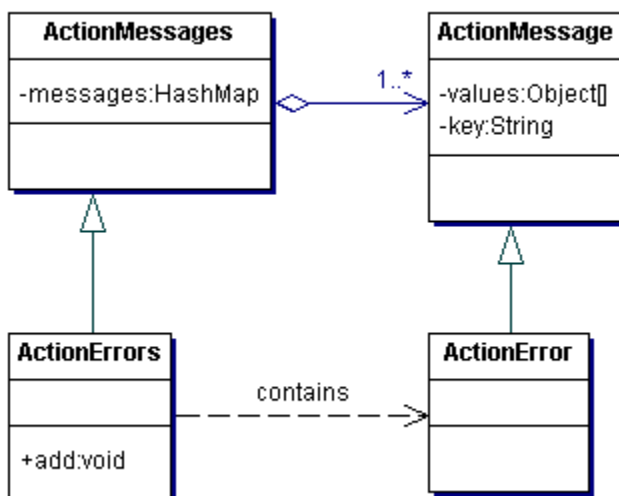
*ActionForm* klasės atspindi duomenų formą aprašytą atitinkamos formos laukuose. *DynaActionForm* būseną automatiškai nustatoma pagal atitinkamus laukus iš *HttpServletRequest* objekto.

Perduodant duomenis iš formos, paspaudus vienokią ar kitokią nuorodą informacinėje sistemoje išskviečiami *UserAction* klasės metodai *execute*. Jie apdoroja HTTP protokolo užklausos antraštę (*HTTP request header*) siunčiamą iš vartotojo naršyklės į tarnybinę stotį *request*’ą ir suformuoja atitinkamą HTTP protokolo atsakymo antraštę (*HTTP response header*). Klasės metodas gražina *ActionForward* objektą nurodantį, kur nukreipti kontrolę.

Šioje klasėje taip pat yra metodas *saveErrors*, kuris išsaugo specifinių klaidų kodus atitinkamoje užklausoje (request).

Veiklos logikai atlikti naudojamos pagalbinės *HelperBean*, *UserDelegate* klasės duomenų saugojimui, perdavimui.

Patogus ir klaidų pranešimų formavimas.



10 pav. Klaidų pranešimų valdymas[16]

*ActionMessages* – *ActionMessage* kolekcija

*ActionMessage* inkapsuliuoja vartotojo pranešimą, su klaidos kodu I18N (internacionalizacijos) palaikymui

*ActionErrors* paveldi *ActionMessages* ir yra *ActionError* kolekcija

*ActionError* inkapsuliuoja klaidų pranešimus

Ryšiai tarp šių komponentų apsiraso Struts karkaso konfigūraciniame XML faile.

## 2.5 Case įrankiai Struts UML diagramoms kurti

„Apache Struts yra de-facto Java 2 Enterprise Edition karkasas“, teigia IBM („IBM and Apache Struts“). <http://missa.bus.csupomona.edu/Missa/Misc/IBMANDApacheStrutsExternal.ppt>). Todėl nenuostabu, kad daugelis CASE ir IDE įrankių suderinami su „Struts“ architektūros programomis.

Renkantis kūrimo įrankius darbu buvo laikomasi šių kriterijų:

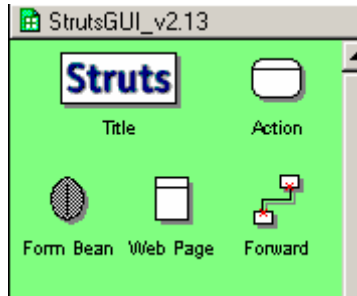
- paplitimas rinkoje;
- nemokamumas;
- patogumas naudoti.

Buvo iširtos Rational Rose, Microsoft VISIO CASE įrankių galimybės, bei Eclipse programavimo aplinkos galimybės. Trumpai aprašomos geriausiai suderinamų su „Struts“ paketu įrankių (Microsoft VISIO ir Eclipse ) galimybės.

### 2.5.1 Microsoft VISIO ir StrutsGUI

Visio – tai ne vien UML projektavimui skirta priemonė, bet diagraminio vaizdavimo sprendimas, skirtas tiek technikos specialistams, tiek ir verslininkams. Mus domina VISIO praplėtimas– pritaikymas Struts diagramoms kurti. Standartiškai galima susikurti UML modelio diagramą, kurioje rasime svarbiausius UML komponentus, tačiau norint pritaikyti Struts karkasui, reikia pačiam sukurti komponentų stereotipus arba susirasti praplėtimą. Kadangi kuriamos architektūros programa apima ne vien standartines objektines klases, tačiau joje naudojami ir JSP tinklalapiai, specifinės Struts *UserAction* klasės ir kita, todėl kodo generavimas ir reversinė inžinerija įmanoma tik panaudojus specifinę taikomąją programinę įrangą.

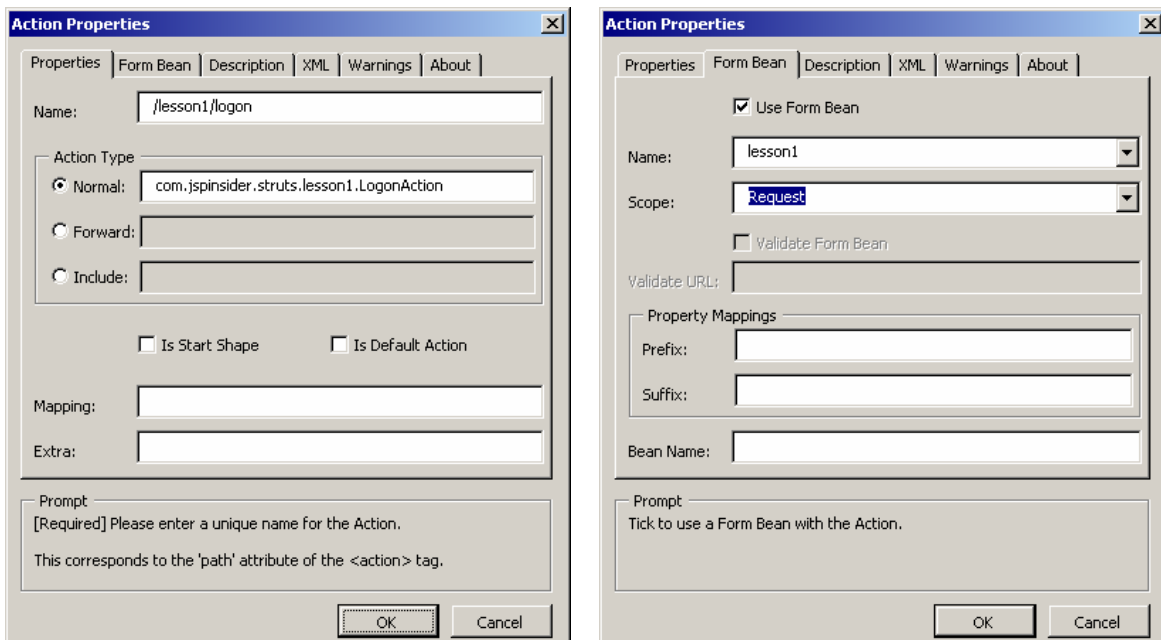
Tokį paketą, pavadintą „StrutsGUI“ galima parsisiųsti adresu <http://www.alienfactory.co.uk/strutsgui/>. Įdiegę paketą, VISIO įrankių bibliotekoje randame keletas specifinių įrankių:

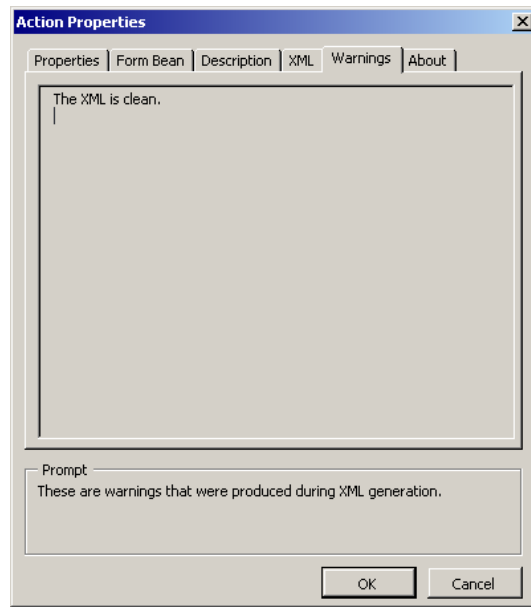
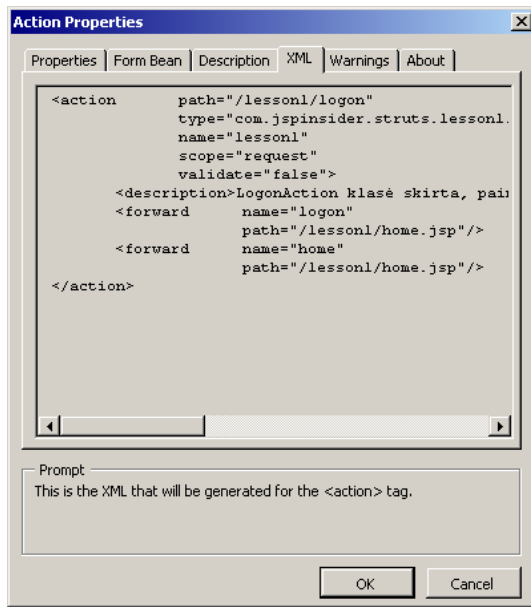


11 pav. StrutsGUI išplėtimo įrankiai

- *UserAction* klasėms naudojamas *Action* įrankis;
- *FormBean* objektas;
- JSP puslapis;
- komponentų sujungimas Forward ryšiui pažymėti.

Į darbo lauką “nusinešę” komponentą “Action” – galima jam nustatyti specifines savybes. Kiekvienas komponentas turi specifines savybes, kurias pavaizduotos sekančiame paveikslėlyje.



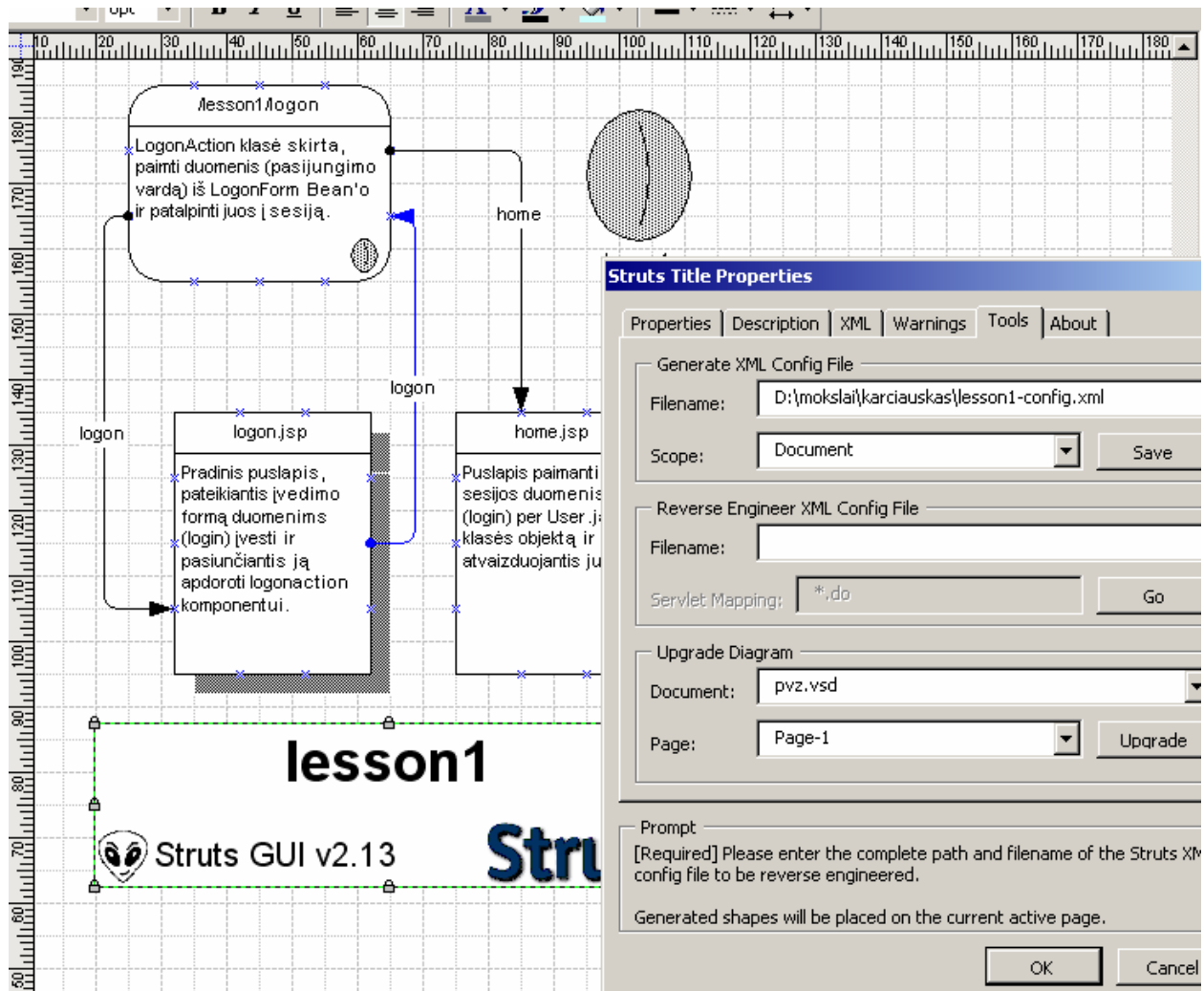


12 pav. StrutsGUI išplėtimo įrankių savybės (properties)

Properties lange atliekami įvairūs nustatymai. Čia nurodomas “Action” komponento vardas, kas reiškia informacinės sistemos adresą (URL). Veiksmo tipas (action type) nustato programuotojo sukurtos klasės vardą (normal), persiuntimą į kitą puslapį (forward) ir įterpimą (include). Form Bean lange galima nustatyti formos klasę (Form Bean), kuri naudojama Action klasėje duomenims paimti iš tinklalapio. Dokumente sukurtų formos klasių vardai automatiškai išvedami pasirinkimui. XML lange pademonstruojamas sukurto komponento XML aprašymas Struts configūraciniame faile *struts-config.xml*. Warnings lange pademonstruojami galimi XML konfliktai, jei tokių yra.

## 2.5.2 Taikomojo uždavinio realizacija

Projektuojant programų sistemos architektūrinį vaizdą, panaudojome visus komponentus. Žemiau pateiktas vartotojo prisijungimo panaudojimo atvejo realizacija, matoma VISIO darbiname lange.



13 pav. Taikomojo uždavinio realizacija su VISIO Struts GUI

Į VISIO dokumentą įkėlus reikalingą komponentą, reikia nurodyti jo parametrus. Daugiausia parametrų reikia aprašyti **Action** komponentui: komponento aprašymas, susiejamo formos komponento (Form Bean) pavadinimas. Formos komponentams reikia nurodyti formos laukų pavadinimus ir tipus. JSP puslapiams reikia nurodyti puslapio pavadinimą. Komponentus reikia sujungti Forward ryšiu, kuriam parenkamas ryšio tipas. Formų komponentai ir veiksmai (Action) tarpusavyje jungiami susiejimu, kuris atliekamas per veiksmų parametrus.

Tačiau architektūros diagramą reikia pradėti įkeliant komponentą – programų sistemos antraštę (Title). Jos parametruose pateikiami pagrindiniai kuriamos programų sistemos parametrai. Be to, per antraštę atliekamas *struts-config.xml* failo kodo generavimas ir atvirkštinė inžinerija pagal šį failą.

Nubraižius diagramą pasirinktam panaudojimo atvejui buvo sugeneruotas konfigūracinis failas (pateikiamas prieduose). Tai labai svarbi CASE įrankio savybė, nes sukurtas konfigūracinis failas detaliai aprašo sistemos komponentus, jų tarpusavio ryšius.

Kreipiant didžiausią dėmesį į generavimo savybes, VISIO buvo palygintas su kitu CASE įrankiu – Rational Rose. Šiam taip pat buvo įdiegtas praplėtimas, pritaikytas Struts.

**Lentelė 1 Rational Rose, Microsoft VISIO praplėtimų „Struts“ karkasui palyginimas**

Savybių palyginimas	Rational Rose praplėtimas			Microsoft VISIO praplėtimas		
	Kūrimas	Generavimas	Atvirkštinė Inž.	Kūrimas	Generavimas	Atvirkštinė Inž.
Struts karkaso klasės	√	√	√	√	×	√
JSP	±	×	×	√	×	√
Veiklos logikos klasės	√	√	√	±	×	±
XML konfigūraciniai failai	±	×	×	±	√	√

Rational Rose paketas pateikia turtingą struktūrą, paaiškinančią Struts karkaso hierarchiją ir ryšius, tačiau tai nei kiek nepalengvina kūrimo proceso, nes sunkiausia kūrimo užduotis – ryšiai tarp komponentų, lieka neautomatizuota. Maža to, paveldėję Struts karkaso klases savo susikurtiems komponentams, kurie praplečia karkasą, pritaikydami funkcionalumą kuriamai informacinei sistemai, turime dar ir numatyti naujai susikurtų komponentų struktūrą, t.y. kintamuosius ir metodus, kuriuos turime perdengti ar sukurti. Kodo generavimo ir atvirkštinės inžinerijos galimybės veikė puikiai, pademonstruodamos CASE įrankio korektiškumą.

Taigi šiam įrankių rinkiniui pritrūko esminės savybės – kuriamų diagramų automatizuotumo, specifinių Struts ryšių tarp komponentų automatizacijos ir pilno įrankių rinkinio informacinių sistemų kūrimui. Pažymėtinas teigiamas kodo generavimo ir atvirkštinės inžinerijos funkcionalumas.

Tuo tarpu visiškai priešinga padėtis VISIO praplėtime, kur nebuvo gausaus kuriamų komponentų pasirinkimo, tačiau parinkti įrankiai ir atitinka svarbiausius kuriamos informacinės sistemos architekto poreikius. Mažai dėmesio skirta laisvos architektūros veiklos logikos klasėms, tačiau jų funkcionalumą gali atlikti Struts karkasui priskiriami Forms Beans komponentai arba galima panaudoti kitų įrankių rinkinių (stencils) priemonėmis. Norint nubraižyti XML konfigūracinius failus, taip pat reikėtų rasti aplinkinius kelius, tačiau to daryti nebūtina, nes svarbiausias XML konfigūracinis failas sukuriamas ir nuskaitomas tiesioginės ir

atvirkštinės inžinerijos būdu. Galima pasigesti Java programinio kodo generavimo galimybės, tačiau sukurtas struts-config.xml failas kompensuoja šį trūkumą. Maža to, panaudojant šį failą ir kitas darbo su Struts priemones galima sugeneruoti trūkstamą programos kodą. O atvirkštinės inžinerijos keliu kuo puikiausiai nuskaitomi visi aprašyti komponentai.

