

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Aivaras Bliūdžius

**Ontologija grindžiamo kūrimo metodo taikymas
interneto portalui kurti**

Magistro darbas

Darbo vadovas
prof. dr. Lina Nemuraitė

Kaunas, 2010

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Aivaras Bliūdžius

Ontologija grindžiamo kūrimo metodo taikymas internetu portalui kurti

Magistro darbas

Recenzentas
doc. dr. A. Janavčiūtė

2010-05-

Darbo vadovas
prof. dr. Lina Nemuraite

2010-05-

Atliko
IFM-4 gr. stud.
Aivaras Bliūdžius

2010-05-27

Kaunas, 2010

Turinys

IVADAS	6
1. MODERNIŲ METODŲ, TAIKOMŲ PORTALAMS KURTI, ANALIZĖ	8
1.1. PORTALŲ KŪRIMO METODŲ ANALIZĖ	8
1.2. ONTOLOGIJŲ KALBŲ IR ĮRANKIŲ ANALIZĖ	11
1.3. METODŲ LYGINAMOJI ANALIZĖ	12
1.4. ANALIZĖS IŠVADOS	12
2. ONTOLOGIJA GRINDŽIAMO PORTALŲ KŪRIMO METODAS	13
2.1. PROJEKTO ONTOLOGIJOS SUKŪRIMAS	13
2.2. ONTOLOGIJOS TRANSFORMAVIMAS Į PORTALO ARCHITEKTŪROS ELEMENTUS	14
2.3. ONTOLOGIJOS TRANSFORMAVIMAS Į DUOMENŲ BAZĖS SCHEMĄ	15
2.4. PORTALO REALIZAVIMAS	16
3. AUTOMATIZUOTO ĮRANKIO REALIZAVIMAS	17
3.1. AUTOMATIZUOTO ĮRANKIO FUNKCIJOS	17
3.2. AUTOMATIZUOTO ĮRANKIO STRUKTŪRA	18
3.3. AUTOMATIZUOTO ĮRANKIO TESTAVIMAS	20
4. ŽINIŲ PORTALO PROJEKTAS	21
4.1. EKSPERIMENTINĖS SISTEMOS ANALIZĖ	22
4.1.1. <i>Vartotojų aibė, tipai ir savybės</i>	22
4.1.2. <i>Vartotojų tikslai ir problemos</i>	22
4.2. REIKALAVIMŲ SPECIFIKACIJA	23
4.2.1. <i>Tikslų modelis</i>	23
4.2.2. <i>Panaudojimo atvejų modelis</i>	23
4.3. DALYKINĖS SRITIES MODELIS	30
4.4. SISTEMOS ARCHITEKTŪROS PROJEKTAS	30
4.4.1. <i>Sistemos loginė architektūra</i>	30
4.4.2. <i>Vartotojo paslaugos (sąsaja, sąsajos navigavimo planas)</i>	31
4.5. DETALUS PROJEKTAS	31
4.6. SISTEMOS ELGSENOS MODELIS	32
4.7. DUOMENŲ BAZĖS SCHEMA	34
4.7.1. <i>Portalo testavimas</i>	41
5. APGYVENDINIMO PORTALO PROJEKTAS	43
6. EKSPERIMENTAS IR METODO ĮVERTINIMAS	44
6.1. EKSPERIMENTO PROJEKTAS	44
6.2. DALYKINĖS SRITIES KŪRIMO EKSPERIMENTAS	44
6.2.1. <i>Žinių portalo eksperimentas</i>	44
6.2.2. <i>Apgyvandinimo portalo eksperimentas</i>	45
6.3. DALYKINĖS SRITIES PAKĖITIMŲ EKSPERIMENTAS	46
6.3.1. <i>Naujienu portalo dalykinės srities pakeitimų eksperimentas</i>	46
6.3.2. <i>Apgyvandinimo dalykinės srities pakeitimų eksperimentas</i>	47
6.4. EKSPERIMENTO IŠVADOS	48
7. IŠVADOS	49
8. LITERATŪRA	50
9. PRIEDAI	51
9.1. ŽINIŲ PORTALO ONTOLOGIJOS FAILO TURINYS	51
9.2. APGYVENDINIMO PORTALO ONTOLOGIJOS FAILO TURINYS	55

Applying Ontology Based Method for Internet Portal Development

SUMMARY

Application development is a process that becomes increasingly complex depending on the intricacy of the application being developed. Development techniques and methodologies exist to manage and control the complexity of this development process. Amongst the techniques to manage the complexity of the application development process is Domain-driven design (DDD).

The Master thesis describes a method for internet portal development where DDD method is supplemented with ontology based on project specification that describes domain models. To automate the application of ontology to the portal development an automated tool is created to read ontology file and create main files needed for DDD architecture to present domain.

Using the implemented automatic tool the project development time is saved and the mistakes are avoided that could be made by programmers when the domain objects are created by hand.

Key words: Domain-Driven Design, automated tool, ontology based development

Santrumpų žodynėlis

DDD - Domain Driven Design

SOA (service oriented architecture) - į paslaugas orientuota architektūra

OWL (Web Ontology Language) – žiniatinklio ontologijos kalba

Media - masinės informacijos priemonės

DO (domain object) – dalykinės srities objektas, pernešantis informaciją vartotojui

DTO (domain transfere object) – dalykinės srities pernešantis objektas, pernešantis informaciją iš web servisų

AI – automatinis įrankis

MDA (Model-driven Architecture) – OMG grupės pasiūlytas principas modeliais grindžiamam sistemų kūrimui

PI – programinė įranga

AOP (Aspect-oriented programming) – aspektinis programavimas

Ivadas

Taikomųjų programų kūrimas yra procesas, kuris tampa vis sudėtingesnis, priklausomai nuo reikalavimų skaičiaus ir jų kitimo dažnumo. Siekiant efektyviau kovoti su reikalavimų pakeitimais, taikomųjų programų sudėtingumą reikia valdyti. Brooks teigia, kad nors esminių sudėtingumų sumažinti negalima, kadangi jie yra neatskiriami nuo problemos, priešingu atveju problema būtų išspręsta, atsitiktinį sudėtingumą galima valdyti taikant projektavimui tinkamus architektūrinius sprendimus ir kūrimo procesus [1]. Norėdami reguliuoti atsitiktinį sudėtingumą, taikysime Domain Driven Design (DDD) modelį, kuris naudoja problemų atskyrimo principą. Šis principas yra naudojamas, kad būtų galima atskirti dalykinės srities modelio dalis, susijusias su funkcionalumu, ir nepriklausomas nuo funkcionalumo. Nuo funkcionalumo nepriklausomi dalykinės srities modeliai palengvina informacinių sistemų paslaugų (t. y. funkcijų) naudojimą.

Tyrimo sritis – ontologijos ir dalykinės srities modeliai, pagrįsti lengvai plečiamų sistemų principu, bei jų projektavimas.

Tyrimo objektas – žinių portalo kūrimo procesas, kuriuo sukurtas portalas galėtų nesunkiai atnaujinti ir papildyti portalo funkcionalumą. Prireikus papildomų funkcijų, portalas nesunkiai praplečiamas atnaujinant ontologiją ir perkeliant pakeitimus į portalą.

Darbo tikslas – automatizuoti DDD kūrimo metodą, papildant jį ontologijos kalba aprašytu dalykinės srities modeliu, leidžiančiu kurti lengvai plečiamus žinių portalus.

Darbo uždaviniai:

- Išanalizuoti DDD kūrimo metodą
- Papildyti DDD kūrimo metodą ontologijos integravimu
- Sudaryti dalykinės srities ontologijos specifikaciją
- Realizuoti automatizuotą įrankį, kuris apdorotų ontologijos specifikaciją ir transformuotų į portalo architektūros elementus
- Sukurti duomenų bazės struktūrą, kuri užtikrintų universalią struktūrą, priimančią pasikeitusius reikalavimus
- Atlikti eksperimentą, kuris patvirtintų automatizuoto įrankio naudingumą.

Tyrimo metodika. Darbe buvo taikoma literatūros analizė, lyginamoji analizė, DDD metodas, ontologijų kalbos, eksperimentinis tyrimas.

Buvo išanalizuoti MDA, AOP, DDD metodai. Pasirinkus DDD projekto kūrimo metodą, šiam metodui pritaikomas automatizuotas reikalavimų pakeitimų įvedimas į projektą. Automatizuotas įrankis sukuria dalykinės srities architektūrinius failus. Norint, kad duomenys su naujom duomenų struktūrom, kurie papildomai atsiranda dėl reikalavimų pakeitimų būtų nesunkiai saugomi duomenų bazėje, reikalinga universali duomenų bazės schema.

Naudojantis ontologijos principais, sukuriama universali duomenų bazės struktūra. Duomenų bazės schema paremta dokumento, atributo ir atributo reikšmės ryšiu.

Naudojant DDD kūrimo metodą, sukuriamas žinių portalas. Ištirti kokią naudą projekto kūrime suteikia automatizuotas įrankis, vykdomas eksperimentas. Eksperimento metu lyginamas laikas, reikalingas įvesti pakeitimus dalykinėje srityje naudojantis automatizuotu įrankiu ir įvedant pakeitimus rankiniu būdu.

Darbo struktūra:

- Pirmame skyriuje atlikta portalų kūrimo analizė, palygintos skirtingų projektų kūrimo metodas, pagrįstos dalykine sritimi, atlikta ontologijos kalbų ir įrankių analizė, pateiktos analizės išvados.

- Antrame skyriuje pateikiamas ontologija grindžiamo portalų kūrimo metodas. Aprašomi projekto ontologijos sukūrimo žingsniai, kaip ontologija transformuojama į portalo architektūros elementus, aprašoma duomenų bazės schema ir portalo realizacija.

- Trečiame skyriuje aprašoma automatizuoto įrankio realizacija, jo panaudojimas, struktūra ir testavimo žingsniai.

- Ketvirtame skyriuje pateiktas žinių portalo projektas su vartotojų analize, reikalavimų specifikacija, architektūros aprašymu ir portale naudojama duomenų bazės schema.

- Penktame skyriuje pateiktas apgyvendinimo portalo projektas.

- Šeštame skyriuje vykdomas eksperimentas ir metodo įvertinimas naudojant automatizuotą įrankį. Lyginama laiko trukmė, užtrunkanti pakeitimus įvesti naudojantis automatizuotu įrankiu ir analogiškus pakeitimus įvedant rankiniu būdu. Pateikiamos eksperimento išvados.

- Septintame skyriuje pateikiamos magistro darbo išvados.

- Aštuntame skyriuje surašyta naudota literatūra.

- Devintame skyriuje pateikiami priedai. Pateikiami ontologijos failų turiniai.

Modernių metodų, taikomų portalams kurti, analizė

1.1. Portalų kūrimo metodų analizė

Magistro darbe analizuojami MDA, AOP, DDD kūrimo metodai.

MDA – tai modeliavimo principai, kuriuos apibrėžė objektų valdymo grupė (Object Management Group (OMG))[2]. Šie principai aprašo modelių, specifiкуotų viena kalba transformavimo į kitomis kalbomis specifiкуojamus modelius taisykles. Tokių modelių transformavimo pavyzdžiai: duomenų bazės schemas generavimas, kodo generavimas remiantis UML modeliu.

MDA PĮ kūrimo procesas yra paremtas PĮ sistemos modeliavimu. MDA procesą sudaro trys žingsniai:

- abstraktaus, nepriklausomo nuo diegimo technologijos modelio sudarymas. Šis modelis vadinamas nuo platformos nepriklausomu modeliu (Platform Independent Model (PIM));
- PIM modelio transformavimas į vieną ar daugiau modelių, įvertinančių konkrečią diegimo technologiją, pvz: duomenų bazės modelis ar EJB (Enterprise Java Beans) modelis. Tokie modeliai vadinami specifiniais platformai modeliais (Platform Specific Models (PSM));
- PSM transformavimas i programos kodą.

PIM modelis, kartais dar vadinamas dalykinės srities modeliu, išreiškia išimtinai veiklos funkcionalumą ir elgseną. Šis modelis kuriamas bendradarbiaujant probleminės srities ekspertams ir IS kūrėjams.

MDA proceso tikslas – formuoti tikslus sistemos modelius, nes kitu atveju PIM transformacija į PSM būtų neįmanoma. Norint išpildyti šiuos MDA reikalavimus, modeliuoti naudojant tik UML nepakanka. Žymiai kokybiškesni modeliai gaunami derinant UML kartu su OCL.

AOP metodas stengiasi spręsti bendros paskirties funkcijų kodo persipynimo ir išsibarstymo problemą, leisdama programuotojui aprašyti bendrosios paskirties funkcijas savarankiškuose moduluose, vadinamuose aspektais.

Tam, kad visą reikalingą informaciją galėtume talpinti viename modulyje – aspekte, turime turėti atsakymus į šiuos klausimus:

- 1) **Kur ir kada** vykdyti bendros paskirties funkciją?
- 2) **Kaip** sklandžiai įkomponuoti bendrosios paskirties funkciją į pagrindinį programos kodą?

3) **Ką** (kokį kodą) vykdyti, pasiekus norimą prijungimo tašką?

Aspektinis (AO) projektavimas [3] turi išskeltus tuos pačius tikslus kaip ir bet kuris kitas programinės įrangos projektavimas: specifikuoti programų sistemos elgesį ir struktūrą. AO projektavimas unikalus tuo, kad bendrosios paskirties funkcijos gali būti išskaidytos į atskirus modulius, ko nebūtų įmanoma padaryti tradiciniais projektavimo metodais. Dažniausiai tokia metodika apima tiek procesą, tiek kalbą. Į procesą kaip įėjimas paduodami reikalavimai, o išėjime gaunamas projekto modelis. Sukurtas projekto modelis vaizduoja bendros paskirties funkcijas ir jų sąryšius tiek tarpusavyje, tiek su kitomis funkcijomis. Projektavimo kalba pagelbėja tuo, jog joje aprašytos konstrukcijos leidžia atvaizduoti šiuos elementus ir sąryšius tarp jų. Kitaip tariant, projektavimo kalbos konstrukcijos leidžia suskirstyti į modulius bendrosios paskirties funkcijas, atsižvelgiant į jų tarpusavio sąveiką ir galimus konfliktus. Nepaisant visų šių naujų AO projektavimo kalbos savybių, individualių modulių projektavimas prilygsta standartiniam komponentiniam projektavimui.

Svarbios AO projektavimo sritys:

- Pats AO projektavimo procesas
- AO projektavimo notacija;
- AO projektavimą palengvinantys įrankiai;
- AO projektavimo įsisavinimas ir integravimas;
- AO projektavimo įvertinimas.

AOP apima ir programavimo metodus, leidžiančius iškelti bendrosios paskirties funkcijas į atskirus modulius išėities tekstų lygmenyje. Taip kaip ir bet kuri kita programavimo kalba, aspektiškai-orientuota kalba dažniausiai susideda iš dviejų dalių: kalbos specifikacijos ir realizacijos. Taigi, tyrimai vykdomi dviem kryptimis: AOP kalbos tyrimai, AOP kalbų taikymų tyrimai.

Domain-Driven Design yra projektų kūrimo metodas, kuris pagrindinį dėmesį skiria sistemos sudėtingumo valdymui, išskirdamas architektūrinius elementus, kurie palengvina dalykinės srities logikos realizavimą. Naudojant DDD siekiama suteikti priemones, padedančias palengvinti priežiūrą, paspartinti projekto vystymą ir pridėti lankstumą ir atsparumą reikalavimų pasikeitimuose. Naujų funkcijų pridėjimas [4] į projektą yra daugiausia problemų sukeliantis veiksnys. Tikimybė, kad gali reikėti naujų funkcijų ar paslaugų sukūrimo, yra didesnė. Kadangi paslaugų pakeitimai paveikia mažesniąją dalį viso kūrimo proceso, dėl to sumažėja ir bendro kūrimo proceso apimtis. Pokyčiai verslo taisyklėse gali būtų lengviau įvesti, nes jie yra centralizuoti dalykinės srities modeliuose. Ši

centralizacija natūraliai įveda trūkumus, nes pokyčiai dalykinės srities modelių įgyvendinimui gali turėti įtakos visų paslaugų realizacijoms. Po dalykinės srities modelių pasikeitimo paslaugos ir jų klasės, kurios naudoja pasikeitusius modelius, turi būti patikrintos, ar teisingai gražina rezultatus.