Atlikto tyrimo metu programavimo automatizavimui rekomenduojamas VISIO programinis paketas ir StrutsGUI praplėtimas.

## **2.6 Programų konstravimo aplinkos Eclipse pritaikymas Struts karkasui**

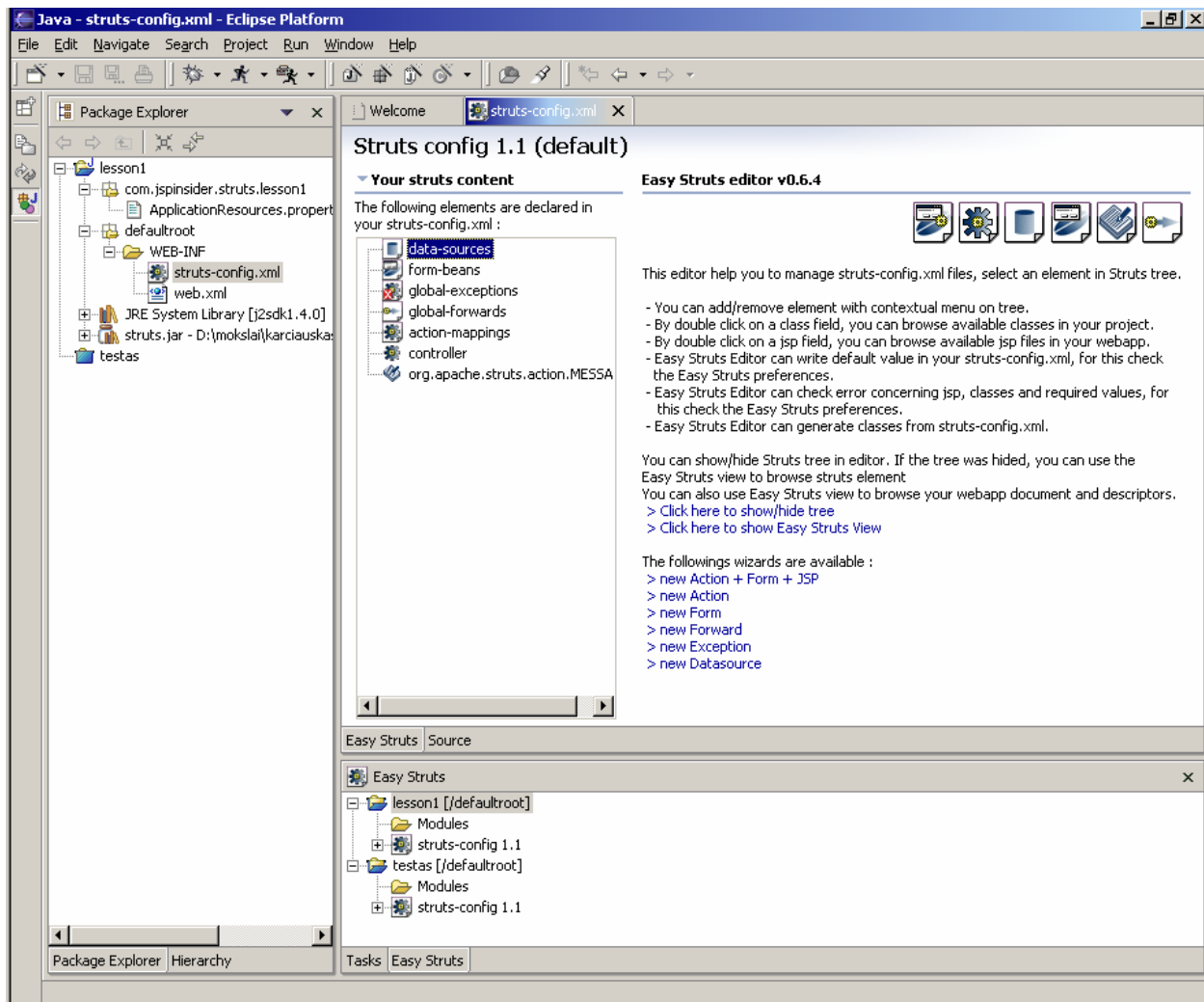
Eclipse programavimo aplinka (IDE) pasirinkta dėl universalumo ir plataus pritaikymo, bei paplitimo tarp vartotojų. Tai yra atvirojo kodo produktas pasiekiamas adresu [www.eclipse.org](http://www.eclipse.org). Struts komponentų generavimo galimybė kuo puikiausiai įmanoma pasiekti naudojant šią programavimo aplinką ir jos praplėtimą “Easy Struts”.

### **2.6.1 Struts projekto kūrimas**

Sukūrus naują Java projektą, reikia jo nustatymuose (project->properties->easy struts->add struts support) pažymėti apie Struts projekto kūrimą ir pasirinkti karkaso nustatymus kuriamam produktui.

Atlikus pakeitimus, į projekto bibliotekas įkeliamos Struts bibliotekos, projekto katalogų struktūra pritaikoma kuriamam projektui – Struts informacinei sistemai. Sukuriamas struts-config.xml failas. Projekto medyje išsirinkus (Pelė->2x2K) šį failą pateikiamas automatizuoto kūrimo meniu.





14 pav. Eclipse Easy Struts praplėtimo automatizavimo pasirinkimas.

## 2.6.2 Automatizavimo galimybės

Matome, kad apačioje pateikiamas praplėtimo peržiūros medis. Kairėje atvaizduojami struts-config.xml failo elementai: data-sources, form-beans, action-mappings ir kiti. Automatizavimo požiūriu svarbiausia yra pateikti žyniai. Naudojantis jais, nesunkiai gauname sugeneruotą konfigūracinį failą struts-config.xml bei komponentų karkasus. Lieka tik suprogramuoti logiką.

Programavimo aplinka puikiai praneša apie vykusias klaidas, joje patogų naršyti po projekto resursus. Projekto valdymas apima ir patogią techninės informacijos generavimą (javadoc). Programos paleidžiamieji failai išsaugomi projekto konfigūracijoje.

Struts karkaso panaudojimas ypač patogus – Easy Struts praplėtimas pateikia komponentų generavimo žynių rinkinį, priemones specifinės architektūros klaidų sekimui, patogią navigaciją.

Nustebino puikios nemokamos programinės aplinkos ir jos praplėtimo savybės, kurios suteikia malonų ir patogų darbo su projektais paketą.

## 2.7 Uždavinio formuluotė

Atlikus rinkos analizę, buvo išsiaiškinti programų sistemos vartotojai, jų tikslai.

Kuriama programų sistema teiks turinio valdymo paslaugas, todėl buvo atlikta turinio valdymo sistemų (TVS, angl. *Content management systems* CMS) pritaikymo galimybių analizė. Kadangi kuriamos programų sistemos duomenų modelis yra specifinis, standartinių turinio valdymo sistemų pritaikymo atsisakyta, tačiau buvo toliau laikomasi turinio valdymo sistemos koncepcijos.

Renkantis kūrimo platformą, svarstyta keletas alternatyvų: PHP, .NET ir Java technologijos. Galutinis pasirinkimas buvo Java technologijos dėl plataus kūrimo, palaikymo priemonių ir komponentų pasirinkimo, lengvai pritaikomos atvirojo kodo koncepcijos, nepriklausomumo nuo platformos. PHP užleido pirmaujančias pozicijas dėl silpnesnio objektinio programavimo koncepcijos, didesnio primityvumo. .NET technologija pralaimėjo dėl kūrimo, palaikymo priemonių monopolio priklausančio „Microsoft“ ir dėl mažesnio nemokamų komponentų pasirinkimo.

Tačiau problema iškilo renkantis Java technologiją, kuri patogiausiai išspręstų specifinės turinio valdymo sistemos pritaikymą projektui. Viena iš problemų yra ta, kad vartotojo sąsaja turi tendenciją keistis daug dažniau, nei duomenų saugojimo sistema. Kita problema, susijusi su duomenų ir vartotojo sąsajos susijungimu, yra ta, kad veiklos logika reikalinga ne tik duomenų perdavimui [13].

Šias problemas išsprendžia MVC šablonas, o Javai yra internetiniams sprendimams *de-facto* standartu tapęs karkasas “Struts”.

Tai vienas iš pirmųjų karkasų pritaikytų JSP, ir:

- pateikia tagų bibliotekas, atliekančias logines, iteracijų operacijas, bei darbą su duomenimis, kas išsprendžia JSP įskaitomumą. Bean ir logic tagų bibliotekos naudojamos šioms problemoms išspręsti;
- naudoja MVC architektūrą, kuri atskiria skaičiavimo ir pateikimo logiką.

Programinės įrangos kūrimui buvo pasirinkti ir kiti atvirojo kodo produktai:

- “Ant” – kompiliavimo įrankis;
- “Eclipse” – programavimo aplinka;
- “Tomcat” – programų serveris;
- anksčiau minėtas karkasas “Struts”.

**Inžinerinė problema - sukurti specifinę turinio valdymo sistemą projektų registravimui naudojant interneto technologijas, atskiriančias pateikties logiką nuo veiklos logikos.**

### **3. Mokslo tyrimo darbų PS realizacija**

Šiame skyriuje aprašomi kuriamos programų sistemos reikalavimai, apibrėžiama veiklos sfera. Taip pat aprašomas architektūrinis projektas įvairiais aspektais. Naudojant RUP proceso rekomendacijas buvo sudaryti šie sistemos vaizdai:

- Panaudojimo atvejų vaizdas. Aprašomi sistemos aktoriai ir jų veiksmai.
- Loginis vaizdas. Pateikiama dalykinės srities klasių diagrama, sistemos objektų sąveikos.
- Proceso vaizdas. Atvaizduojamos objektų scenarijų ir bendradarbiavimo diagramos.
- Paskirstymo vaizdas. Pateikiama programų sistemos paskirstymo diagrama.
- Realizacijos vaizdas. Pateikiama programinių sistemos komponentų diagrama.
- Duomenų vaizdas. Pateikiama duomenų bazės diagrama.

Toliau pateikiamas glaustas reikalvimų ir architektūrinių vaizdų aprašymas, bei svarbiausios realizacijos metu sudarytos diagramos.

#### **3.1 Projekto kūrimo pagrindimas**

Mokslo tyrimo darbų programinė sistema skirta skelbti informaciją apie vykdomus projektus, pritraukti naujus projekto vykdytojus, investuotojus bei potencialius vartotojus.

Sistemoje pateikiama daug teminių laukų projekto informacijai. Registruotas projekto administratorius gali redaguoti įvestą informaciją.

Įvesta informacija dubliuojama kiekvienai programų sistemos kalbai. Daugiakalbė sistema praplečia tikslinę auditoriją neapsiribojant vien Lietuvos rinka. Aiški hierarchinė struktūra suteikia galimybę lengvai naršyti po temines projektų sritis.

#### **3.2 Projekto apribojimai**

Kadangi kuriamo produkto esmė yra mokslo tyrimo darbų (projektų) duomenys, neišvengiamai reikalinga papildoma informacija, kaip tapti produkto vartotoju, kur rasti informacijos apie investuotojus, mokslinių projektų rėmimo programas. Todėl internetinės svetainės forma – kaip bus pateiktas sukurtas produktas – suteikia galimybę jį papildyti paprasčiausiais interneto puslapiais.

Programinės sistemos vartotojas naudojasi naršykle, kad įvestų, redaguotų ir peržiūrėtų duomenis. Naršyklę turi būti suderinama su HTML 4.0, CSS 2.0 standartais. Vartotojo darbo

vieta yra mobili. Kadangi sprendimo apribojimuose nurodyti standartai, su kuriais turi būti suderintas sukurtas produktas, tai kliento pusei nekeliama jokie reikalavimai. Rinkoje paplitusios naršyklės naudojamos įvairiose platformose atitinka keliamus reikalavimus.

Programiniai moduliai, bei vartotojo sąsaja apdorojami servletų konteineryje. Servletų konteineris bus įdiegtas į kompiuterį su “Windows 2000 Server” OS. Tačiau pasirinktas servletų konteineris “Tomcat” pateikiamas įvairioms operacijų sistemoms, todėl įmanoma nesunkiai perkelti programinę sistemą į kitokios konfigūracijos serverį.

Duomenys saugojami DB valdymo sistemoje “Microsoft SQL 2000”. Duomenų bazių serveris įdiegtas į kompiuterį su “Windows 2000 Server” OS. Kadangi Microsoft SQL 2000 Server DBVS veikia tik “Windows 2000” operacijų sistemoje, todėl duomenų bazės migravimas galimas panašios konfigūracijos kompiuteriuose.

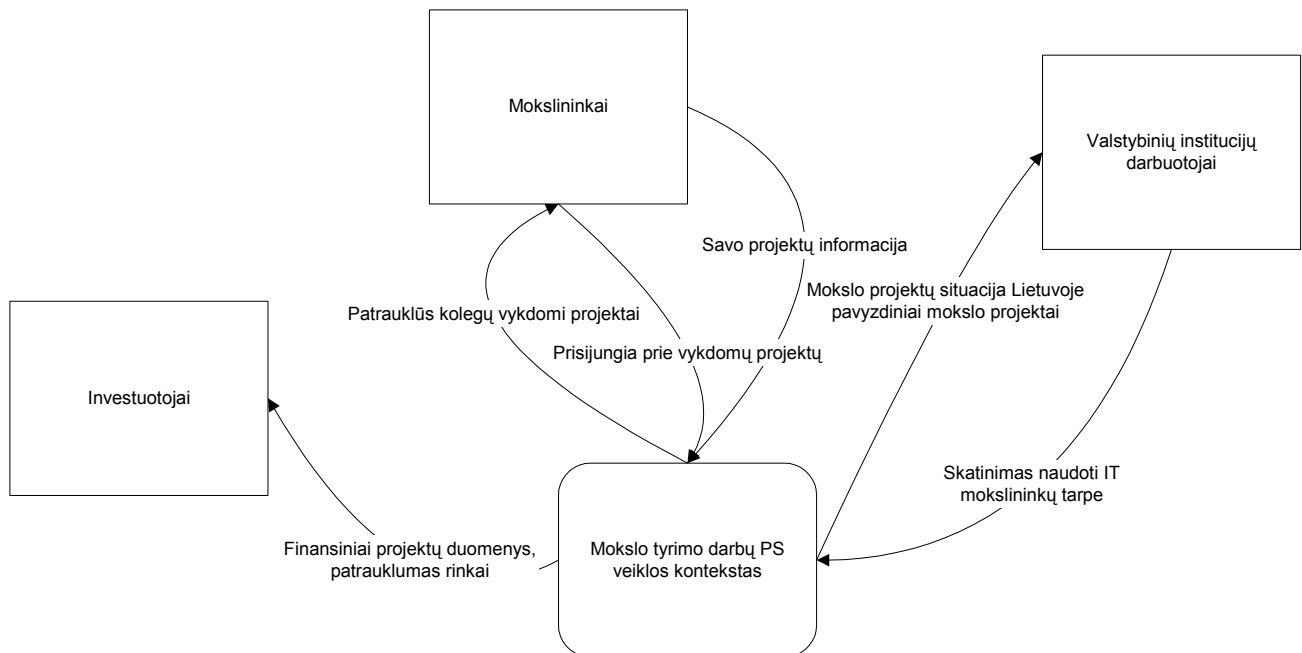
Sukurtas produktas gali būti integruotas į susijusios tematikos portalą. Galimas ir atvirkštinis variantas – projektų pateikimo sistema gali būti papildyta papildomais interneto puslapiais, kurie talpinami į servletų konteinerį.

Sistemos architektūra kuriama Model-View-Controller karkaso pagrindu. Tam naudojamas nemokamas šablonas “Struts”.

### **3.3 Funkciniai reikalavimai**

#### **3.3.1 Veiklos sfera**

15-ame paveikslėlyje pateikta projekto veiklos sfera. Joje pavaizduotas duomenų judėjimas. Kaip matome pagrindiniai duomenys, kuriuos naudoja sistemos vartotojai yra projektų duomenys. Juos pateikia ir gauna sistemos vartotojai.



15 pav. Veiklos konteksto diagrama

### 3.3.2 Funkciniai reikalavimai

Vartotojo sąsaja skiriamas į tris dalis. Vienoje (viršuje) pateikiamas sistemos kalbų sąrašas, kitoje (kairėje) projektų grupių sąrašas, kuriame pasirinkus atitinkamą grupę trečioje (dešinėje) pateikiamas jai priklausančių projektų sąrašas.

Rodant informaciją apie projektą: rodomas informacinio lauko pavadinimas, o po juo celėje parodoma informacija. Informacija apie projekto dalyvius pateikiama lentelės pavidale.

Kuriant projektą formose pateikiami visi projekto informaciniai laukai

Redaguojant projektą formose pateikiama jau įvesta projekto informacija visiems informaciniams laukams

Sistemos administratoriui suteikiama galimybė įvesti naujus sistemos vartotojus.

### 3.4 Nefunkciniai reikalavimai

Sistemos vartotojo sąsajai naudojamas stilių šablonas CSS, kuriame skoningai parinktas 3-4 spalvų rinkinys.

Projekto informaciniai laukai apima daug sričių, aprašančių projektus.

Internetinė vartotojo sąsaja

Kliento pusės PĮ naudoja tiek darbo vietos resursų, kiek reikalauja interneto naršyklė.