Dalykinės srities modeliavimo projektas paprastai susideda iš šių etapų:

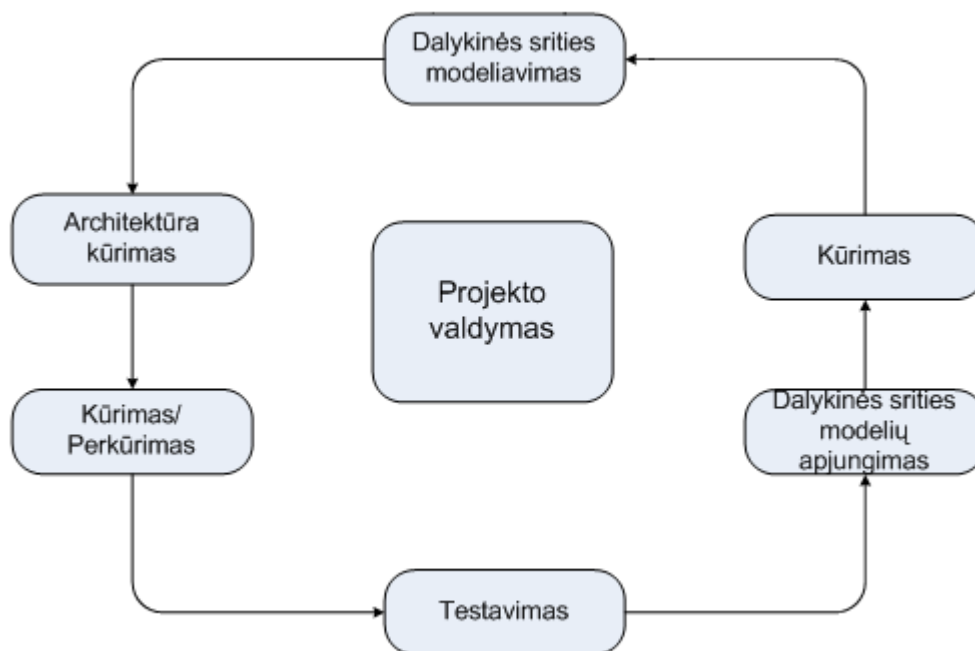
- Modelių ir verslo procesų nustatymas
- Modelių aprašymas UML kalba
- Nustatyti paslaugas, reikalingas verslo procesui. Šios paslaugos gali būti vienos pakopos arba kelių žingsnių su arba be darbo srautų.
- Objektų elgesio nustatymas ir dokumentavimas, naudojamų paslaugų panašumo įvertinimas

Svarbu išlaikyti modelį aukšto lygio, pirmiausia dėmesys skiriamas pagrindinių elementų verslo srityje.

Nuo projekto valdymo iki praktinio DDD įgyvendinti projektą sudaro tos pačios fazės kaip ir bet kurio kito programinės įrangos kūrimo projekto. Etapai yra šie:

- Dalykinės srities projektavimas
- Kūrimas
- Testavimas
- Patikslinimas dalykinės srities, architektūros atnaujinimas
- Pakartojimas aukščiau aprašytų veiksmų, naudojant atnaujintą dalykinės srities modelį.

Lanksčiojo programavimo metodika puikiai tinka, nes dažnai besikeičiantys reikalavimai leidžia sutelkti dėmesį į dalykinės srities naudojimą, taikant DDD. Projekto ciklas, naudojamas kuriant projektus pateiktas 1 pav.



1 pav. DDD projektų ciklas

1.2. Ontologijų kalbų ir įrankių analizė

Ontologijos kalba – tai formali kalba, skirta ontologijos kurti. Tinkamos ontologijos kalbos pasirinkimas priklauso nuo dalykinės srities ir ontologijos kūrimo tipo. Kalba turi būti suprantama, lengvai išmokstama, suderinama su kitais įrankiais. Konceptai turi būti nepriklausomi nuo sintaksės. Taip pat svarbu, kad kalba būtų suderinama su standartais: XML, XML Schema, RDF ir UML [4].

Kuriant ontologijas reikalingos žinios. Jos išgaunamos įvairiais būdais [5]:

- Interviu (kalbamasi su atitinkamos srities žmonėmis, bandoma išgauti kuo daugiau naudingos informacijos, tačiau jų nespaudžiant t. y. leidžiama kalbėti, pasakoti žmogui tol, kol stoja ilga tyla. Interviu tęsiamas toliau trumpais žurnalistiniais klausimais: kodėl? kada? kas? kur? dar kas nors? Ir pan.);
- Stebėjimais (atliekami stebint atitinkamų darbuotojų kasdieninę veiklą. Priklausomai nuo „aktorių“ situacijos gali būti skirtingos, tarkim naujas darbuotojas ieškos informacijos, vadovas bandys paaiškinti ir t. t.);
- Dokumentų analize (iš egzistuojančių dokumentų išrenkama reikalinga informacija);
- Anketomis bei apklausomis (dalykinės srities atstovams užduodami anksčiau gerai apgalvoti, lengvai suprantami klausimai);
- Kolektyvinių naujų idėjų generavimu bei diskusijomis (kartais naudojamas ir minčių, idėjų rašymas ant popieriaus anonimiškai).

Ontologijų kūrimui [7] sukurta daug įrankių: „Apollo“, „LinkFactory“, „OILED“, „OntoEdit“, „Ontolingua Server“, „OntoSaurus“. Šiame darbe bus naudojama vienas iš naujausių ontologijos kūrimo įrankių „Protégé 2000“

Yra skiriamos dvi įrankių šeimos [6]. „OILED“ ir „OntoSaurus“ priskiriami vienai šeimai (logika paremti įrankiai), o likusieji – kitai (paremti karkasu bei pirmos eilės logika). Pagrindinio įrankio žinių modelio išraiškingumas taip pat labai svarbus. Visi įrankiai leidžia reprezentuoti klases, ryšius, atributus, aksiomas. Tik „Ontolingua Server“, „OntoSaurus“ ir „Protégé 2000“ suteikia lanksčius modeliavimo komponentus (metaklases).

Portalų ontologijas kurti nuspręsta su „Protégé 3.4.1“ („Protégé 2000“ naujesnė versija). Klasių medžio valdikliai leidžia greičiau ir paprasčiau naviguoti per klasių hierarchiją, klasių savybes.

1.3. Metodų lyginamoji analizė

Išanalizavus skirtingus projektų kūrimo metodus, atliekama lyginamoji analizė.

1 lentelė. Metodų lyginamoji analizė

	MDA	AOP	DDD
Didelis dėmesys dalykinei sričiai	+	-	+
Dėmesys aspektams	-	+	+
Lengvas kūrimo metodo įsisavinimas	+	-	+
Išvengia papildomų dalykinės srities transformacijų	-	+	+

1.4. Analizės išvados

1. Iš tirtų metodų DDD tinkamiausias žinių portalo realizacijai, kadangi DDD taiko dalykinės srities priklausomas nuo funkcionalumo atskyrimą nuo dalykinės srities, nepriklausomas nuo funkcionalumo.
2. Naudojant DDD bus lengviau įgyvendinti reikalavimų pasikeitimus, keičiant dalykinę sritį.
3. Norint pakeitimus įvesti naudojant MDA, reikia naudoti daugiau transformacijų, kas reikalauja didesnių laiko sąnaudų projekto vykdyme.
4. Naudojant AOP iškyla problemų, kad programuotojai gali neteisingai interpretuoti aspektus. Jei norime išvengti klaidų, turime garantuoti, kad programuotojai perskaite aspekto kodą suprastų, ką jis daro. Daugelis programuotojų gali taip ir neįprasti prie tokio programų kūrimo modelio, kai pasikeitimai į kodą įterpiama vėliau, kai sistema jau sukurta.
5. DDD metodui patobulinti pasirinkta OWL kalba ir ontologijų kūrimo įrankis Protégé, kadangi tai plačiausiai naudojama kalba ir labiausiai ištobulintas įrankis.

2. Ontologija grindžiamo portalų kūrimo metodas

Šiame skyriuje aprašytas ontologija grindžiamo portalų kūrimo metodas, kuris susideda iš šių žingsnių:

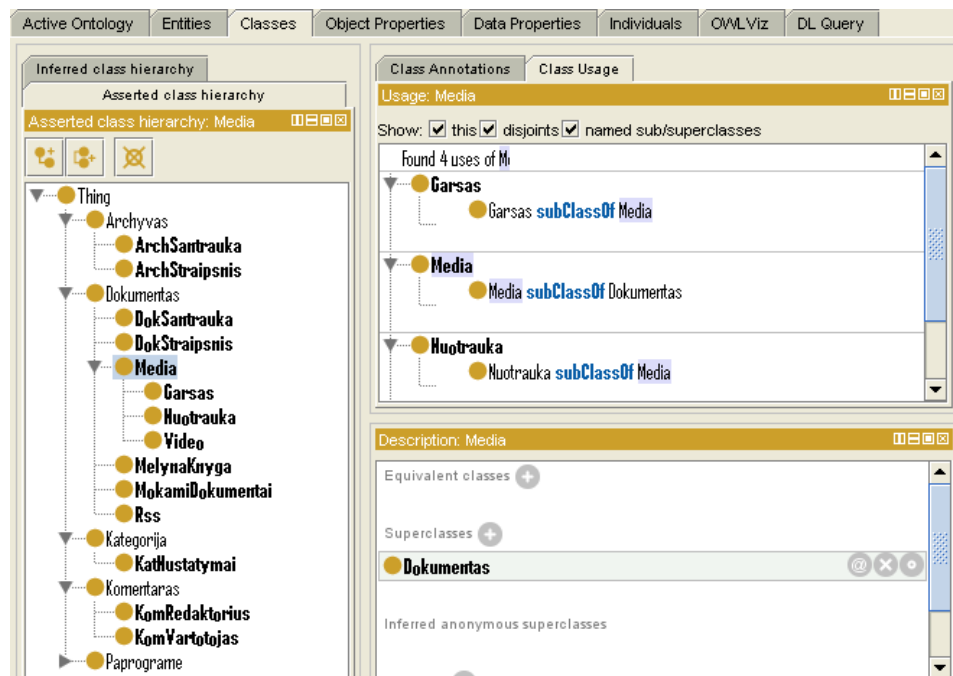
1. Ontologijos sukūrimas
2. Ontologijos transformavimas į dalykinės srities objektus
3. Ontologijos transformavimas į duomenų bazės schema.