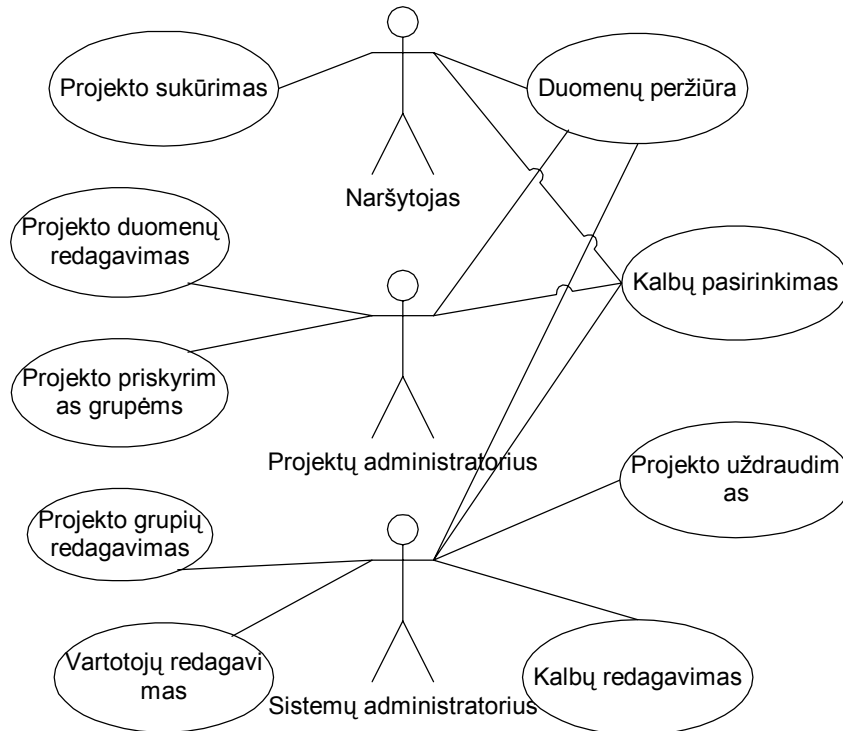
Serverio pusės PĮ naudoja tiek darbo vietos resursų, kiek reikalauja DBVS, bei servletų konteineris.

PS aptarnauja keletą vienu metu prisijungusių vartotojų.  
Kliento pusės PĮ paleisti reikalinga naršyklė  
Programiniams moduliams paleisti reikalingas servletų konteineris  
Servletų konteineriui reikia Java SDK programinio paketo  
PS priežiūrai nereikia ypatingų programavimo žinių  
Taikoma “multi-tier” architektūra  
PS duomenų redagavimas suteikiamas tik atlikus autorizuotą pasijungimą.  
Sistemoje įtraukiamos lietuvių, anglų ir vokiečių kalbos  
PS išėties tekstas platinamas nemokamai su galimais apribojimais numatytais licencijoje.

### **3.5 Panaudojimo atvejų vaizdas**

Pasinaudojant atlikta tikslinės auditorijos analize buvo išskirtos funkcijos rubrikų, sistemos kalbų, projektų, projektų dalyvių ir projektų stadijų kūrimui, atvaizdavimui, redagavimui ir trynimui. Programų sistemos funkcionalumas buvo padalintas trijų lygių vartotojams ir buvo gautas toks panaudojimo atvejų modelis (16 pav.).

Buvo išskirti trys aktoriai: sistemos administratorius, projektų administratorius ir naršytojas. O taip pat devyni panaudojimo atvejai: duomenų peržiūra, kalbų pasirinkimas, projekto sukūrimas, projekto duomenų redagavimas, projekto priskyrimas grupėms, projekto grupių redagavimas, projekto uždraudimas, kalbų redagavimas, vartotojų redagavimas.



16 pav. Projektų registravimo sistemos funkcijos

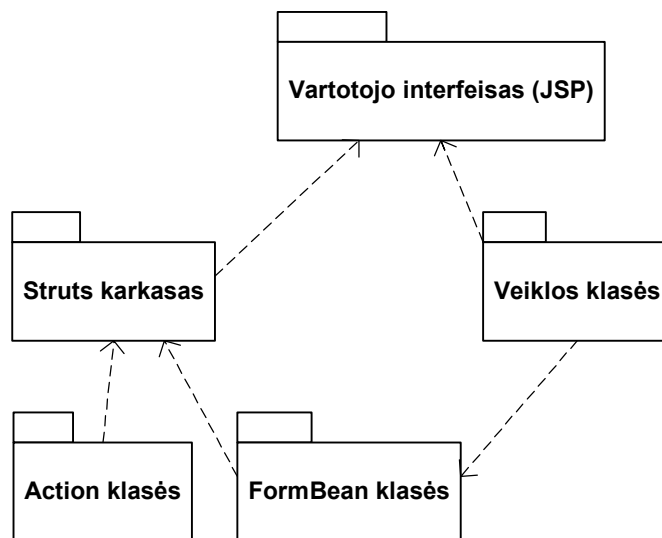
Kadangi kuriamos programų sistemos duomenų modelis yra specifinis, standartinių turinio valdymo sistemų pritaikymo atsisakyta, tačiau buvo toliau laikomasi turinio valdymo sistemos koncepcijos.

### 3.6 Loginis vaizdas

Kuriamos sistemos klases galima suskirstyti į keletą paketų. Tai:

- veiklos klasės, teikiančios duomenų apdorojimo paslaugas, bei ryšį su DB;
- Struts karkaso UserAction klasės, vykdomos atlikus tam tikrą veiksmą;
- Struts karkaso UserForm klasės, vykdomos įvedant, redaguojant duomenis į formą;
- Struts karkaso ActionError klasės, vykdomos aptikus kažkokią klaidą;
- JSP puslapiai – vartotojo sąsaja, atliekantis vaizdavimą ir apimantys minimalų kiekį kodo;
- tiesiogiai nenaudojamos ir nekuriamos Struts karkaso klasės.





17 pav. Paketų diagrama

Detali „Struts“ paketo diagrama pateikta analizės dalyje. Šioje diagramoje yra “Struts” klasės, kurios naudojamos programų sistemoje. Naujai sukuriamos UserAction, UserForm, bei ActionErrors klasės paveldi atitinkamas “Struts” kontrolierio klases, tačiau atlieka kūrėjų nustatytus veiksmus.

Pasinaudodami analizės dalyje aprašytais įrankiais, sukūrėme “Struts” paketą panaudojančių komponentų diagramas. Diagramose pavaizduoti Action komponentai realizuojantys veiksmus, FormBean veiklos klasės ir vartotojo sąsaja (JSP). Diagramos pateiktos prieduose.

Dalis veiklos realizacijos perkelta į JavaBeans ir laisvai pasirinktos architektūros veiklos klasių paketą. Jame realizuojami toki veiksmai, kaip pasijungimas prie duomenų bazės, duomenų perdavimas į/iš duomenų bazę ir kiti veiksmai. Šios klasės diagrama taip pat pateikiama prieduose.

### 3.7 Proceso vaizdas

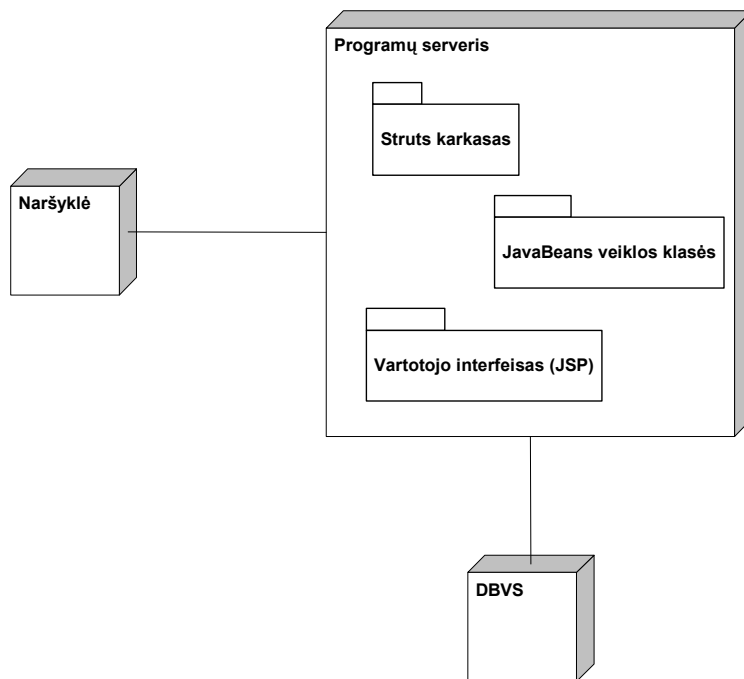
Šioje dalyje aprašyta pagrindinių programų sistemos duomenų – projekto informacijos kūrimo procesas. Sistemų administratorius, projektų administratorius bei naršytojas sukuria, redaguoja, bei peržiūri projekto informaciją. Tik sukurtą projekto informaciją galima peržiūrėti, redaguoti, priskirti atitinkamai projektų grupei.

Taip pat aprašomas konkrečių komponentų bendradarbiavimas, kuriant, bei redaguojant projektą. Iš vartotojo sąsajos (JSP) pateiktų formų iškvietus duomenų įvedimą, vykdomas Action

veiksmas, kuris panaudoja duomenis perduodančius komponentus FormBean. Toliau yra kviečiami duomenų bazės pasijungimą realizuojantys veiklos klasės komponentai ir duomenys išsaugojami saugykloje. Struts valdiklis iškviečia JSP puslapį, į kurį nukreipiama, sėkmingai įvykdžius veiksmus.

### 3.8 Paskirstymo vaizdas

Klasikinį trijų lygių sistemos pasiskirstymo modelį, pritaikome ir šiam inžineriniam sprendimui. Vartotojas duomenis įveda ir rezultatus pamato interneto naršyklėje. Ši bendradarbiauja su programine įranga per programų serverį „Tomcat“, kuriame įdiegtas „Struts“ karkasas, JavaBeans komponentai ir vartotojo sąsaja (JSP). Duomenys saugojami duomenų bazėje, kitoje programinėje stotyje (18 pav.).



18 pav. Paskirstymo diagrama

### 3.9 Realizacijos vaizdas

Visumoje sistemą sudaro keletas paketų. Detaliau apie juos kalbama loginio vaizdo aprašyme. Pilnas sistemos vaizdas atsispindi žemiau pavaizduotoje diagramoje. Ten matome į vartotojo sąsają, FormBean, bei Action komponentus suskirstytą programinę įrangą. Atskirame pakete pateikiamas JavaBeans ir kitos veiklos klasės. Pilną programinę įrangą sudaro dar

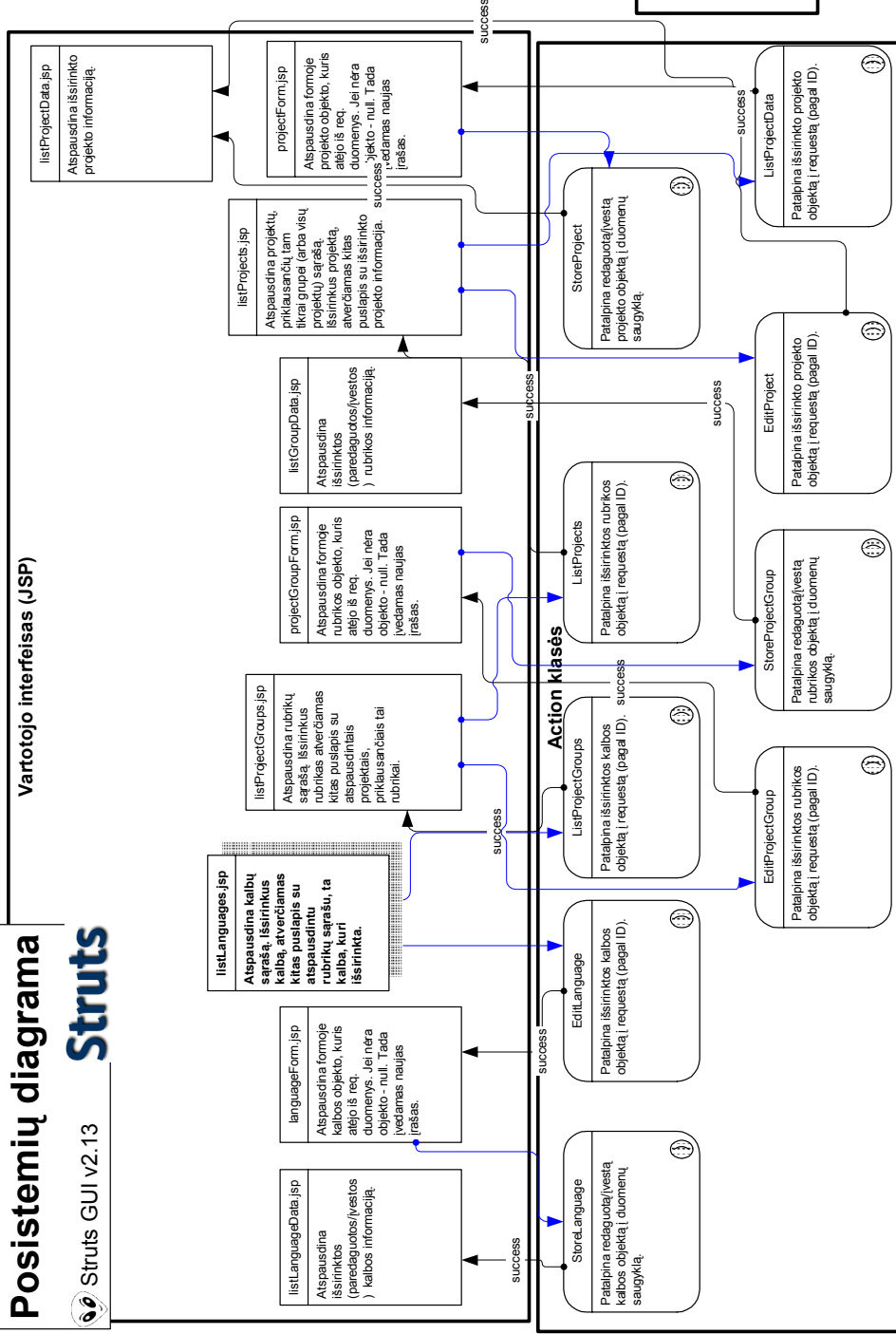
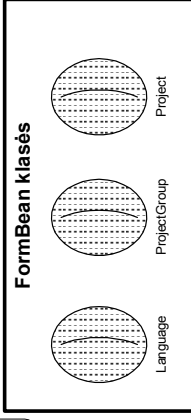
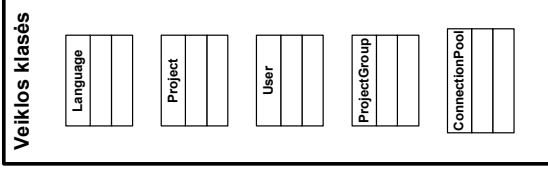
daugiau komponentų, tačiau nurodytoje diagramoje pateikiami pagrindiniai atliekantys kalbų, projektų, bei rubrikų redagavimo veiksmus.

# Posistemiu diagrama

## Struts

Struts GUI v2.13

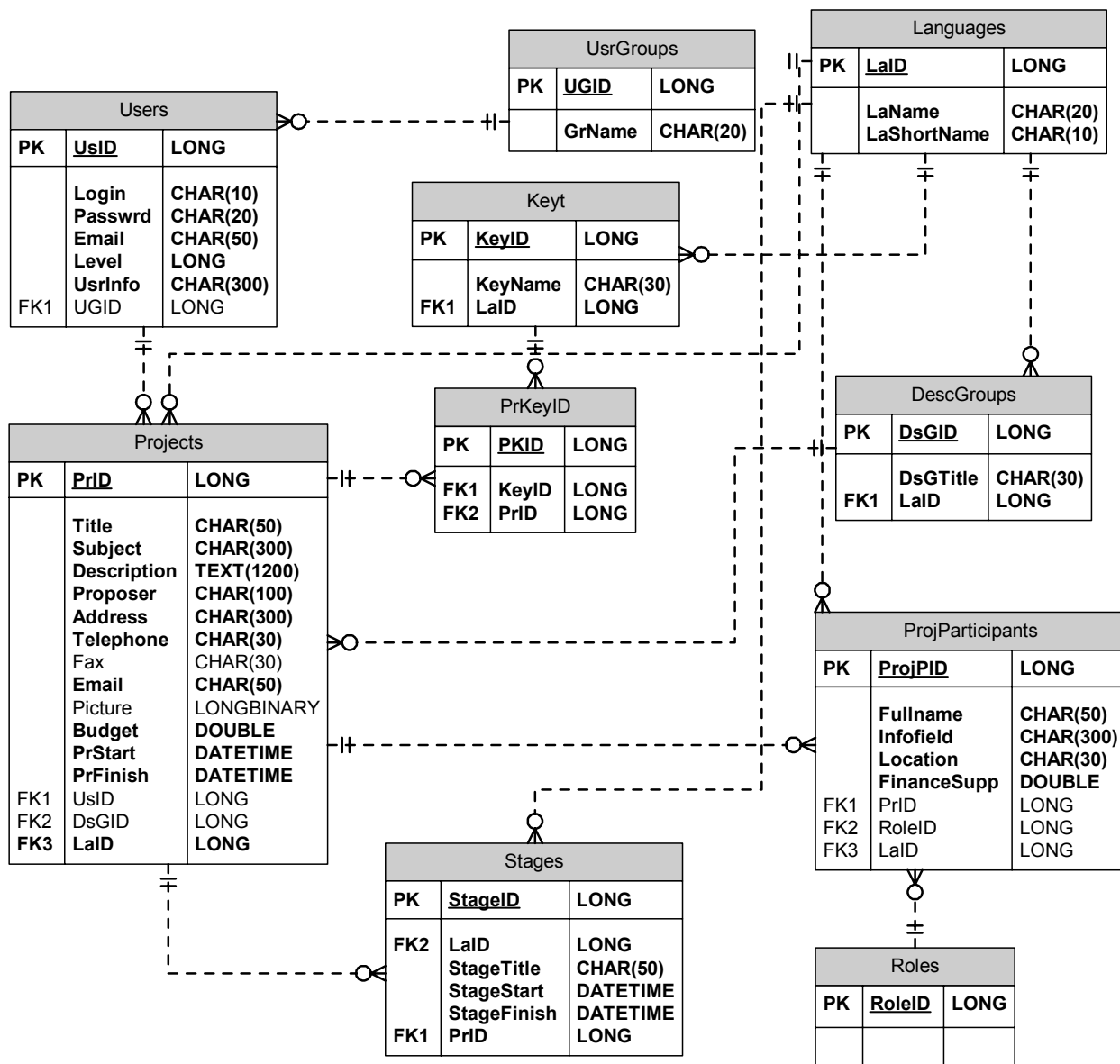
### Vartotojo interfeisas (JSP)



19 pav. Realizacijos vaizdas

### 3.10 Duomenų vaizdas

Duomenų vaizdas pateikiamas sekančiame paveikslėlyje. *Projects* lentelėje saugoma projektų informacija, kuri dubliuojama įvestoms sistemos kalboms. Kiekvienas projektas turi vieną arba kelis vykdytojus, kurių informacija saugoma atskiroje lentelėje *ProjParticipants*. Projekto vykdytojams priskiriama tam tikra rolė *Roles*. Ši lentelė numatyta galimiems programų sistemos praplėtimams, įtraukiant ir rolių redagavimą. Sukurtas projektas priskiriamas tam tikrai grupei *DescGroups*, pagal projekto tematiką. Grupė taip pat gali apimti vieną arba daugiau projektų. Informacija gali būti saugojama keliomis kalbomis, taigi visos tekstinę informaciją turinčios lentelės papildomai identifikuojamos kalbos lauku, kuris rodo į lentelę *Languages*. Kadangi skirtingos programų sistemos funkcijos pasiekiamos skirtingų teisių vartotojams, jie kurie registruojami atskiroje lentelėje *Users*.