2.1. Projekto ontologijos sukūrimas

Kuriant projekto ontologiją vykdomi šie ontologijos kūrimo žingsniai naudojantis programa „Protégé- 3.4.1“:

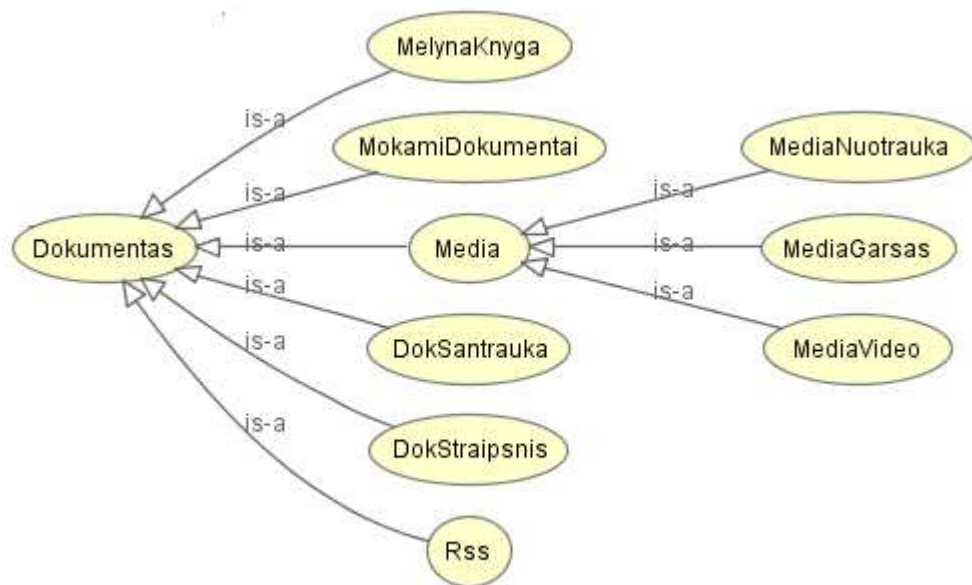
1. Sukuriamos klasės
2. Sukuriami duomenų savybės
3. Duomenų savybės priskiriamos klasėms
4. Sukuriami objektų savybės
5. Objektų savybės priskiriamos klasėms

Kuriant klases galimas klasių paveldimumas, detali informacija apie klasių panaudojimą (2 pav.).



2 pav. Klasių su paveldėjimais sukūrimas

„Protégé“ programoje ontologijos duomenų parametrai, kurie priskirti klasėms. Priskyrimą identifikuoja „domain“, „range“ – parametro tipas (3 pav.).



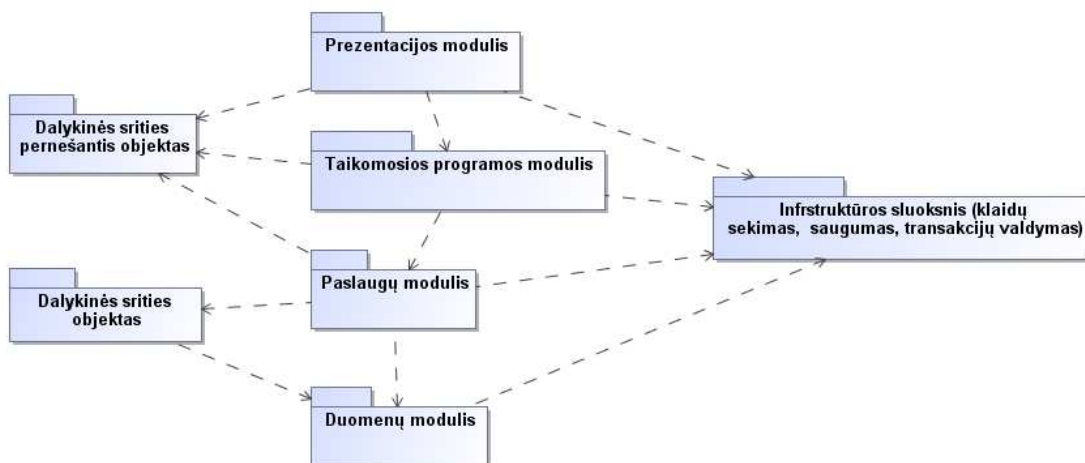
3 pav. Naujienų portalo OWL klasės

2.2. Ontologijos transformavimas į portalo architektūros elementus

Remiantis praktine DDD architektūra, architektūra turi dalykinės srities pernešančius objektus, kuriuos naudoja servais, dalykinės srities objektus, kuriuos naudoja duomenų modelis (4 pav.), apjungiamuosius objektus, kurie naudojami iš dalykinės srities pernešančių objektų informaciją perteikti į dalykinės srities objektus. Išvardintos architektūros dalys yra tiriamos magistro darbe.

Kadangi dalykinės srities modelius kurti yra monotoniškas darbas, magistro darbe šią sritį automatizuosime. Automatizavimas padeda išvengti klaidų, kurias programuotojai gali padaryti kurdami dalykinės srities objektus rankiniu būdu.

Turint dalykinės srities aprašymą ontologija, naudojantis automatizuotu įrankiu galima sukurti pagrindinius DDD architektūros komponentus, failus. Iš ontologijos sukuriama dalykinės srities objektai. Norint iš duomenų bazės gauti duomenis, naudojama paslaugų modeliu, kuriam reikalingi pernešantys dalykinės srities objektai. Žinant dalykinės srities objektus ir aprašymą, sukuriama pernešantys dalykinės srities objektai.



4 pav. Sistemos loginė architektūra

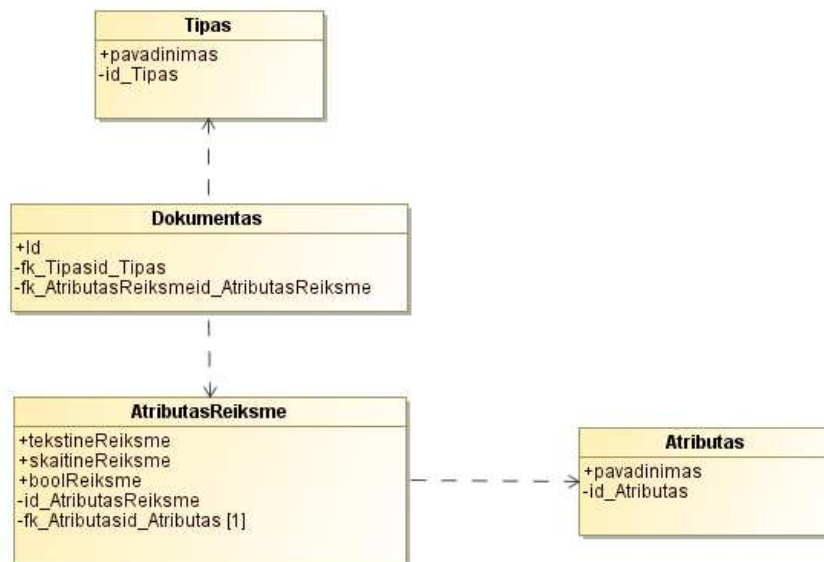
4 pav. pateikta architektūra suskirstyta į:

- Prezentacijos modulį, kuris atsakingas už informacijos pateikimą vartotojui ir vartotojo komandų vykdymą
- Taikomųjų programų modulį, kuris koordinuoja programos veiklą. Taikomųjų programų modulyje nevykdoma veiklos logika, bet žino vykdomo objekto statusą.
- Paslaugų modulyje yra web servisai, kurie vykdo užklausas į duomenų modulyje esančią duomenų bazę
- Duomenų modulis atsakingas už duomenų saugojimą/perdavimą.

2.3. Ontologijos transformavimas į duomenų bazės schemą

Stebint architektūros komponentus, pernešantys dalykinės srities objektai komunikuoja su paslaugomis, pernešantys dalykinės srities objektais sąveikauja su dalykinės srities objektais per jungiamuosius objektus. Kadangi mes turime dalykinę sritį, kuri gali būti nesunkiai atnaujinta, naudojantis automatizuotu įrankiu, turim užtikrinti, kad dalykinės srities pakeitimai kuo mažiau veiktų paslaugų lygmenį. Norint tai užtikrinti, turi būti sukurta universali duomenų bazė, kuri pajėgtų aprėpti besikeičiančius reikalavimus dalykinėje srityje.