20 pav. Duomenų vaizdas

## 4. Mokslo tyrimo darbų portalų tyrimas

### 4.1 Panašūs portalai

Kad programų sistema būtų priimtinesnė vartotojams, buvo atliktas rinkos tyrimas, palyginant panašius portalus. Buvo kreipiamas dėmesys į vartotojų iškeltus tiklus, paslaugų įvairovę, bei jų būtinumą vartotojams.

Tyrimo metu buvo orientuojamasi į tai, kiek kita paslauga pranašesnė ar prasčiau įgyvendinta už sukurtą produktą.

Internetu jau galima pasiekti įvairius projektus aprašančias sistemas, kur tenkinant vienas ar kitas sąlygas galima įtraukti ir nacionaliniu lygiu vykdomą projektą.

IST programos skirto projekto “Ideal-ist” tinklalapyje, adresu [8] galime rasti Research Priorities 2002 – 2006 for FP6 list. Patogu ir paprasta išsirinkti dominančius projektus pagal tipą (*Integrated Projects; Networks of Excellence*).

Projektų peržiūra prieinama per “*FP6 Partner Search*” skiltį, kur suteikiama galimybė ieškoti informacijos pagal projekto ID, Proposal name, Subject, Description laukus. Tačiau pasigendama aiškesnės klasifikacijos paprastai projektų peržiūrai. Norėtusi pasiekti visus projektus priklausančius pasirinktai tyrimų temai, kuri iš dalies prieinama tik per partnerių paieškos langą [8]. Sugalvojus įdiegti daugiakalbystę – darbo kiekis padvigubėtų ar patrigubėtų priklausomai nuo kalbų skaičiaus.

“Sixth Framework Programme” [7] svetainė skirta tai pačiai FP6 programai, tačiau programoje dalyvaujančių projektų sąrašas kur kas didesnis ir galimybės patogesnės. Per interneto naršyklę prieinama programinė įranga suteikia galimybę patiems projektų vadovams redaguoti savo projektų informaciją. Trūksta tik aiškia hierarchija išdėstytų projektų, tačiau tai atperka trijų rūšių paieška. Abi paieškos skirtos rasti partneriams skiriasi tik šalių sąrašu: viena apima kandidačių į ES šalių projektus, kitoje išvardintos visos įtrauktos valstybės.

“*Find a partner*” Bendrųjų interesų paieškoje (*Expressions of Interest*) pateikiamas platus mokslo sričių sąrašas – jame galima pasižymėti kelias dominančias sritis. Atskirai galima įtraukti pageidaujama paieškos frazę, pasirinkti dominančius projektus pagal tipą, bei šalį.

Svetainėje nėra įdiegta daugiakalbystės, nėra atskiros portalo dalies, kur išvardinami visi vykdomi projektai.

Portalas puikiai tinka įtraukti nacionaliniams projektams pretenduojantiems į FP6 paramą, tačiau dėl plataus kitų šalių projektų sąrašo apribojamas bendravimas ir tikslų tenkinimas nacionaliniu lygiu.

Prototipinė IST programos portalo versija sukurta [12] adresu. Tai galingas įrankis, kuriam kurti pasirinktos Java priemonės. Vartotojui suteikiama daug galimybių personalizuojant portalą – galima susikurti savo meniu, pakeisti meniu pavadinimus (taigi ir išsiversti į savo kalbą), tačiau tai gali sukelti sunkumų mažiau patyrusiam vartotojui. Pateikiama galimybė rinktis projektus pagal kategorijas, atlikti paiešką, tačiau kategorijos skirtingomis kalbomis pateikiamos viename sąraše, kas nėra taip patogiu.

Nepaisant patogios ir šiuolaikiškos darbo aplinkos, vėlgi pastebime kalbų pasirinkimo trūkumą – vartotojo aplinka angliška (ją galima dalinai “išsiversti”), nors dalykinė informacija pateikiama keliomis kalbomis. Kalbant apie projektų informaciją jos taip pat pateikiama mažokai – vos kelios eilutės apie projektą ir prisegtas dokumentas.

Galvojant apie nacionalinių projektų įtraukimą šiame portale išlieka lokalizavimo problema – didelis pašalinės informacijos kiekis trukdys bendrauti nacionaliniu lygiu.

Šaltinyje [1] pateikiami išsamūs susijusių su IST informacijos skelbimu tinklalapių užsienio kalba analizės duomenys.

Labiausiai struktūrizuoto Lietuvos inovacijų centro portalo [10] peržiūros galimybės skurdokos. Vienintelė projektų informacijos pasiekimo galimybė per partnerių arba technologijų paieškos formas. Paieška neveikia neužpildžius visų laukų, t.y. negalima gauti viso darbų sąrašo tik pasirinktai temai.

Tarp atsitiktinai sugeneruotų paieškos rezultatų dominuoja kitų šalių įmonės, taigi lietuviškas interneto adresas negarantavo nacionalinių projektų netrukdomo bendradarbiavimo nesimaišant pašalinei informacijai.

Yra kalbinių problemų Lietuvos inovacijų centro portale: lietuviška ir angliška interneto svetainių versijoms suteikiami adresai skirtinguose domenuose, o informacija apie projektus pateikiama tik angliškai.

Nacionalinių projektų, siekiančių dalyvaujančių FP6 programoje galime sužinoti iš “Sixth Framework Programme” portalo [6] atliktos paieškos pagal šalį, pvz. įvedus “Lithuania”. partnerių paieška gražina 8 įrašus, bendrųjų interesų paieška (Expressions of Interest) gražina 80 įrašų. Taigi matome, jog Europos erdvėje veikiančių projektų yra pakankamai ir netgi 80



Lietuvos projektų yra užregistruoti mažiau žinomame portale. Todėl galima numanyti, jog mažesnių projektų ir veikiančių tik Lietuvoje, bei norinčių keistis informacija ir pritraukti partnerių ar investitorių, skaičių atsiras daugiau.

Matyti, kad panašių produktų pasaulyje yra, tačiau juose trūksta lokalizacijos ir pritaikymo nacionalinių mokslo tyrimo darbų plėtrai. Tačiau jie pateikia nemažai patogių funkcijų, kai kur net daugiau, negu sukurtas produktas. Tuo tarpu nacionaliniai portalai yra pritaikyti Lietuvos mokslo tyrimo darbų reklamavimui, tačiau juose trūksta funkcionalumo.

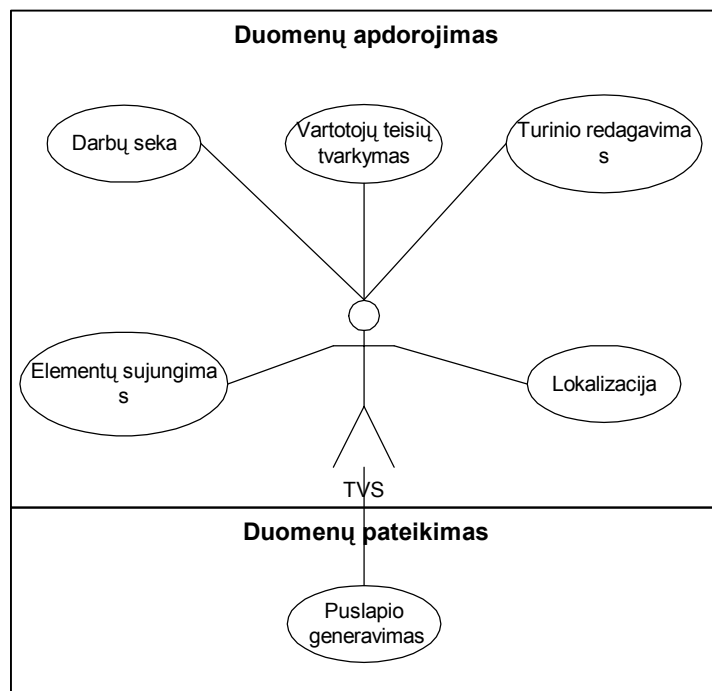
Šio tyrimo duomenys apibendrinami šioje lentelėje:

Lentelė 2 projektų registravimo portalų palyginimas

Portalo savybė	Lietuviški portalai	ES portalai	Skurta projektų registravimo PS
Lokalizacija	Gerai	Silpnai	Gerai
Turinio redagavimas	Silpnai	Gerai	Gerai
Klasifikacija	Vidutiniškai	Gerai	Gerai
Kitos TVS funkcijos	Silpnai	Vidutiniškai	Vidutiniškai

## 4.2 TVS funkcionalumas

Funkcionalumą galime plačiau apžvelgti lygindami sukurtą produktą su TVS funkcijomis, apibrėžtomis analizės dalyje. Palyginant su standartinėmis TVS funkcijomis (pav. 2, 3) gaunamas šios sistemos turinio valdymo funkcijų PAM modelis:



21 pav. Projektų registravimo sistemos funkcijos TVS kontekste

Kurdami specifinę turinio valdymo sistemą, projektų registravimui, įgyvendinome svarbiausiomis, laikytas TVS funkcijas [2]:

Iš apdorojimo fazės:

- vartotojų teisės: buvo sukurti trijų lygių vartotojai. Naršytojui suteikta teisė skaityti informaciją, projektų administratoriaus jurisdikcijoje paliktas kiekvieno projekto informacijos tvarkymas, sistemos administratoriui prieinami sistemos lygio nustatymai – rubrikų, kalbų redagavimas.
- duomenų tvarkymas: rubrikos, kalbos, projektų informacija, projektų dalyviai, projektų fazės yra kuriama, trinama, redaguojama ir peržiūrima informacija.
- darbų seka: projektų informaciją įveda projektų administratorius, ją priskiria rubrikai, o sistemos administratorius, pastebėjęs nekorektišką informaciją, uždraudžia viešą projekto rodymą. Projekto administratoriui, pakoregavus informaciją, sistemos administratorius gali nuimti uždraudimo apribojimą.
- šablonų redagavimas: programų sistemos išvaizda keičiama koreguojant vieną šabloninį CSS failą.
- lokalizacija: projektų informacija dubliuojama priklausomai nuo sistemos kalbų skaičiaus. Programoje pasirinkus kalbą informacija atvaizduojama pasirinkta kalba.

Duomenų pateikimui išpildytas puslapio generavimas: tinklalapis sugeneruojamas naudojant programinį kodą ir duomenų bazėje saugomus duomenis.

Lygindami išskirtus panaudojimo atvejus su TVS funkcijomis gauname atitikimų lentelę:

**Lentelė 3 Panaudojimo atvejų atitikmenys**

<b>Programų sistemos panaudojimo atvejai</b>	<b>TVS funkcionalumo atitikmuo</b>
duomenų peržiūra	Puslapio generavimas
kalbų pasirinkimas	Puslapio generavimas, lokalizacija
projekto sukūrimas	Turinio redagavimas, darbų seka, elementų sujungimas
projekto duomenų redagavimas	Turinio redagavimas, darbų seka, elementų sujungimas
projekto priskyrimas grupėms	Turinio redagavimas, darbų seka, elementų

	sujungimas
projekto grupių redagavimas	Turinio redagavimas, elementų sujungimas
projekto uždraudimas	Turinio redagavimas, darbų seka, elementų sujungimas
kalbų redagavimas	Turinio redagavimas, elementų sujungimas
naudotojų redagavimas	Vartotojų teisių tvarkymas

## 5. Sukurtos mokslo tyrimo darbų PS eksperimentinis vertinimas

Vertinant produktą buvo apklausiami vartotojai, taip įvertinant programų sistemos sukūrimo pilnumą. Taip pat pagal vartotojų įvestus duomenis sistemoje buvo atlikta vartojimo analizė.

Vartotojas tyrimo metu užpildė sekančią lentelę, kurioje pažymėjo programų sistemos atitikimą specifikacijai. Išpildyti reikalavimai pažymėti pliusu, o atskiriems panaudojimo atvejams nenumatyti veiksmai pažymėti juodu fonu.

Aktoriai: SA – sistemos administratorius, PA – projektų administratorius, NR – neregistruotas naršytojas

Lentelė 4 produkto įvertinimo rezultatai

Nr.	Funkcionalumas	Pastabos	Aktoriai		
			SA	PA	NR
1.1	Prisijungimas prie darbo aplinkos		+	+	+
1.2	Atsijungimas nuo darbo aplinkos		+	+	
2	Darbo aplinkos kalbos pasirinkimas		+	+	+
3.1	Projekto grupės sukūrimas		+		
3.2	Projekto grupės redagavimas		+		
4.1	Naujo vartotojo sukūrimas		+		
5.1	Projekto sukūrimas		+		
5.2	Projekto duomenų redagavimas			+	
5.3	Projekto prisikyrimas grupei		+		
5.4.1	Projekto dalyvių įvedimas			+	

5.4.2	Projekto dalyvių redagavimas			+	
6	Projektų informacijos peržiūra		+	+	+

Veiklos klasių paketas turi aukštą panaudojamumo įvertinimą tiems patiems duomenims, kurie naudojami kuriamoje programų sistemoje. Pasirinkus kitokią PS kūrimo architektūrą, veiklos klasių paketą galima panaudoti naujoje programinės įrangos versijoje.

Kiti klasių paketai siejasi ne tik su programine įranga, bet ir su pasirinkta architektūra taigi jie nėra tokio aukšto panaudojamumo.

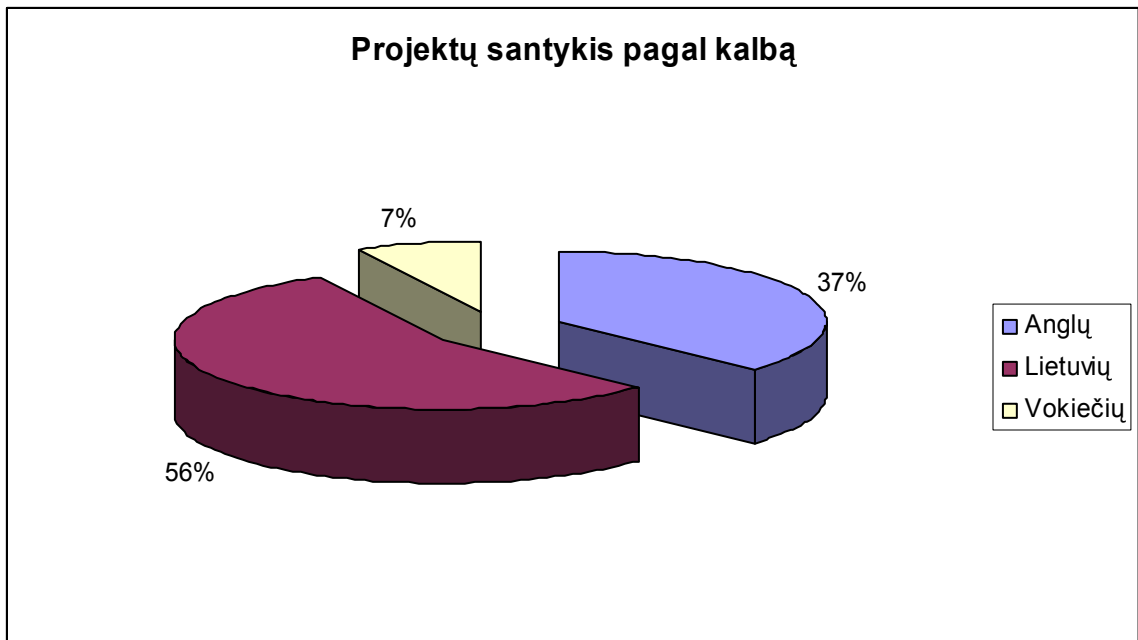
Tačiau, pasirinkta architektūra MVC – yra suderinama su kitais Jakarta projekto įrankiais internetinių programų kūrimui. Todėl įmanomas ir kitų paketų panaudojamumas pritaikant Jakarta įrankius, tokius kaip “Turbine”.

Silpniausia pasirinktos sistemos grandis yra JSP puslapiai vartotojo sąsajai. Šio paketo panaudojamumo, praplėtimo galimybės ypač menkos – didelė dalis kodo turi būti keičiama, kad pritaikyti pasirinktam poreikiui.

Kalbant apie praplėtimo galimybes – pasirinkta architektūra leidžia nesunkiai padidinti PS paslaugas, tačiau JSP vartotojo sąsaja reikalauja daugiausiai darbo atliekant pakeitimus.

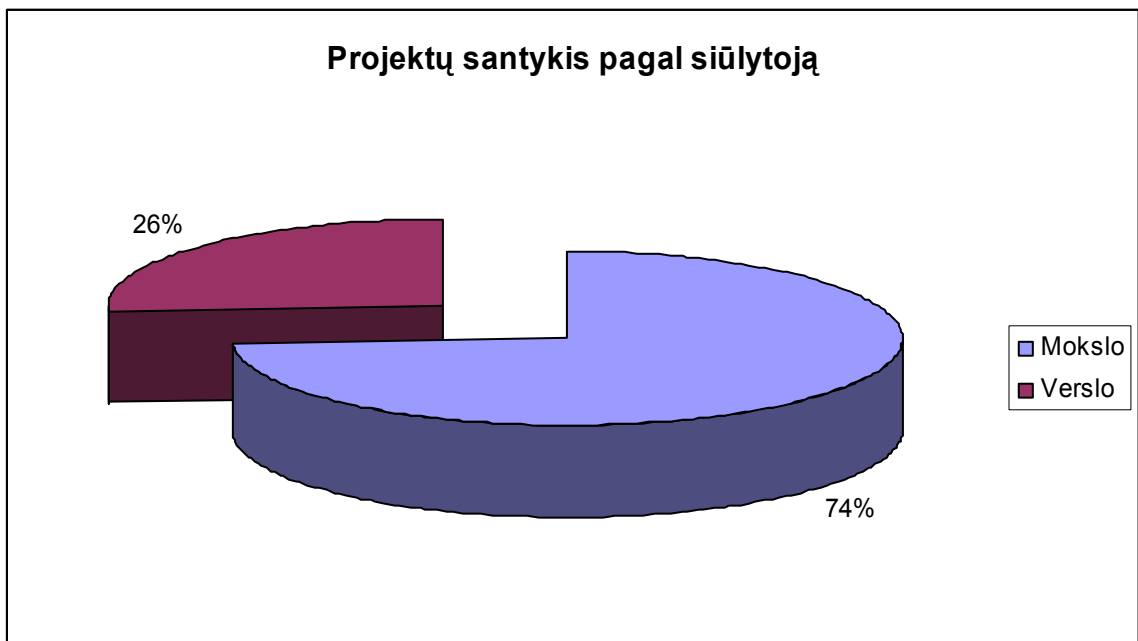
Specifikacijoje numatytos programų sistemos savybės buvo išpildytos ir išbandytos vartotojų. Naudodamiesi programų sistema vartotojai pateikė duomenis specializuotajai turinio valdymo sistemai. Pagal šiuos duomenis pabandėme išsiaiškinti programų sistemos stipriąsias ir silpnąsias puses.