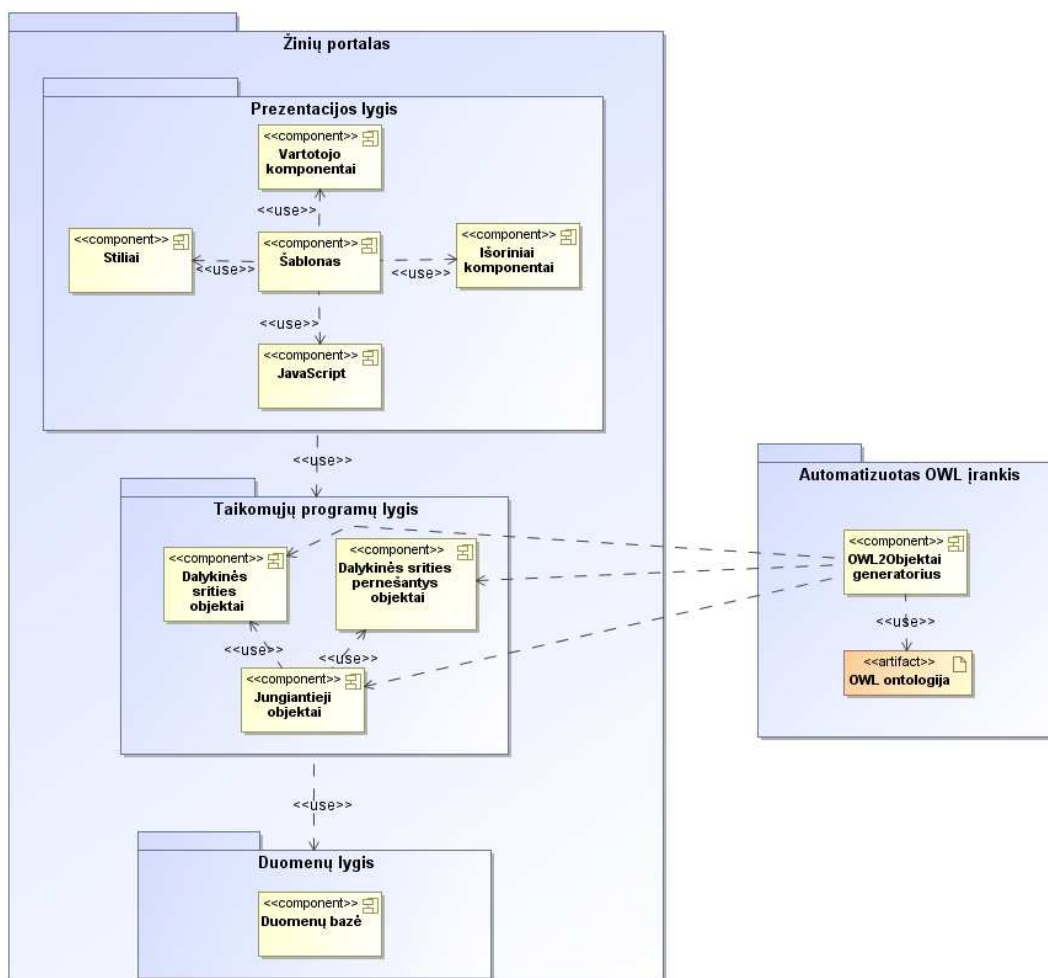
Žvelgiant į ontologiją iš praktinės pusės, ontologija sudaryta iš objektų, objektai sujungiami per predikatus su jų savybėmis, duomenys užpildomi per papildomus objektus su predikatais. Analogiškai sukuriama duomenų bazės schema, kuri tenkintų ontologijos principus. Duomenų bazė turi lentelę „Dokumentas“, kuri ontologijoje atstoja objektą, „Atributas“, ontologijoje atstoja predikatą, „AtributasReiksme“ susieja ontologijos objektą ir predikatą, kuris taipogi turi kažkokią reikšmę (5 pav.).



5 pav. Bazinės ontologijos duomenų bazės schema

2.4. Portalo realizavimas

Kuriant žinių portalą, programuotojams reikia sukurtus komponentus panaudoti projekto kūrime (6 pav.).



6 pav. Automatizuoto įrankio sukurtų komponentų panaudojimas portalo kūrime

Projekto kūrimo žingsniai:

- Sukurti duomenų bazės schemą pagal pavyzdinę schemą.
- Nukopijuoti sukurtą paslaugų sluoksnį į projektą.
- Panaudojus automatizuotą įrankį, sugeneruotus failus nukopijuoti į reikiamą architektūrinį sluoksnį.
- Sukurti kreipinius į paslaugą, kuri gauna duomenis iš duomenų bazės.
- Naudojant dalykinės srities objektus sukurti vaizdavimo (vartotojo sąsajos) lygį.

Automatizavimo įrankis ypač aktualus, kai reikia keisti jau sukurtą portalą ir yra komponentų, kuriuos pakartotinai galima panaudoti. Kuriant žinių portalą, visus komponentus teko sukurti nuo pradžių.

3. Automatizuoto įrankio realizavimas

3.1. Automatizuoto įrankio funkcijos

Automatizavimo įrankis iš sukurtos ontologijos (ontologijos kūrimui panaudojus „Protégé“ programą) gali nuskaityti klases ir duomenų savybes, priskirtas klasėms su šiais duomenų tipais:

- Sveikas skaičius
- Realus skaičius
- Tekstinis laukas
- Taip/Ne

Norint atlikti automatizuotą fizinių klasių kūrimą, automatizuotas įrankis vykdo tokias operacijas:

1. Nuskaito ontologiniame faile sukurtas duomenų savybes. Savybių aprašymas owl:

```
<owl:DatatypeProperty rdf:about="#Antraste">  
  <rdfs:range rdf:resource="&xsd:string"/>  
  <rdfs:domain rdf:resource="&owl;Thing"/>  
</owl:DatatypeProperty>
```

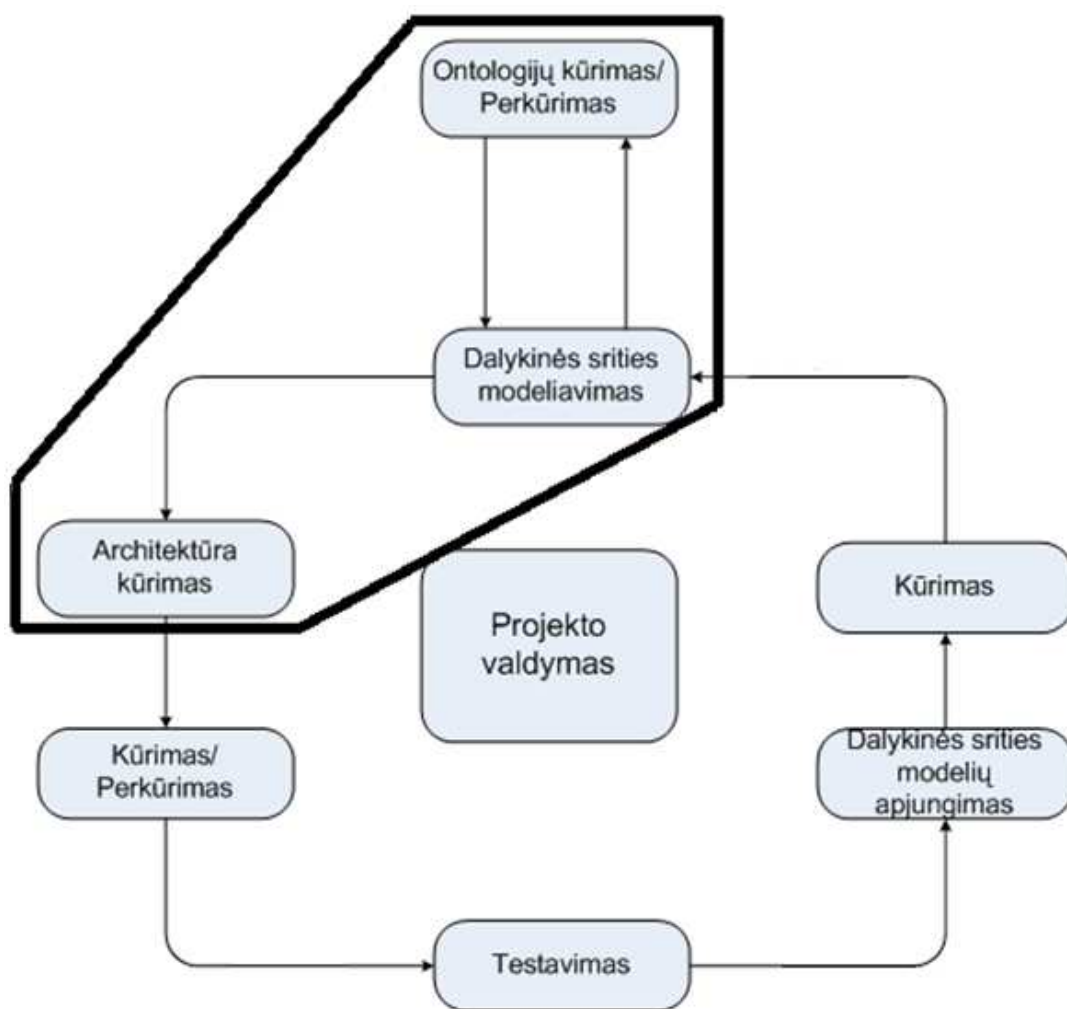
2. Sukuria klasių hierarchiją. Klasės aprašymas owl:

```
<owl:Class rdf:about="#Aukcionai">  
  <rdfs:subClassOf rdf:resource="&#Meno_rinka"/>  
</owl:Class>
```