Prototipinė sistema veikė vienų metų laikotarpyje, po kurio buvo atlikta įvestų duomenų analizė. Pradinė versija buvo paleista kaip trikalbis portalas, leidžiant įvesti duomenis anglų, lietuvių, vokiečių kalbomis. Metų laikotarpyje nebuvo nustatyta reikalavimų padidinti sistemos kalbų skaičių. Taip pat matyti, kad daugiausia informacijos buvo įvesta lietuvių ir anglų kalbomis.



*22 pav. Eksperimento rezultatai – santykis pagal kalbą*

Projektus siūlė mokslo arba verslo atstovai. Kaip matyti iš diagramos pagal projektų siūlymą, dominavo mokslo atstovai, kurių pasiūlyti projektai beveik tris kartus viršijo verslo atstovų pasiūlytų projektų kiekį.



*23 pav. Eksperimento rezultatai – santykis pagal siūlytoją*

Tyrimo metu buvo įsitikinta, kad programų sistema sėkmingai naudojasi vartotojai. Be visų kitų programų sistemos funkcijų pasitvirtino, kad daugiakalbiškumas sėkmingai funkcionuoja ir vartotojai mielai naudojami šia savybe.

Nevienodas vartotojų pasiskirstymas tarp verslo ir mokslo atstovų reiškia silpną mažiau suinteresuotų vartotojų pritraukimą. Labiau suinteresuotais vartotojais laikomi mokslo atstovai, kaip investicijų ieškantys inovatoriai. Šią atskirtį galima sumažinti padidinant turinio pateikimo paslaugų spektrą

Tobulinant pasirinktą sistemą verta pasidomėti duomenų perdavimu kitoms programų sistemoms XML formatu, personalizavimo ir paieškos galimybėmis.

## 6. Išvados

1. Atlikus turinio pateikimo internete analizę, prieita išvada kurti projektų registravimo sistemą naudojant turinio valdymo sistemų modelį: metodiką turinio valdymui ir atvaizdavimui. Darbe nagrinėjama problematika apibrėžta besivystančiame standarte JSR 170 – TVS laikmenų pasiekiamumo standartizavimas.
2. Pasirinkta Java kūrimo technologija dėl stipraus OO programavimo, nepriklausomumo nuo platformos, atvirojo kodo prieinamumo ir gausaus įrankių, bei komponentų pasirinkimo.
3. Naudojantis vienu iš Java komponentų MVC karkasu “Struts” išspręsta pateikties ir veiklos logikos atskyrimo problema, tarp objektų pasiekta mažesnio glaudumo (*less coupling*).
4. Remiantis analitinėje dalyje suformuluotais reikalavimais ir priimtais projekciniais sprendimais, naudojant interneto technologijas, atskiriančias pateikties logiką nuo veiklos logikos, realizuota mokslo tyrimo darbų registravimo ir pateikimo programų sistema.
5. Atliktas efektyvumo tyrimo eksperimentas rodo, jog programų sistema sėkmingai prigijo rinkos segmente skelbiant informaciją apie projektus Lietuvoje ir užsienyje ir yra pranašesnė už išnagrinėtus analogus: palaikomas daugiakalbiškumas, išvystytos turinio pateikimo funkcijos, pritaikoma tiek lokaliai, tiek tarptautiniam naudojimui.
6. Programų sistemos vartotojų segmentacija parodė, kad reikalinga tolimesnė plėtra jos vartojimui skatinti.
7. Atlikti tyrimai ir išvados užfiksuotos dviejų konferencijų leidiniuose: „Informacinės technologijos verslui-2004“ pranešimų leidinys (priimta spausdinti) ir 5-toji tarptautinė konferencija „Verslo galimybės, problemos ir sprendimai globalizacijai“ (“*Business Options, Problems and Solutions Concerning Globalisation*”).



## 7. Literatūra

1. **Butleris R., Taujanskas V.** The portal of research and development projects for business and science integration. 5th international conference "Business Options, Problems and Solutions Concerning Globalisation" 2004. pp. 430-437, 2004.
2. **Butleris R., Taujanskas V.** Turinio valdymo sistemos mokslo tyrimo darbų registracijai. Konferencijos „Informacinės technologijos verslui-2004“ pranešimų leidinys (priimta spausdinti).
3. **Cheong S.N., Azhar K.M., Hanmandlu M.** Personalization of Interactive News through J2EE, XML, and SMIL in a Web-Based Multimedia Content Management System, Lecture Notes in Computer Science, Vol.2532, pp.287-294, 2002.
4. **CMSWorks.** CMSWatch™. [interaktyvus]. [žiūrėta 2004.01.20]. Priega per internetą: <http://www.cmswatch.com>
5. **CORDIS: Community Research & Development Information Service.** IST and the European Research Area. [interaktyvus]. [žiūrėta 2004.01.20]. Priega per internetą: <http://www.cordis.lu/ist/about/era.htm>
6. **CORDIS: Community Research & Development Information Service.** Sixth Framework Programme. [interaktyvus]. [žiūrėta 2004.01.20]. Priega per internetą: <http://fp6.cordis.lu/fp6/home.cfm>
7. **European Commission.** The Sixth Framework Programme (2002-2006). [interaktyvus]. [žiūrėta 2004.01.20]. Priega per internetą: [http://europa.eu.int/comm/research/fp6/index\\_en.html](http://europa.eu.int/comm/research/fp6/index_en.html)
8. **Ideal-IST.** FP6 Partner Search. [interaktyvus]. [žiūrėta 2004.01.20]. Priega per internetą: <http://www.ideal-ist.net/partnersearch.php>
9. **Kochmer C.** An Introduction to Struts. [interaktyvus]. [žiūrėta 2004.01.20]. Priega per internetą: <http://www.jspinsider.com/content/jsp/struts/strutsintro.jsp>
10. **Lithuanian Innovation Centre.** [interaktyvus]. [žiūrėta 2004.01.20]. Priega per internetą: <http://www.innovation.lt>
11. **Lu J.** Reengineering of Database Applications to EJB Based Architecture. CAISE 2002, Lecture Notes in Computer Science, Vol. 2348, pp. 361-376, 2002.
12. Prototype system of the IST Research portal. [interaktyvus]. [žiūrėta 2003.01.20]. Priega per internetą: <http://193.74.145.219/EUPortal>.
13. **Rutkauskaitė R., Nemuraitė L.** Šablonų naudojimas kuriant duomenų apdorojimo sistemas internete. Iš: Informacinės technologijos 2004: konferencijos pranešimų medžiaga, Kaunas 2004 sausio 28 - 29 d.. Kaunas, 2004, p. 453-459. Informacinės technologijos 2004.
14. **Schuster E., Wilhelm S.** Web-Content-Managementsysteme. Informatik-Spektrum. vol. 23, Number 6, Publisher: Springer-Verlag Heidelberg, 2002.
15. ServerSideScripting techniques analysis. [interaktyvus]. [žiūrėta 2004.01.20]. Priega per internetą: <http://www.b2bsim.de/documents/wewior/main.html>
16. Struts 1.1 Controller UML diagrams. [interaktyvus]. [žiūrėta 2004.01.20]. Priega per internetą: <http://rollerjm.free.fr/pro/Struts11.html>
17. **SUN.** Comparing JavaServer Pages and Microsoft Active Server Pages Technologies. [interaktyvus]. [žiūrėta 2004.01.20]. Priega per internetą: <http://java.sun.com/products/jsp/jsp-asp.html>

## 8. Terminų ir santrumpų žodynas

Terminas	Apibrėžimas
Ant	Java kompiliavimo įrankis
DBVS	duomenų bazių valdymo sistema
Java	objektinė programavimo kalba
JavaBean	specialios architektūros Java klasė
JDBC	sujungimo su duomenų bazėmis Java programavimo kalba technologija
JSP	Java Server Pages – dinaminių puslapių integruojantys Java kodą
Karkasas	Framework. Taip apibūdinamas sprendimas pateikiantis architektūrinį sprendimą (design template)
Model 1, Model 2	JSP programų sistemų kūrimo modeliai. Model 1 atitinka JSP puslapių rinkinį, kai komponentai veikia dažniausiai nesusiję vienas su kitu. Model 2 atitinka MVC karkasą, kai servletas arba visas karkasas (servletų rinkinys) atitinka kontrolerio funkcijas, apdorojančias visas vartotojo užklausas.
Daugialypė architektūra	Multitier. Architektūra, kai skirtingi programų sistemos komponentai (kodas, vartotojo sąsaja, duomenys) talpinami skirtinguose serveriuose.
MVC	Model-View-Controller – interneto aplikacijose naudojamas architektūrinis sprendimas (design template) atskiriantis logiką (model), pateikimą (view) ir kontrolerį (controller) apdorojantį visas vartotojo užklausas.
Servletų konteineris	internetu serveris apdorojantis dinamiškus JSP puslapius bei Java komponentus, naudojamus internetinei aplikacijai.
Struts	MVC karkasas
Tomcat	Servletų konteineris
Serverio komponentai	Server-side. Komponentai talpinami tarnybinėje stotyje, sąveikaujantys su kliento kompiuteriu pagal daugialypės architektūros principus
Deploy	Patalpinti. Kalba eina apie Java serverio komponentų talpinimą tarnybinėje stotyje
Scriptlet	JSP tagai, naudojami veiklos logikos veiksams įterpti tarp HTML tagų JSP puslapyje.
Business logic	Veiklos arba skaičiavimų logika
Presentation logic	Vaizdavimo logika
Tags	Tagai
CSS/style-sheets	Stiliaus šablonai
HTTP request HTTP response	HTTP protokolo antraštės užklausiai ir atsakymui
I18N	Internacionalizacija – metodika, kaip dinamiškai išvesti pranešimus vartotojui pagal kompiuterio lokalę.

## 9. Priedai

### 9.1 Straipsnis Turinio valdymo sistemos mokslo tyrimo darbų registracijai.

Konferencijos „Informacinės technologijos verslui-2004“ pranešimų leidinys.  
Turinio valdymo sistemos mokslo tyrimo darbų registracijai  
Vytautas Taujanskas<sup>1</sup>, Rimantas Butleris<sup>2</sup>

KTU, Informacijos sistemų katedra  
Studentu 50-308,51368, Kaunas  
<sup>1</sup>vyttauj4@merkys.ktu.lt, <sup>2</sup>rimbut@if.ktu.lt

Straipsnyje pateikiami turinio valdymo sistemų skiriamieji bruožai, išvardinamos jų funkcijos, aptariamas aktualumas. Turinio valdymo sistemų modelis pritaikomas mokslo tyrimo darbų registravimo sistemai. Identifikuojamos reikalingos TVS funkcijos, jos įgyvendinamos, atliekama ateities darbų analizė.

## 1 Įvadas

Kuriant mokslo tyrimo darbų (arba projektų) registravimo sistemą išskirti informacijos vienetai reikalingi informacijai apie projektus pateikti: projektai, projektų temos, projektų kūrėjai, projektų stadijos. Šiems informacijos vienetais apdoroti reikalingos standartinės turinio valdymo sistemų funkcijos: CRUD (Create, Retrieve, Update, Delete) – sukūrimas, atvaizdavimas, redagavimas, ištrynimasis. Tačiau informacijos klasifikavimas kur kas sudėtingesnis nei turinio valdymo sistemų atveju. Todėl pasirinkta sukurti naują turinio valdymo sistemos modelį, pritaikytą projektų informacijai saugoti ir apdoroti.

## 2 Turinio valdymo sistemų savybės

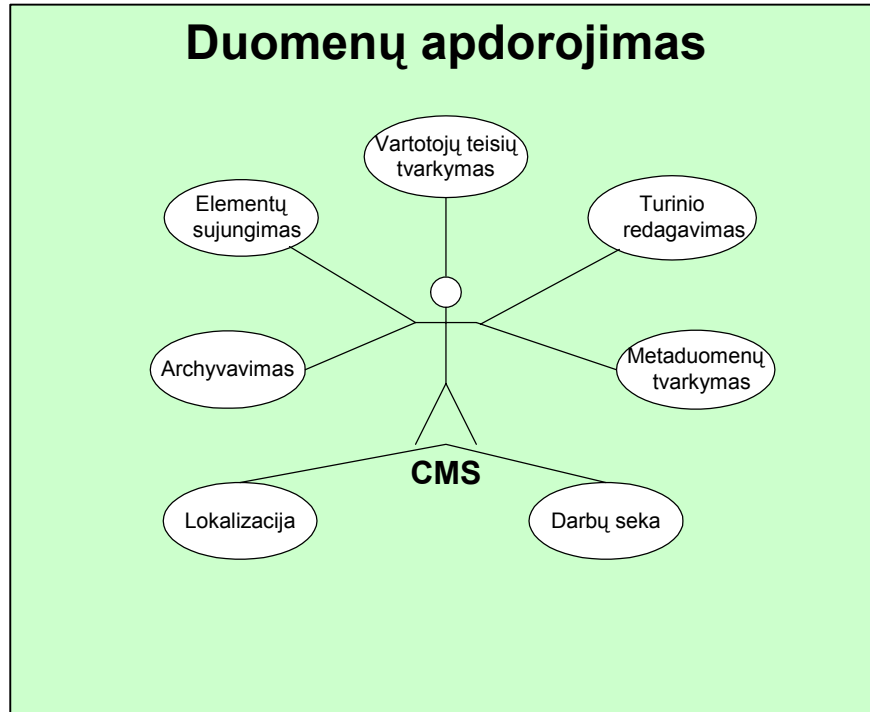
Turinio valdymo sistemų (TVS arba angliškai Content management systems CMS) paslaugas galima išskirti į dvi aiškias kategorijas: duomenų apdorojimą ir duomenų pateikimą [2].

Apdorojimo fazėje (pav. 1) yra šios funkcijos:

- vartotojų teisių tvarkymas: kam bus prieinamos įvairios sistemos funkcijos;
- duomenų tvarkymas: turinio talpinimas į sistemą;
- metaduomenų tvarkymas;
- darbų seka, apibrėžianti turinio talpinimą, redagavimą ir patvirtinimą;
- atskirų TVS paslaugų apjungimas ir talpinimas į TVS, šablonų redagavimas;
- archyvavimas: versijų kontrolė ir versijų valdymas;
- lokalizacija: programų sistemos pritaikymas atskiroms kalboms;

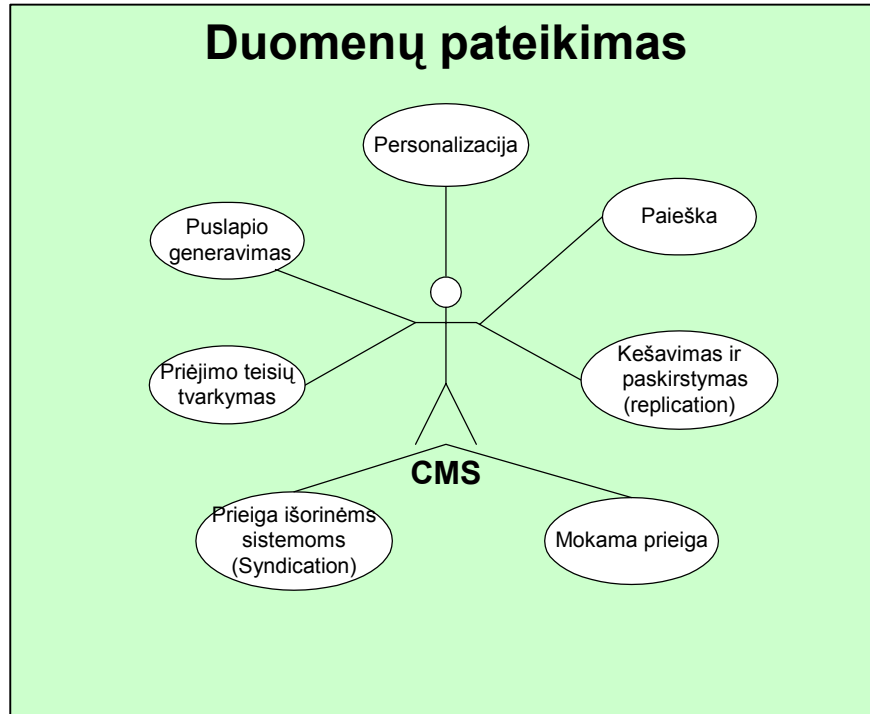
Duomenų pateikimas (pav. 2) apima šias TVS funkcijas:

- puslapio generavimą: pagrindinė TVS funkcija informacijos peržiūrai;
- paiešką;
- personalizaciją – kiekvienam vartotojui asmeninis TVS pritaikymas;
- priėjimo teisių tvarkymą – skirtingi vartotojai turi skirtingas informacijos peržiūros teises;
- prieigos apmokestinimas suteikia galimybes uždirbti turinio teikėjams;
- prieiga išorinėms sistemoms praplečia TVS matomumą;
- kešavimas ir paskirstymas pagreitina veikimą;



*1 pav. Turinio valdymo sistemos redagavimo dalies galimybės*

**Duomenų apdorojimo** paslaugos skirtos vidiniams sistemos vartotojui, kurio jurisdikcijoje yra duomenų įvedimo bei apdorojimo darbai. Saugaus prisijungimo užtikrinimas – vartotojo autorizacija ir teisių tikrinimas realizuojama per vartotojo grupes: kiekvienas vartotojas priklauso tam tikrai grupei, kuri apibrėžia skirtingas naudojimosi sistema teises. Autoriai pateikia medžiagą, redaktoriai ją patvirtina, o administratoriai gali redaguoti sistemos kodą, taigi keisti išvaizdą ir galimybes. Informacijos pateikimas TVS sistemai apibūdina darbų seką: pateikimas, po pateikimo seka redagavimas, po redagavimo – patvirtinimas. Įdiegus redaguotos informacijos įvairių versijų saugojimą, pristatoma archyvavimo bei versijų valdymo galimybė. Versijas skirtingoms kalboms vadiname lokalizacija – pritaikymu atskiros lokalizacijos vartotojams.



2 pav. Turinio valdymo sistemos pateikimo dalies galimybės

**Duomenų pateikimas** remiasi puslapio generavimu. Informacijos pasiekimui galima numatyti skirtingas priėjimo teises – įvedamas teisių tvarkymas ir išoriniams vartotojams, kurie prieina tik prie jiems skirtos informacijos. Tampa įmanomas mokamų paslaugų teikimas: priėjimas prie informacijos suteikiamas tik susimokėjusiems už peržiūrą naršytojams.

Personalizacijos paslauga didina svetainės patrauklumą – atvaizdavimas pritaikomas priklausomai nuo vartotojo. Fiksuojami bent trys personalizacijos tipai:

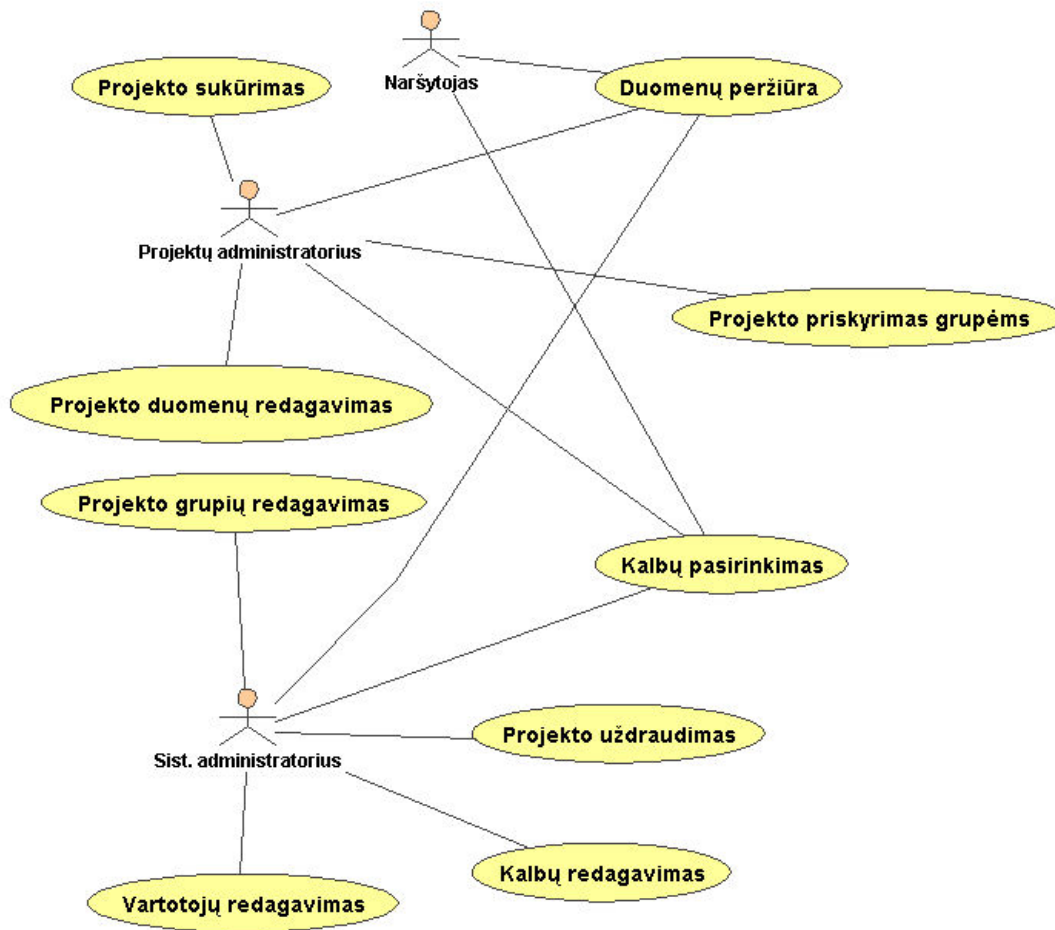
- grupinė,
- asmeninė,
- savybėmis paremta.

Pirmuoju atveju pateikiamos skirtingos personalizacijos galimybės kelių tipų vartotojų grupėms. Asmeninės personalizacijos atveju kiekvienas naršytojas gali sau asmeniškai nusistatyti svetainės nuostatas. Įdiegus savybėmis paremtą personalizaciją, internetiniai agentai stebi vartotojo elgseną ir pagal tai pritaiko nuostatas.

Kitos funkcijos pagerina TVS patogumą, greitaveiką, bei pasiekiamumą. Daug duomenų apimančiuose tinklapiuose, informacijos pasiekimo patogumui praverčia paieška. Padidinti sistemos veikimo greitį naudojamas kešavimas. Išskaidyti sisteminę apkrovą naudojama keletas tarnybinių stočių, todėl galimas duomenų surinkimas iš skirtingų šaltinių. Informaciją taip pat galima perduoti kitiems portalams – pvz. naujienų santrauką XML formatu.

### 3 Mokslo tyrimo darbų registravimo sistema – specializuota TVS

Žinodami aukščiau išvardintas TVS funkcijas, dalį jų pritaikome projektų registravimo sistemai. Žemiau pavaizduotas sukurtos programų sistemos paslaugų modelis.



3 pav. Projektų registravimo sistemos funkcijos

Kurdami specifinę turinio valdymo sistemą, projektų registravimui, įgyvendinome svarbiausiomis, laikytas TVS funkcijas:

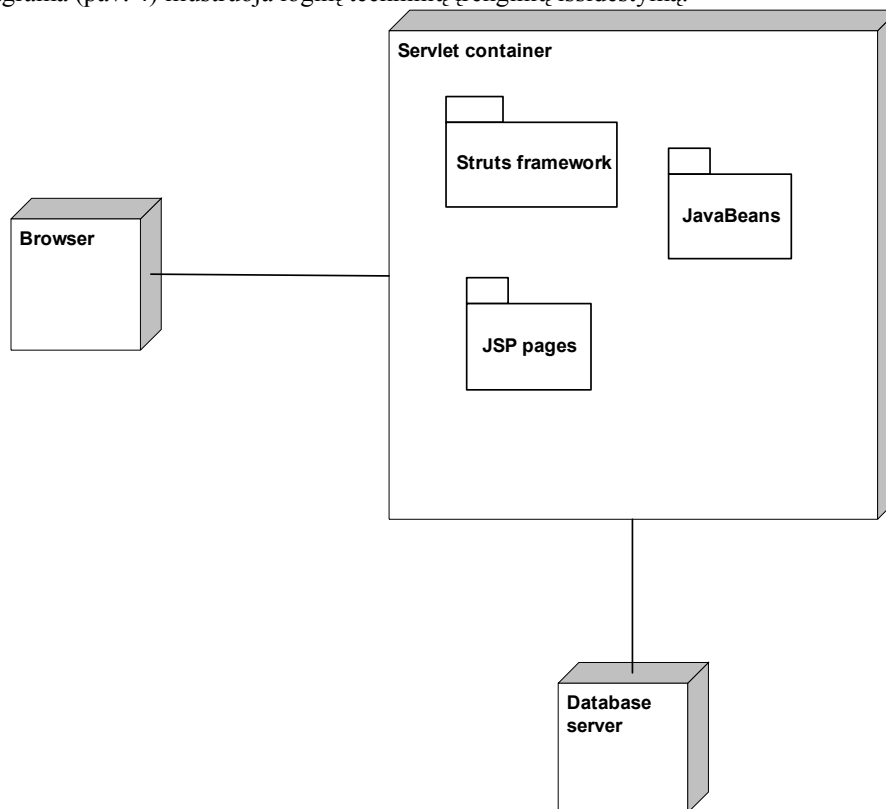
Iš apdoravimo fazės:

- vartotojų teisės: buvo sukurti trijų lygių vartotojai. Naršytojui suteikta teisė skaityti informaciją, projektų administratoriaus jurisdikcijoje paliktas kiekvieno projekto informacijos tvarkymas, sistemos administratoriui prieinami sistemos lygio nustatymai – rubrikų, kalbų redagavimas.
- duomenų tvarkymas: rubrikos, kalbos, projektų informacija, projektų dalyviai, projektų fazės yra kuriama, trinama, redaguojama ir peržiūrima informacija.
- darbų seka: projektų informaciją įveda projektų administratorius, ją priskiria rubrikai, o sistemos administratorius, pastebėjęs nekorektišką informaciją, uždraudžia viešą projekto rodymą. Projekto administratoriui, pakoregavus informaciją, sistemos administratorius gali nuimti uždraudimo apribojimą.
- šablonų redagavimas: programų sistemos išvaizda keičiama koreguojant vieną šabloninį CSS failą.
- lokalizacija: projektų informacija dubliuojama priklausomai nuo sistemos kalbų skaičiaus. Programoje pasirinkus kalbą informacija atvaizduojama pasirinkta kalba.

Duomenų pateikimui išpildytas puslapio generavimas: tinklalapis sugeneruojamas naudojant programinį kodą ir duomenų bazėje saugomus duomenis.

## 4 Sistemos kontekstas

Paskirstymo diagrama (pav. 4) iliustruoja loginių techninių įrenginių išsidėstymą.



4 pav. Paskirstymo diagrama [1]

Vartotojo prieiga suteikiama per naršyklę (Browser). Programiniai moduliai (JavaBeans) bei vartotojo sąsaja (JSP dinaminiai puslapiai) apdorojami servletų konteineryje (Servlet container). Duomenys saugojami DB tarnybinėje stotyje (DatabaseServer).

Programų sistemai buvo panaudotas klasikinis interneto sistemoms taikomas Model-View-Controller (MVC) karkasas.

Tai senas modelis, kuris atlaikė laiko testą ir atskyrė pateikties logiką nuo veiklos logikos. [4]

Daugelio interneto sistemų paskirtis yra paimti duomenis iš duomenų bazės ir parodyti juos vartotojui, o kai vartotojas juos modifikuoja – išsaugoti. Kadangi tarp duomenų bazės ir vartotojo sąsajos nuolat perduodamas duomenų srautas, šias dvi dalis tikslinga sujungti, kad sumažinti kodavimo apimtį ir pagerinti užduočių vykdymą. Vis dėlto šis atrodytų natūralus suartinimas turi keletą ryškių problemų.

Viena iš problemų yra ta, kad vartotojo sąsaja turi tendenciją keistis daug dažniau, nei duomenų saugojimo sistema.

Kita problema, susijusi su duomenų ir vartotojo sąsajos susijungimu, yra ta, kad veiklos logika reikalinga ne tik duomenų perdavimui. [4].

Buvo pasirinktas vienas iš pirmųjų MVC karkasų pritaikytų JSP “Struts”. Jis pasižymi tuo, kad:

- pateikia tagų bibliotekas, atliekančias logines, iteracijų operacijas, bei darbą su duomenimis, kas išsprendžia JSP įskaitomumą. **Bean** ir **logic** tagų bibliotekos naudojamos šioms problemoms išspręsti
- naudoja MVC architektūrą, kuri atskiria skaičiavimo ir pateikimo logiką.

Sukurtas produktas gali būti integruotas į susijusios tematikos portalą. Galimas ir atvirkštinis variantas – projektų pateikimo sistema gali būti papildyta papildomais interneto puslapiais, kurie talpinami į servletų konteinerį.

## 4 Išvados

Mokslo tyrimo darbų registravimo ir pateikimo programinei sistemai realizuoti buvo taikomas turinio valdymo sistemų modelis. Buvo sukurta specifinė turinio valdymo sistema, pritaikyta kuriamos programų sistemos informaciniams vienetams: rubrikoms, kalboms, projektams, projektų dalyviams, projektų etapams apdoroti.

Sistemai sukurti buvo naudojamas MVC karkasas "Struts" kuris išsprendė kūrėjams kylančias pateikties logikos atskyrimo nuo veiklos logikos problemas. Kartu naudojamos komponentų bibliotekos padėjo optimaliai kurti programų sistemą.

Lyginant sukurta programų sistemą su klasikine turinio valdymo sistema pastebėta, kad informacijos pateikimo funkcijų pirmoje yra kur kas mažiau. Tobulinant pasirinktą sistemą verta pasidomėti duomenų perdavimu kitoms programų sistemoms XML formatu, personalizavimo ir paieškos galimybėmis.

Vykdamas galimus ateities išplėtimus, pasirinktas architektūrinis MVC modelis leis nesunkiai atnaujinti programų sistemą.

## Literatūra

1. R.Butleris, V.Taujanskas. The portal of research and development projects for business and science integration. 5th international conference "Business Options, Problems and Solutions Concerning Globalisation" 2004.
2. CMSWorks. CMSWatch™. <<http://www.cmswatch.com>>
3. S.N.Cheong, K.M.Azhar, M.Hanmandlu. Personalization of Interactive News through J2EE, XML, and SMIL in a Web-Based Multimedia Content Management System. Y.-C. Chen, L.-W.Chang, and C.-T. Hsu (Eds.): PCM 2002, LNCS 2532, pp. 287-294, 2002. Springer-Verlag Berlin Heidelberg 2002
4. R.Rutkauskaitė, L.Nemuraitė. Šablonų naudojimas kuriant duomenų apdorojimo sistemas internete. Informacinės technologijos 2004. P. 453-459.
5. E.Schuster, S.Wilhelm. „Web-Content-Managementsysteme“. Informatik-Spektrum. Volume 23, Number 6. 18 April 2002. Publisher: Springer-Verlag Heidelberg
6. B. Surjanto, N. Ritter, H. Loeser. XML Content Management Based on Object-Relational Database Technology. WISE 2000, Proceedings of the First International Conference on Web Information Systems Engineering, vol. I (Main Program), Hong Kong 2000. Basel. <<http://www.astrum.ch/uni/wi/diplom/diplom-leisi-em.pdf>>



## 9.2 Straipsnis. The portal of research and development projects for business and science integration.

5th international conference "Business Options, Problems and Solutions Concerning Globalisation" 2004.  
The portal of research and development projects for business and science integration

Vytautas Taujanskas

Information Systems Department, Kaunas University of Technology, Lithuania  
Phone:+370-7-45 34 45, e-mail: vytatauj@soften.ktu.lt

Rimantas Butleris

Information Systems Department, Kaunas University of Technology,  
Department of Informatics, Vilnius University Kaunas Faculty of Humanities,  
Phone:+370-7-45 34 45, e-mail: rimbut@if.ktu.lt

Analysis of already created portals, which appeal to national research and development works presentation, are made. Lacks and advantages of existing portals are overlooked and considered for prototype development. National research and development portal functions are described and solution is presented. Facilities of research and development portal prototype are examined and compared to possible functions of such system.

### 1 Introduction

Improving the access to reliable and comparable information on national research efforts has been identified as a first challenge to overcome in order to move into the direction of an European Research Area (ERA).

European research activities are structured around consecutive four-year programs, or so-called Framework Programmes. The Sixth Framework Programme (FP6) sets out the priorities - including the Information Society Technologies (IST) priority - for the period 2002-2006 [2].

Under the third block of FP6 - Strengthening the Foundations of the ERA - specific actions are envisaged under the title "coordination of research activities". This is intended to facilitate the networking (and opening) of national or regional RTD programs - the so-called ERA-NET.

An IST ERA Working Group of Member State representatives has been established to facilitate progress on preparing IST ERA coordination activities [1].

The goal is to create IST RTD (Information society technologies Research and Technological Development) portal, which will serve for scientific research and business communication and research projects advertisement. The main communication actions supported by this portal are:

- Research and development to business (RD2B);
- Business to research and development (B2RD);
- Research and development to research and development (RD2RD).

Facilities of research and development portal prototype are examined and compared to possible functions of such system. Analysis of already created portals, which appeal to national research and development works presentation, are made. Lacks and advantages of existing portals are overlooked and considered for prototype development. National research and development portal functions are described and solution is presented.

## **2 Scope and objectives**

### **2.1 Purposive market**

Users groups of IST RTD Portal were determined, following feasibility evaluations in Final report. User requirement survey and National Web site survey documents supplied by IST ERA Working Group [1]:

- politics;
- science and business representatives;
- mixed group of users.

The first group of users include politics, responsible for national research and development works registration and management, government representatives related with educational institutions or functionary responsible for ISTRTD or other FP6 programs.

Responsible for institute administration, university community and individual scientists are pointed among science and business representatives. Some of them could have relationship with market.

Mixed group representatives are Europe IST programs candidates, companies' executives for development or marketing and investors.

### **2.2 Objectives**

ISTRTD portal is for information publication about future or already executing RTD projects, participant attraction and information exchange (see Figure 1). Every of possible user groups have its own interests of benefit.

Politics will have a tool for RTD project classification, possibility to search for outstanding projects information, implement goals for national RTD projects expansion.

Science representatives acquire information about related field RTD projects, communication facilities projects participants and possibility to raise investments. Universities could use portal for education programs renewal according to modern projects. Young scientists will have possibility to choose interesting research field and possibility to participate in project development.

Businessmen will find knowledge for entrepreneur possibilities – point business to innovative field of science, acquire profitable contacts [9].