3. Priskiria duomenų parametrus klasėms
4. Atspausdina klasių struktūrą su parametrais
5. Sukuria dalykinės srities pernešančius objektus

6. Atspausdina dalykinės srities pernešančius objektus
7. Sukuria dalykinės srities objektus
8. Atspausdina dalykinės srities objektus
9. Sukuria jungiamuosius objektus
10. Atspausdina jungiamųjų objektų turinius

Stebint DDD projekto ciklo modelį (7 pav.), automatizavimo įrankis apjungia ontologijų kūrimas/perkūrimas, dalykinės srities modeliavimas, architektūros kūrimas procesus.

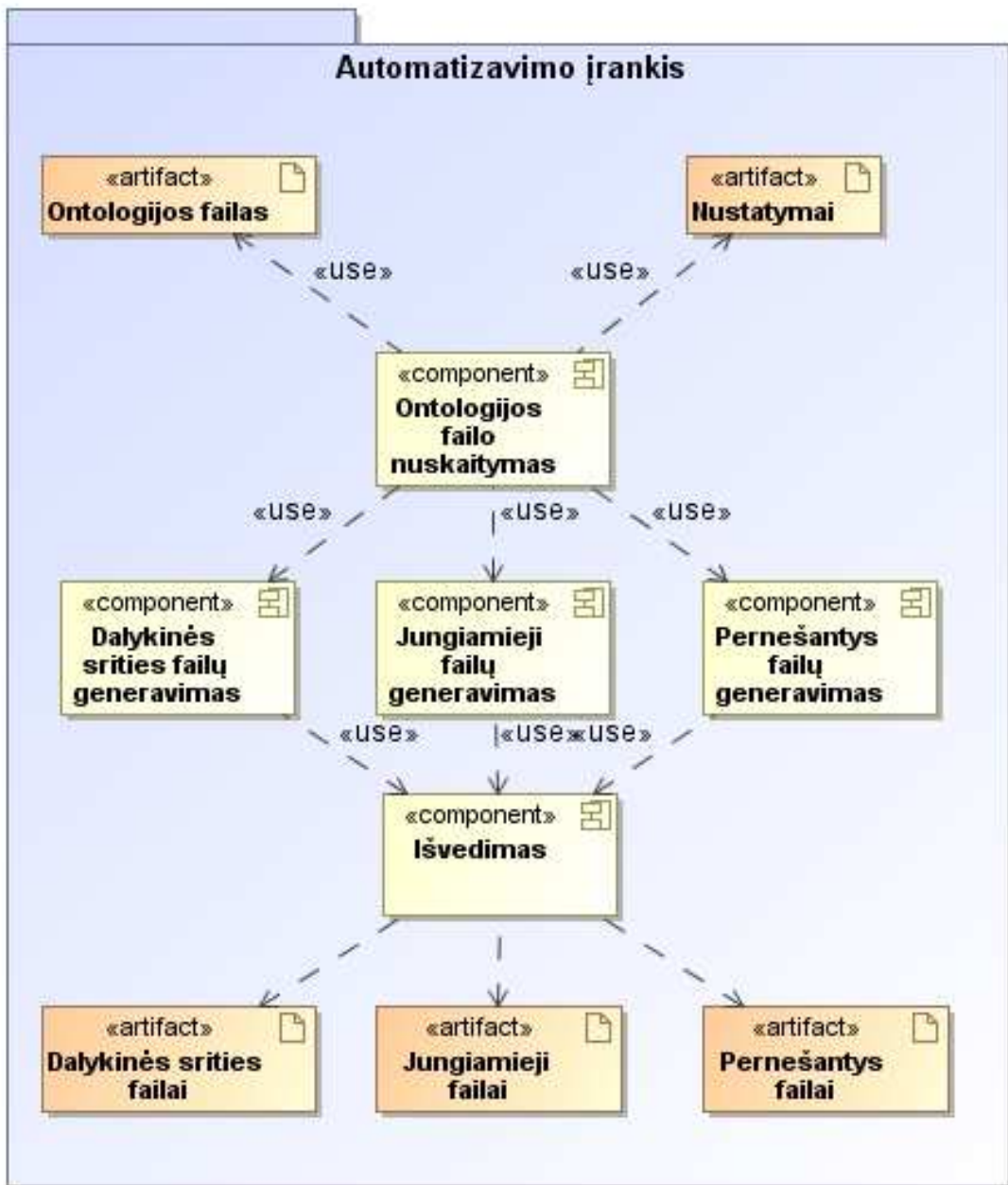


7 pav. Įrankio taikymas projekto vykdymo cikle

3.2. Automatizuoto įrankio struktūra

Automatizavimo įrankis, atliekantis dalykinės srities sukūrimą iš ontologijos, suskirstytas į nustatymų, ontologijos failo nuskaitymo, dalykinės srities failų, jungiamųjų failų, pernešančių failų ir išvedimo komponentus. Automatizuotam įrankiui perdavus apdoroti

ontologijos failą, sugeneruojami trijų tipų fiziniai failai: dalykinės srities, jungiamųjų, pernešančių (8 pav.).



8 pav. Automatizavimo įrankio struktūra

3.3. Automatizuoto įrankio testavimas

Automatizuoto įrankio testavimas apima fizinių klasių sukūrimo testavimą iš ontologijos failo. Norimas rezultatas yra teisingas, jei “Protégé” programa sukūrus ontologijos failus, automatizuotas įrankis sugeba teisingai sukurti klases, klasių parametrus, jungiamuosius objektus.

ONTOLOGIJOS FAILO SUKŪRIMAS

1. Atverti programą „Protégé”
2. Sukurti naują ontologijos failą
3. Sukurti klases
4. Sukurti duomenų savybes
5. Duomenų savybes priskirti klasėms

AĮ klasių nuskaitymas

1. Paleisti AĮ
2. Nurodyti ontologijos failą
3. Nurodyti projekto katalogo kelią
4. Vykdyti klasių sukūrimo operaciją
5. Palyginti sukurtų klasių pavadinimus ir kiekį su ontologijos failo turiniu
6. Uždaryti AĮ

AĮ duomenų savybių nuskaitymas

1. Paleisti AĮ
2. Nurodyti ontologijos failą
3. Nurodyti projekto katalogo kelią
4. Vykdyti duomenų savybių sukūrimo operaciją
5. Palyginti sukurtų duomenų savybių pavadinimus ir skaičių su ontologijos failo turiniu
6. Uždaryti AĮ

AĮ pernešančių objektų sukūrimas

1. Paleisti AĮ
2. Nurodyti ontologijos failą
3. Nurodyti projekto katalogo kelią
4. Vykdyti pernešančių objektų sukūrimo operaciją
5. Peržiūrėti pernešančių objektų turinius, palyginti su laukiamais rezultatais
6. Uždaryti AĮ

AI jungiamųjų objektų sukūrimas

7. Paleisti AI
8. Nurodyti ontologijos failą
9. Nurodyti projekto katalogo kelią
10. Vykdyti jungiamųjų objektų sukūrimo operaciją
11. Peržiūrėti jungiamųjų objektų turinius, palyginti su laukiamais rezultatais
12. Uždaryti AI

4. Žinių portalo projektas

„Ontologija grindžiamo kūrimo metodo taikymas interneto portalui kurti“ yra žinių portalo kūrimo procesas taikant dalykinės srities modelį, kuris aprašoma kaip dalykinė sritis susijusi su interneto portalo funkcionalumu. Kiekvienas projekto variantas turi konkrečių privalumų ir problemų, nes skirtinguose projektuose taikomi skirtingi būdai bendradarbiavimui tarp srities susijusios ir nuo dalykinės srities nepriklausomo funkcionalumo integravimui sukurti.