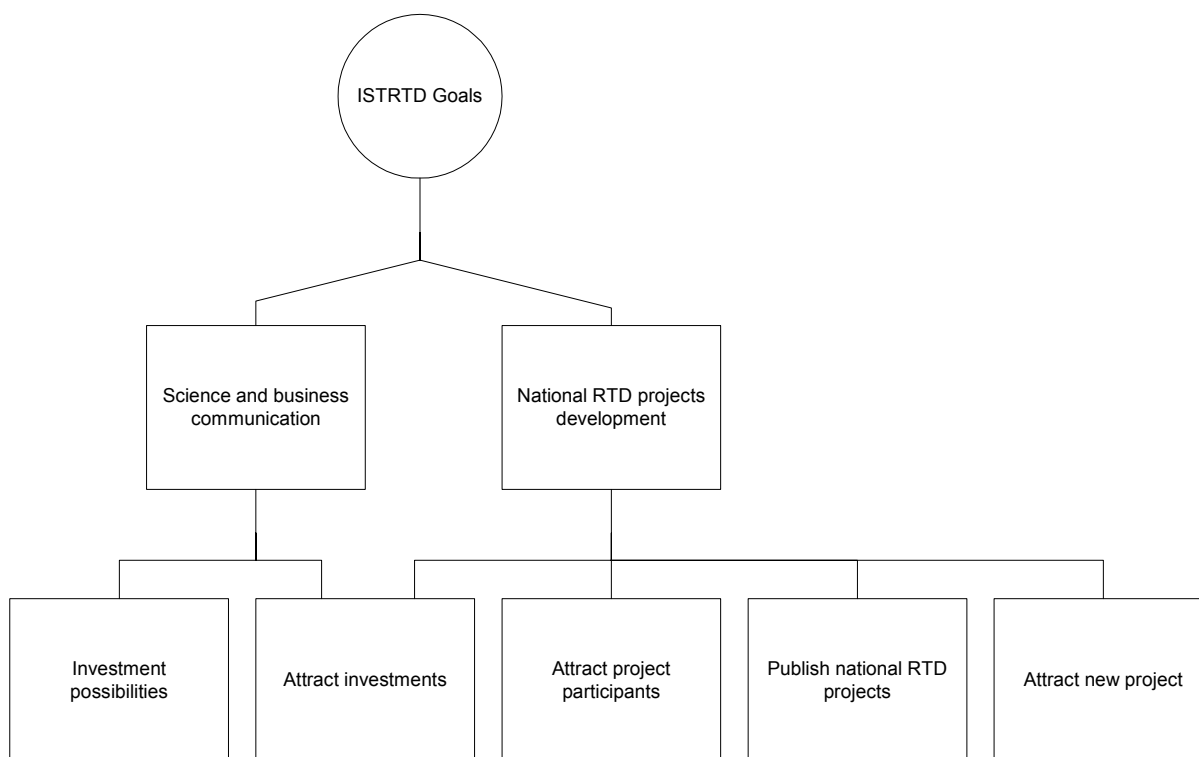


Figure 1. IST RTD portal goals model

### 3 Research of existing portals

#### 3.1 European scope portals

Some of the Internet portals could be used for RTD projects information. More or less they match RTD portal goals (figure 1).

“Ideal-IST” portal publishes Research Priorities 2002 – 2006 for FP6 list [4]. Classification of projects is produced as a matrix, so this function could be more simply and offer a larger number of categories. There is a lack to select any side of matrix and to retrieve associated projects, but qualitative “FP6 Partner Search” compensates classification drawback.

“Partner Search” offers information retrieval by project ID, Proposal name, Subject, Description fields. Portal informs about projects using only one language, what is a remarkable disadvantage.

“Sixth Framework Programme” [2] offers couple kind of search for RTD information: “Find a Call<sup>1</sup>” and “General Search”, which has other Search forms for the FP6 Expressions of interest and for FP6 partners from the Candidate States.

Various information fields in every search results offers powerful information spread capabilities, but there is also a lack to browse projects by selecting projects groups. Of course this is compensated by powerful search capabilities.

---

<sup>1</sup> A legal text calling interested parties to submit proposals for projects. The text defines the necessary specifications to prepare and submit a proposal, i.e. thematic priorities, instruments used, address and other technical means for submission, deadlines etc.. [6]

Classification (FP6 Activity codes) although as a separate field is provided in every search. This portal also lacks multilingual information about projects, which is partially solved in call information adding attached documents, which can be provided in multiple languages.

Prototype of IST RTD portal was available in 2002 [8]. This was a powerful engine for project information publishing, which included personalization, classification browsing, search, and some multilingual capabilities.

This also was produced with some lacks, for example, user interface language was only English, although project information was produced in couple languages. Numbers of project information fields also were too little, but prototype provided additional information by attaching document.

Categories list in different languages was provided in the same list, which made harder using classification.

### 3.2 Lithuanian R&D portals

[1] site provided research of national RTD portals.

Table 1. Information structure of Lithuanian RTD portals

Surveyed URL	<a href="http://www.innovation.lt">http://www.innovation.lt</a>	<a href="http://www.ktu.lt/en">http://www.ktu.lt/en</a>	<a href="http://www.mokslas.cr.vu.lt/english">http://www.mokslas.cr.vu.lt/english</a>	<a href="http://www.mokslas.lt/english">http://www.mokslas.lt/english</a>
<b>Identification data</b>				
Name of organization	Lithuanian Innovation Center	Kaunas University of Technology	Vilnius University	Department of Science and Higher Education of the Ministry of Education and science
Organization type				
Federal government				X
Regional government				
University		X	X	
National research networks				
Research institutions	X			
Funding organizations				
Special programmes				
Industries				
Other				
<b>Website's last update</b>	1 to 6 months	Info not available	>12 months	Info not available
<b>Part I: Content measurement</b>				
<b>Information Society Technology Categories</b>				
Software engineering	X			
Middle-ware and distributed systems	X			

Surveyed URL	<a href="http://www.innovation.it">http://www.innovation.it</a>	<a href="http://www.ktu.lt/en">http://www.ktu.lt/en</a>	<a href="http://www.mokslas.cr.vu.lt/english">http://www.mokslas.cr.vu.lt/english</a>	<a href="http://www.mokslas.lt/english">http://www.mokslas.lt/english</a>
<b><u>Identification data</u></b>				
Dependable systems and infrastructures	X			
Knowledge and information management	X			
Agent technologies	X			
Optimization tools and decision support systems	X			
Content authoring tools	X			
Trust and security	X			
Supply chain management and generic organizational tools				
Simulation and CAD/CAM	X			
Embedded systems	X			
Internet technologies	X			
Network management	X			
Mobile and wireless communication	X			
Switching, routing and communication systems	X			
Computing architectures	X			
Micro- and opto-electronics	X			
Micro-systems and sensors	X			
Signal and data processing and conditioning	X			
Multi-model multi sensorial interfaces	X			
Languages and speech technologies	X			
Visualization, virtual environment and image processing	X			
Bio-informatics, neuron-informatics				
Others				
<b><u>Information Society Technology Categories</u></b>				
Finance/Insurance	X			

Surveyed URL	<a href="http://www.innovation.lt">http://www.innovation.lt</a>	<a href="http://www.ktu.lt/en">http://www.ktu.lt/en</a>	<a href="http://www.mokslas.cr.vu.lt/english">http://www.mokslas.cr.vu.lt/english</a>	<a href="http://www.mokslas.lt/english">http://www.mokslas.lt/english</a>
<b>Identification data</b>				
Electronic publishing and media industry	X			
Cultural sector	X			
Retail sector	X			
Administrations	X			
Social services	X			
Healthcare	X			
Education and training sector	X			
Tourism	X			
Energy	X			
Environment	X			
Transport	X			
Consumer electronics and home appliances	X			
Information technology manufacturing and service industry	X			
Telecom equipment industry	X			
Telecom operators and service providers	X			
Chemical industry	X			
Textile and clothing industry	X			
Agriculture, agri-food and fishery	X			
Pharmaceuticals and genetics	X			
Construction, engineering and architecture	X			
Others				

The only portal, which provides detailed information about RTD projects, is Lithuanian Innovation Center. But the only possibility to access project data is search, which is spare and provided only in English. Some information is provided in other portals as well. But there is a lack of collection in one place.

#### 4 IST RTD portal prototype functions

Most of analyzed portals didn't have multilingual capability and user-friendly classification of projects. These and detailed project information are the main facilities of national IST RTD portal prototype. Interface to input project information and edit other portal activities is provided to save maintenance costs. User has to be computer literate to input information and to use Internet browser.

Project information review doesn't need additional knowledge. But user must to read the manual to provide prototype administrator functions (see Figure 2).

System administrator creates, edits and removes system users, languages, forbids project information if erroneous data noticed.

Project administrator creates, edits and removes projects, assigns projects to selected classification group.

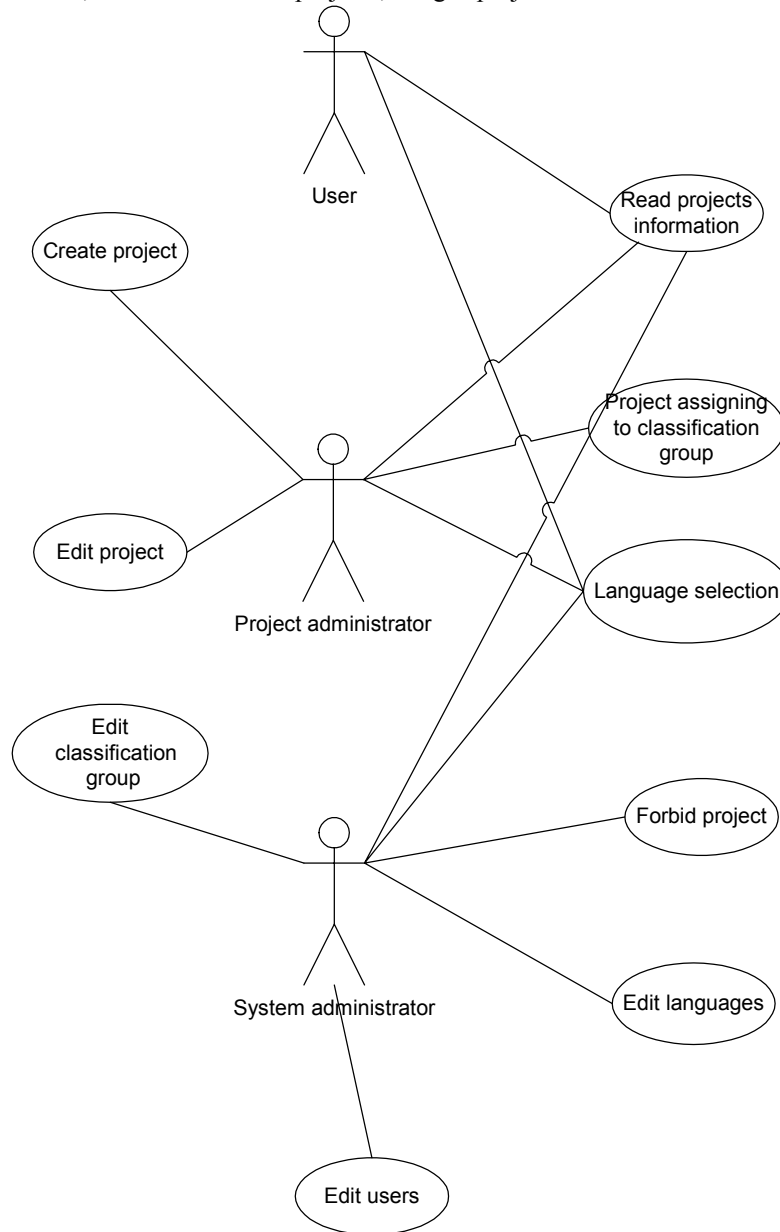
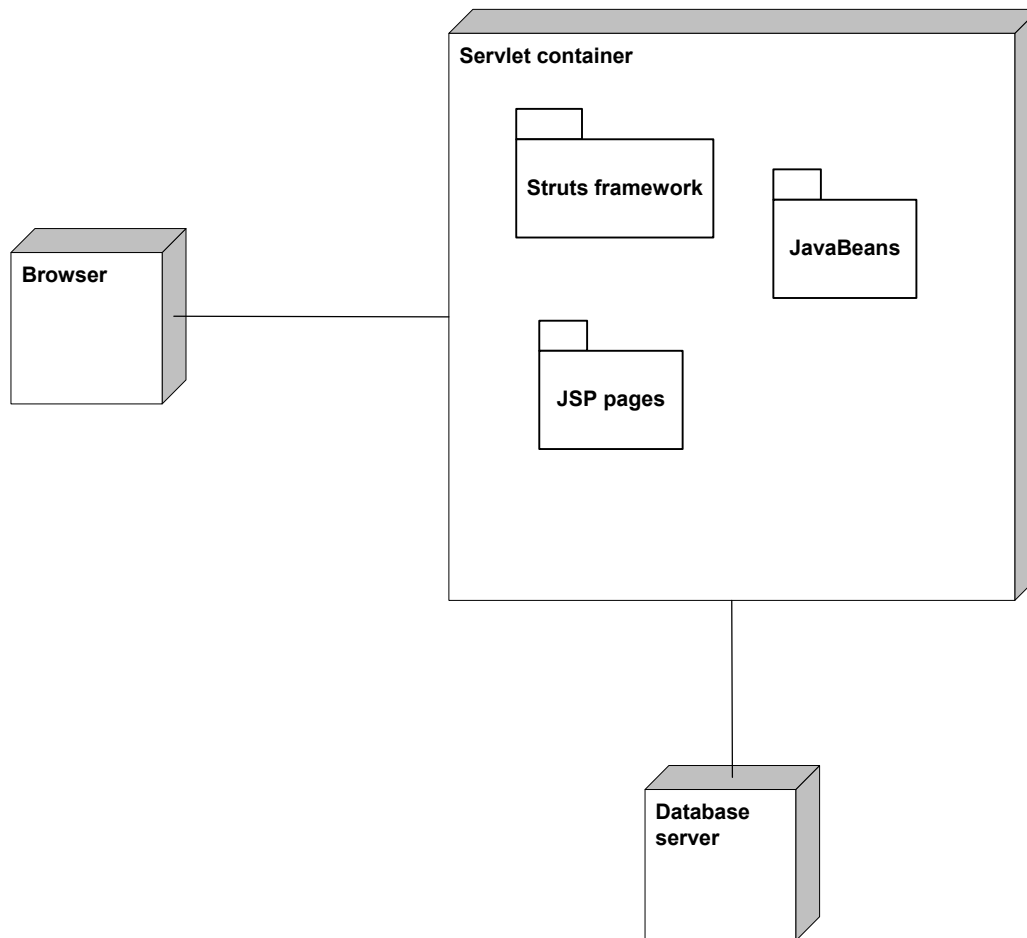


Figure 2. Use cases of ISTRTD portal

Prototype is provided as multi-tier system (see Figure 3). Browser is used for user interface presentation. All dialogs and information are provided to that tier. Database is for project, classification, user, language and other information storage. Servlet container is for all computation units storage and processing. These includes JSP, JavaBeans and Struts controller [9].

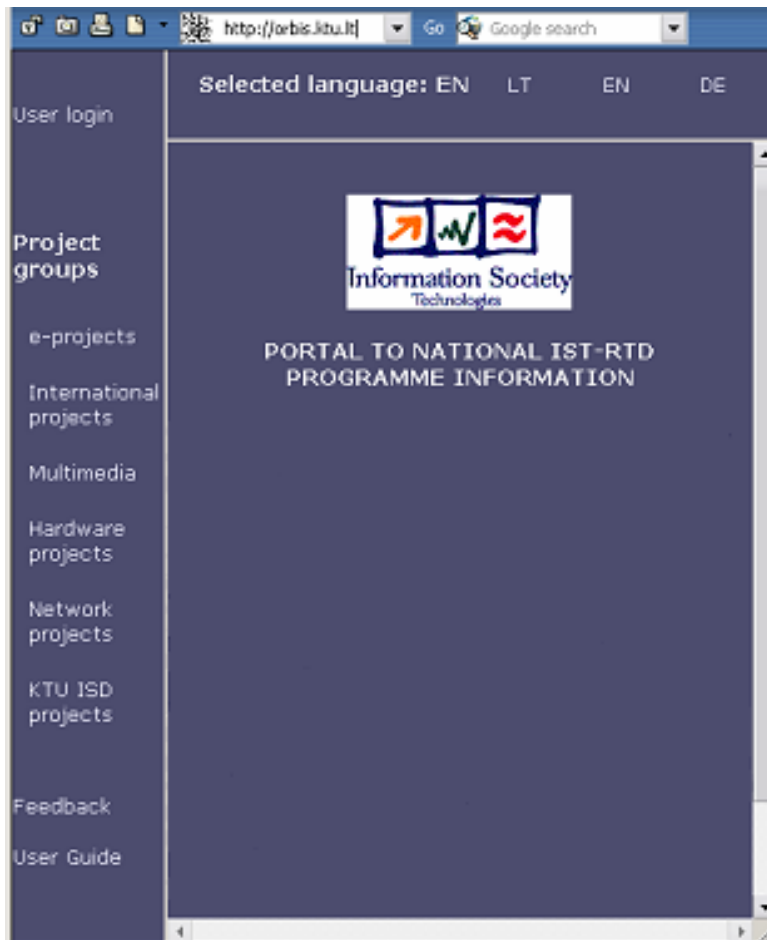
Prototype could be integrated to associated portal, or vice-versa, because projects classification needs to have more information provided to explain RTD context, research area or other topics.



*Figure 3. Context diagram of portal prototype*

Final prototype version provides functions described in figure 2. The result is portal providing ISTRTD information. Information is classified in research areas. The selection of current research area delivers involved RTD projects. Multilingual function provided as language list, which returns related language data, when some language is selected. There is a welcome screen in figure 4. You can see language list at the top of it and classification subjects at the left. Content management possibilities also provided delivering data input and editing possibilities to administrators.





*Figure 4. Prototype screenshot: welcome screen with languages list and classification groups*

## **5 Conclusions**

Analysis of related portals helped to detect main functions of created prototype and describe objectives. The objectives are oriented to stimulate business and science collaboration and provide a place for RTD information publishing.

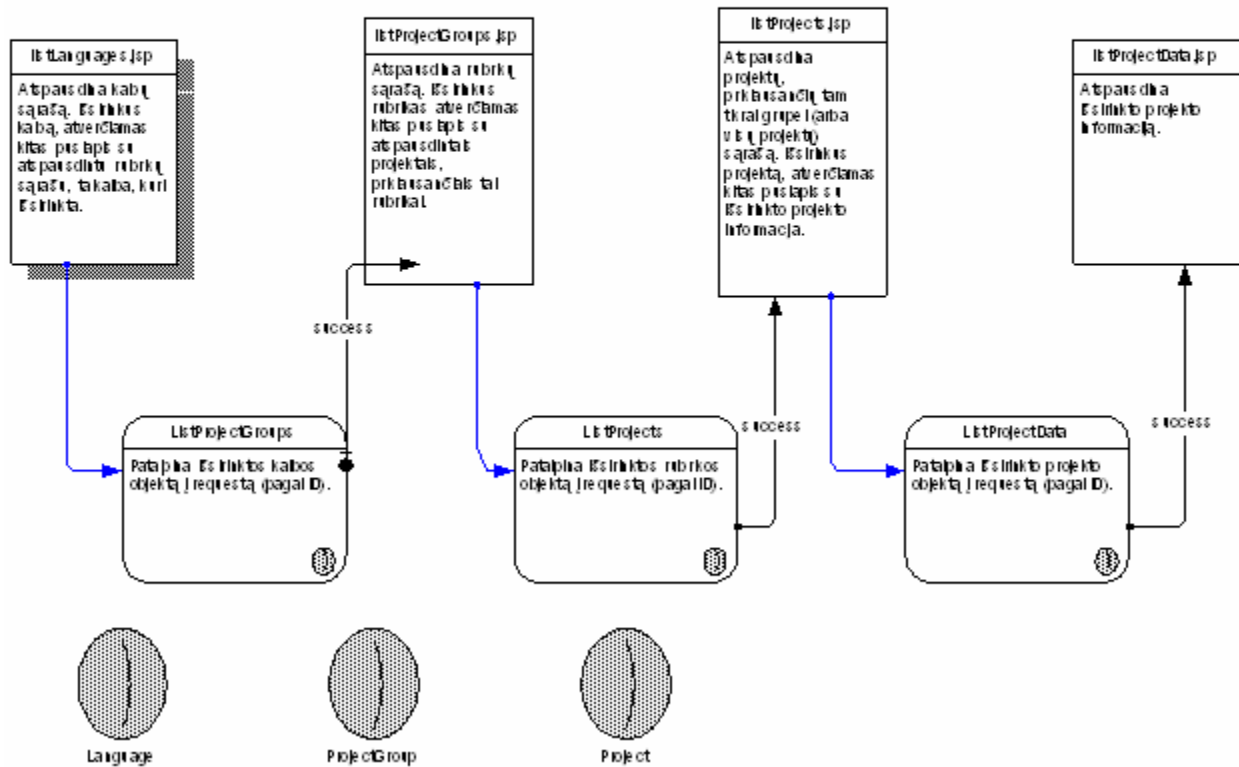
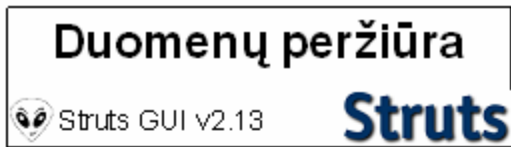
This is attempt to create e-Commerce portal for business and RTD representatives. Prototype solves scientists' problem to publish and advertise projects, communicate, find participants. Businessmen obtain possibility to get information about modern science fields.

## References

1. **CORDIS: Community Research & Development Information Service.** “IST and the European Research Area”. [Checked 2004 01 10] <http://www.cordis.lu/ist/about/era.htm>
2. **CORDIS: Community Research & Development Information Service.** “Sixth Framework Programme”. [Checked 2004 01 10] <http://fp6.cordis.lu/fp6/home.cfm>
3. **European Commission.** “The Sixth Framework Programme (2002-2006)”. [Checked 2004 01 10] [http://europa.eu.int/comm/research/fp6/index\\_en.html](http://europa.eu.int/comm/research/fp6/index_en.html)
4. **Ideal-IST .** FP6 Partner Search.. [Checked 2004 01 10] <http://www.ideal-ist.net/partnersearch.php>.
5. **Ideal-IST.** [Checked 2004 01 10] [http://www.ideal-ist.net/home/i\\_lithuania.php](http://www.ideal-ist.net/home/i_lithuania.php)
6. Integrating JSP/JSF and XML/XSLT: The Best of Both Worlds. [Checked 2004 01 10] <http://www.theserverside.com/resources/article.jsp?l=BestBothWorlds>
7. Lithuanian Innovation Centre. [Checked 2003 11 15] <http://www.innovation.lt>, November of 2003.
8. Prototype system of the IST Research portal. [Checked 2003 11 10] <http://193.74.145.219/EUPortal/>.
9. **Taujanskas V.** Bachelor work “Universal internet database”. Kaunas: Kaunas University of Technology, 2002.



## 9.3 Loginio vaizdo diagramos

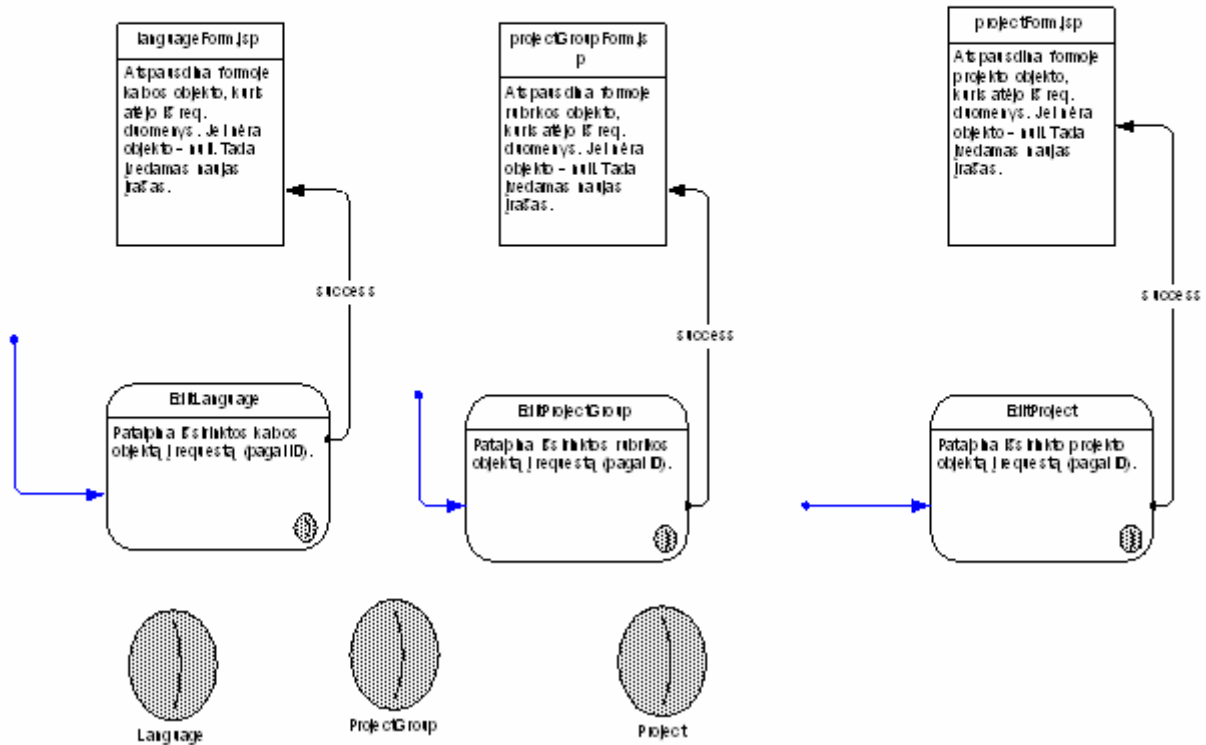
### 9.3.1 Duomenų peržiūra



### 9.3.2 Duomenų redagavimo procesas

## Duomenų redagavimas

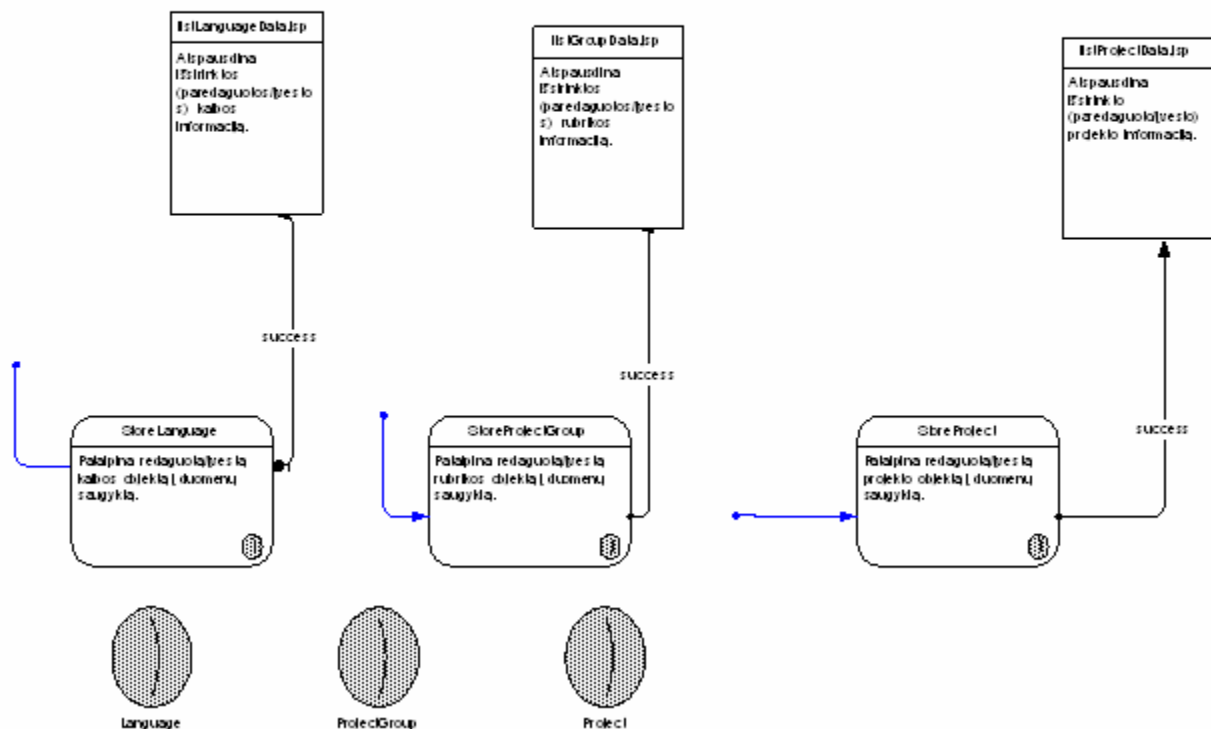

Struts GUI v2.1.3




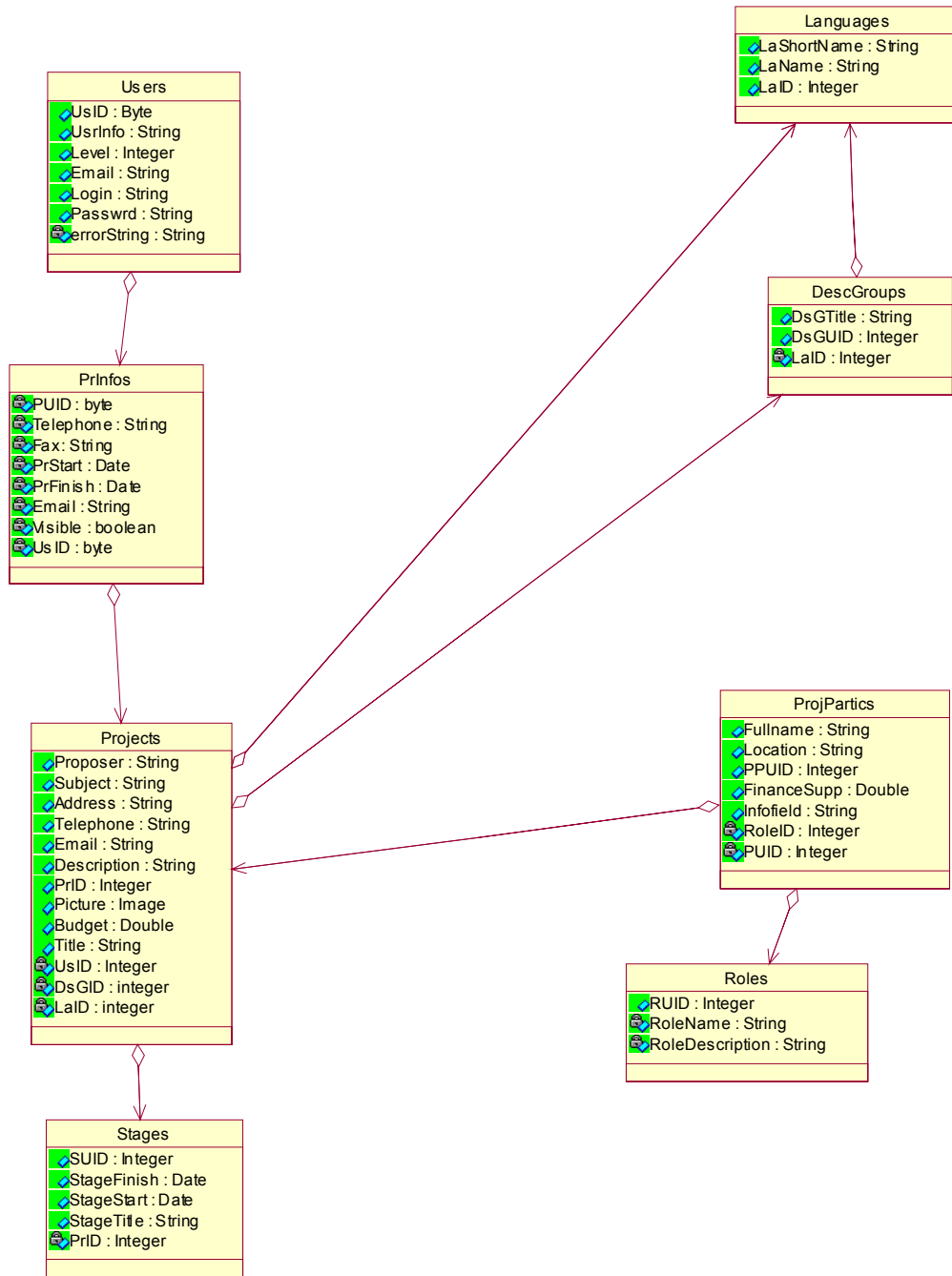
### 9.3.3 Duomenų išsaugojimo procesas

**Duomenų įvedimas į duomenų bazę**

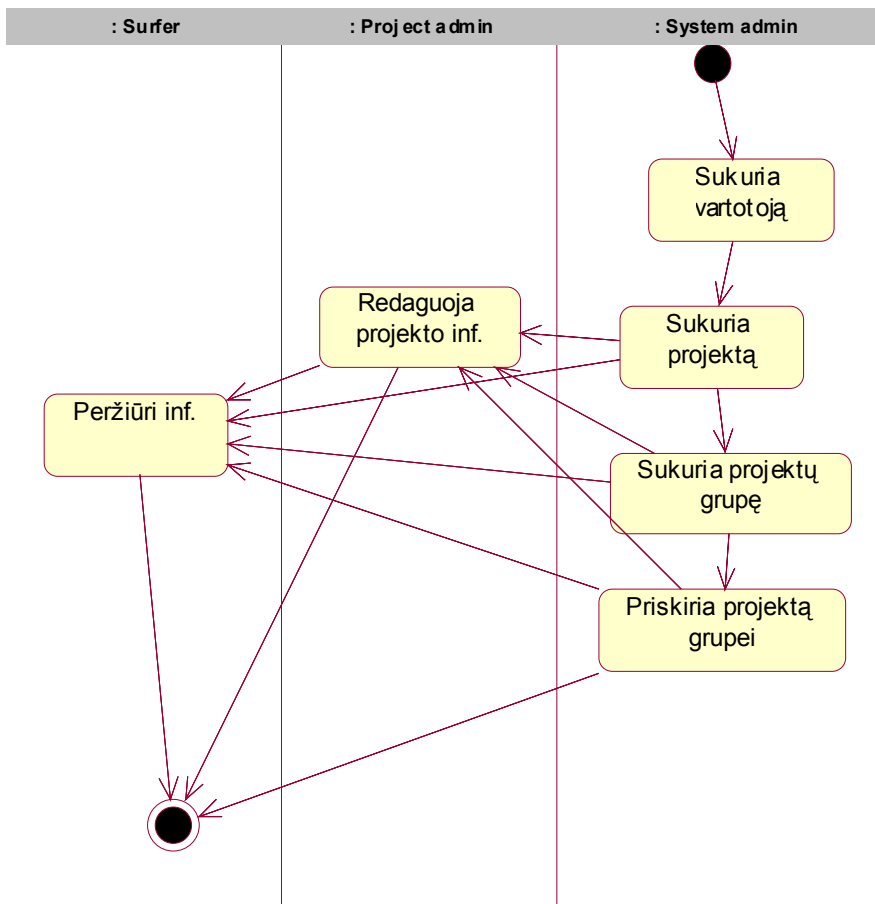
Struts GUI v2.13 **Struts**



### 9.3.4 Veiklos klasių diagrama



## 9.4 Projektų registravimo sistemos procesas



## 9.5 Taikomojo uždavinio struts-config.xml failas

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Struts Config XML - lesson1 -->
<!-- ===== -->
<!-- AutoGenerated from : d:\mokslai\karciauskas\pvz.vsd -->
<!-- AutoGenerated on   : 12-18-2003 04:22:05 -->
<!-- AutoGenerated by   : Struts GUI v2.13 (c)2002 Alien-Factory -->
<!--                   : See 'http://www.alien-factory.co.uk' for details -->
<!-- GET YOUR STICKY FINGERS OFF! i.e. Do not edit. -->

<!DOCTYPE struts-config PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 1.0//EN"
    "http://jakarta.apache.org/struts/dtds/struts-config_1_0.dtd">
<struts-config>
    <!-- ===== Form Bean Definitions
    ===== -->
    <form-beans>
        <form-bean     name="lesson1"
                       type="com.jspinsider.struts.lesson1.LogonForm">
            <description>UserForm klasė, naudojama paimti duomenis (login) iš
formas</description>
        </form-bean>
    </form-beans>
    <!-- ===== Global Forward Definitions
    ===== -->
    <global-forwards>
    </global-forwards>
    <!-- ===== Action Definitions
    ===== -->
    <action-mappings>
        <action path="/lesson1/logon"
                type="com.jspinsider.struts.lesson1.LogonAction"
                name="lesson1"
                scope="request"
                validate="false">
            <description>LogonAction klasė skirta, paimti duomenis (pasijungimo
vardą) iš LogonForm Bean'o ir patalpinti juos į sesiją.</description>
            <forward     name="logon"
                        path="/lesson1/home.jsp"/>
            <forward     name="home"
                        path="/lesson1/home.jsp"/>
        </action>
    </action-mappings>
</struts-config>
```



## 9.6 Sukurtos programų sistemos vartotojo sąsajos vaizdai

The screenshot displays a web application interface for project management. The browser address bar shows the URL `http://forbis.ktu.lt/index.jsp?lshortn=EN`. The page features a dark blue navigation menu on the left with the following items: User login, Project groups, e-projects, International projects, Multimedia, Hardware projects, Network projects, KTU ISD projects, Feedback, and User Guide. The main content area is titled "Selected language: EN" and includes the following sections:

- Phone:** +370 37 300397
- Fax:** +370 37 300352
- Email:** sakys@soften.ktu.lt
- Budget:** 10.0
- Project Start:** 2002-01-01
- Project Finish:** 2002-12-31

**Project participants:**

Fullname	Information	Location	Finance Support
KTU Department of Information Systems	Project participant	Lithuania	0.0
Aabenraa Rotary Club, Denmark	Project participant	Denmark	0.0
Rotary Club Kauno Tauras, Lithuania	Project participant	Lithuania	0.0