DDD nėra kaip sprendimas pagerinti architektūrinę dalį, tačiau naudojant DDD ir ontologija grindžiamą dalykinę sritį siekiama suteikti priemones, padedančias palengvinti priežiūrą, paspartinti projekto vystymą ir pridėti lankstumą ir atsparumą reikalavimų pasikeitimuose.

Pagal analizės dalyje pasirinktą modelį, kuriamas žinių portalas, kurio pagrindinės funkcijos:

- Naujienu importavimas
- Vienetinių naujienu pateikimas
- Naujienu archyvavimas
- Naujienu paieška
- Reklamos pateikimas
- Naujienu komentarų rašymas, administravimas
- Interaktyvių paprogramių pateikimas
- Bendravimas su išoriniais servisais

4.1. Eksperimentinės sistemos analizė

4.1.1. Vartotojų aibė, tipai ir savybės

Vartotojo kategorija: Vartotojas svečias

Vartotojo sprendžiami uždaviniai:

- Naujienų portalo naujienų skaitymas
- Naujienų komentavimas
- Archyvo skaitymas

Patirtis dalykinėje srityje: patirtis nereikalaujama

Patirtis informacinėse technologijose: Nepatyręs

Papildomos vartotojo charakteristikos: Visos amžiaus grupės

Vartotojo prioritetai: Svarbus vartotojas

Vartotojo kategorija: Registruotas vartotojas

Vartotojo sprendžiami uždaviniai:

- Naujienų portalo naujienų skaitymas
- Naujienų komentavimas
- Archyvo skaitymas
- Apmokestintų naujienų skaitymas
- Apmokestintų archyvo naujienų skaitymas
- Naršymo istorijos peržiūra, portalo naujienų išdėstymas pagal svarbumą

Patirtis dalykinėje srityje: patirtis nereikalaujama

Patirtis informacinėse technologijose: Nepatyręs

Papildomos vartotojo charakteristikos: Nusiteikęs naujovėms, greitai perprantantis vaizdinę informaciją.

Vartotojo prioritetai: Svarbiausias vartotojas

4.1.2. Vartotojų tikslai ir problemos

Vartotojų tikslai:

Vartotojas svečias:

Portalas turi palengvinti informacijos surinkimą, naujienų įsisavinimą

Registruotas vartotojas:

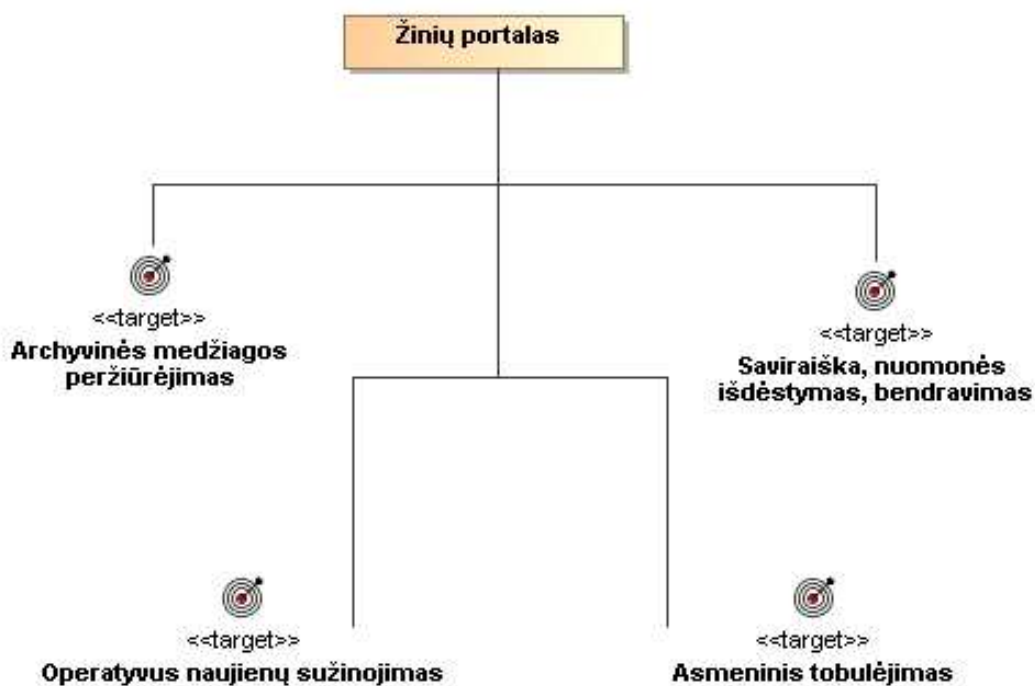
Portalas turi palengvinti informacijos surinkimą, naujienų įsisavinimą

Portalas užtikrina vartotojų konfigūracijos išsaugojimą, didesnę portalo funkcionalumą

4.2. Reikalavimų specifikacija

4.2.1. Tikslų modelis

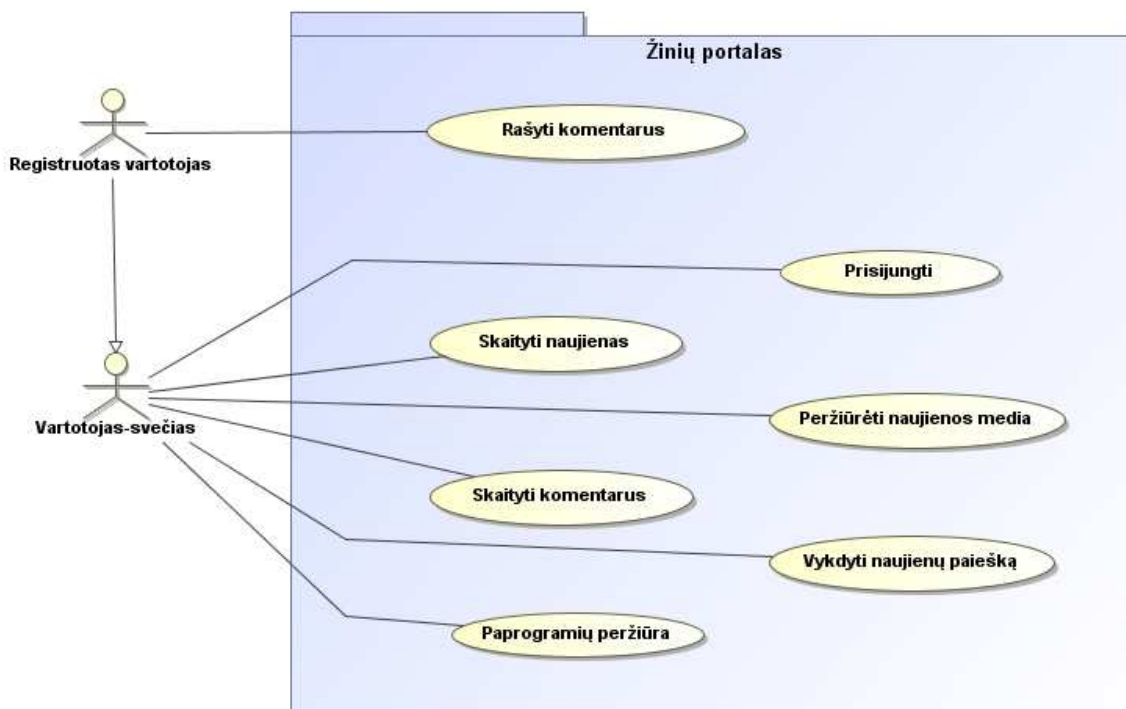
Pagrindiniai kuriamos informacinės sistemos tikslai pavaizduoti 9 pav.



9 pav. Žinių portalų tikslai

4.2.2. Panaudojimo atvejų modelis

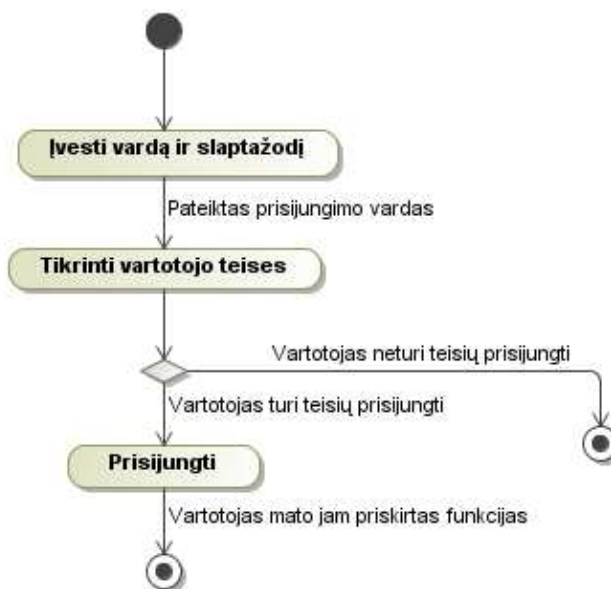
Naujienų portalų detalizuota panaudojimo atvejų diagrama pateiktas 10 pav.



10 pav. Žinių portalų kompiuterizuojamų panaudojimo atvejų diagrama

2 lentelė. PA „Prisijungti“ specifikacija

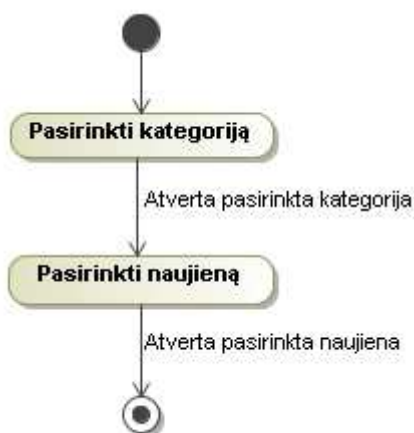
PA „Prisijungti“	
Tikslas. Registruotiems vartotojams prisijungti prie sistemos ir naudotis suteiktomis papildomomis valdymo funkcijomis	
Aprašymas. Šis PA apima registruotų vartotojų prisijungimą	
Prieš sąlyga	Atidarytas prisijungimo langas
Aktorius	Registruotas vartotojas
Sužadinimo sąlyga	Vartotojas nori naudotis papildomomis funkcijomis
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas įveda prisijungimo vardą	Sistema pateikia papildomas valdymo funkcijas
2. Vartotojas įveda slaptažodį	Sistema rodo papildomas valdymo funkcijas ir naujienų turinį
3. Vartotojas baigia PA	
Po sąlyga:	
Alternatyvūs scenarijai	
1 alternatyvus scenarijus	
Prisijungimo duomenys neteisingi	



11 pav. „Prisijungti“ veiklos diagrama

3 lentelė. PA „Skaityti naujienas“ specifikacija

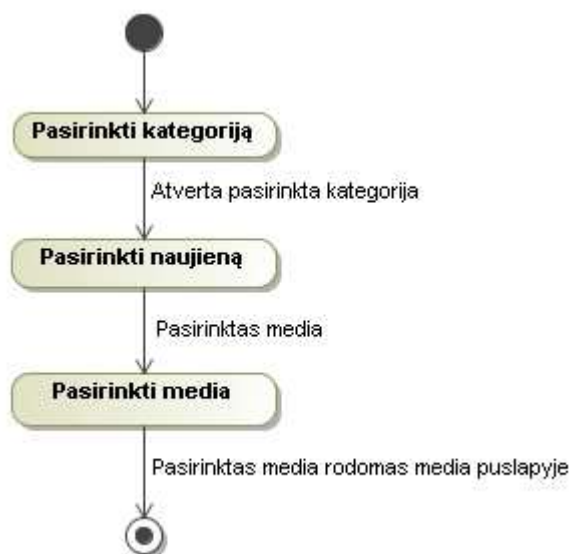
PA „Skaityti naujienas“	
Tikslas. Vartotojams svečiams ir registruotiems vartotojams peržiūrėti naujienas, susipažinti su aktualia informacija.	
Aprašymas. Šis PA apima naujienų peržiūrą.	
Prieš sąlyga	Atidarytas pagrindinis puslapis
Aktorius	Vartotojas svečias, registruotas vartotojas
Sužadinimo sąlyga	Vartotojas nori peržiūrėti naujienas
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka kategoriją	Sistema pateikia naujienų sąrašą
2. Vartotojas pasirenka naujieną peržiūrai	Sistema rodo naujienos turinį
Po sąlyga:	
Alternatyvūs scenarijai	
1 alternatyvus scenarijus	
Kategorijoje naujienų nėra	



12 pav. „Skaityti naujienas“ veiklos diagrama

4 lentelė. PA „Peržiūrėti naujienos media“ specifikacija

PA „Peržiūrėti naujienos media“	
Tikslas. Vartotojams peržiūrėti naujienos media informaciją (filmuota medžiaga, garsas, nuotraukos, animacija)	
Aprašymas. Šis PA apima naujienų media peržiūrą.	
Prieš sąlyga	Importuotos naujienos
Aktorius	Vartotojas svečias, registruotas vartotojas
Sužadinimo sąlyga	Vartotojas nori peržiūrėti naujienas
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka kategoriją	Sistema pateikia naujienų sąrašą
2. Vartotojas pasirenka naujieną peržiūrai	Sistema rodo naujienos turinį
3. Vartotojas pasirenka media	Sistema rodo pasirinktą media „Media“ puslapyje
Po sąlyga:	
Alternatyvūs scenarijai	
1 alternatyvus scenarijus	
Naujienoje media nėra	



13 pav. „Naujienos media peržiūra“ veiklos diagrama

5 lentelė. PA „Skaityti komentarus“ specifikacija

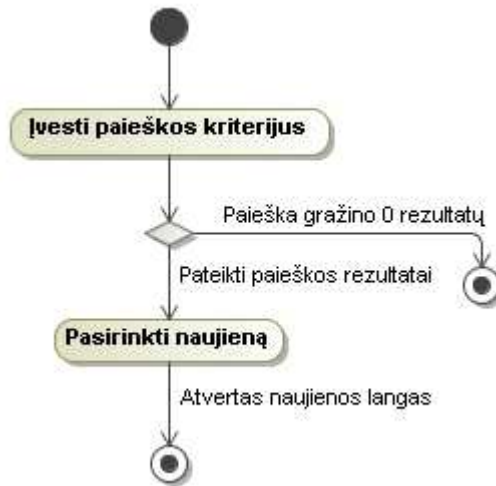
PA „Skaityti komentarus“	
Tikslas. Vartotojams svečiams ir registruotiems vartotojams peržiūrėti naujienos komentarus.	
Aprašymas. Šis PA apima naujienos komentarų peržiūrą.	
Prieš sąlyga	Importuotos naujienos, komentuoti straipsniai
Aktorius	Vartotojas svečias, registruotas vartotojas
Sužadinimo sąlyga	Vartotojas nori peržiūrėti naujienos komentarus
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka kategoriją	Sistema pateikia naujienų sąrašą
2. Vartotojas pasirenka naujieną peržiūrai	Sistema rodo naujienos turinį
3. Vartotojas peržiūri komentarus	
Po sąlyga:	
Alternatyvūs scenarijai	
1 alternatyvus scenarijus	



14 pav. „Skaityti komentarus“ veiklos diagrama

6 lentelė. „Vykdėti naujienų paiešką“ specifikacija

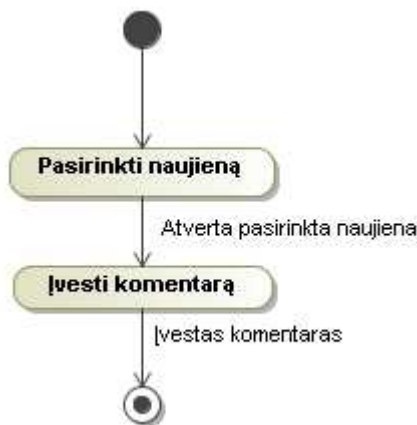
PA „Vykdėti naujienų paiešką“	
Tikslas. Vartotojams surasti aktualią naujieną, naudojantis paieška.	
Aprašymas. Šis PA apima naujienų paiešką.	
Prieš sąlyga	Importuotos naujienos
Aktorius	Vartotojas svečias, registruotas vartotojas
Sužadinimo sąlyga	Vartotojas nori peržiūrėti naujienas
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas įveda paieškos kriterijus	Sistema pateikia paieškos rezultatus
2. Vartotojas pasirenka naujieną	Sistema atidaro naujos langą
Po sąlyga:	
Alternatyvūs scenarijai	
1 alternatyvus scenarijus	



15 pav., „Vykdėti naujienų paiešką“ veiklos diagrama

7 lentelė. „Rašyti komentarus“ specifikacija

PA „Rašyti komentarus“	
Tikslas. Registruotiems vartotojams išreikšti savo nuomonę apie naujienas.	
Aprašymas. Šis PA apima komentarų rašymą.	
Prieš sąlyga	Importuotos naujienos
Aktorius	Registruotas vartotojas
Sužadinimo sąlyga	Vartotojas nori peržiūrėti naujienas
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka naujieną	Sistema rodo naujienos turinį
2. Vartotojas parašo komentarą	Sistema išsaugo komentarą patikrai
Po sąlyga:	
Alternatyvūs scenarijai	
1 alternatyvus scenarijus	



16 pav., „Rašyti komentarus“ veiklos diagrama

8 lentelė. PA „Paprogramių peržiūra“ specifikacija

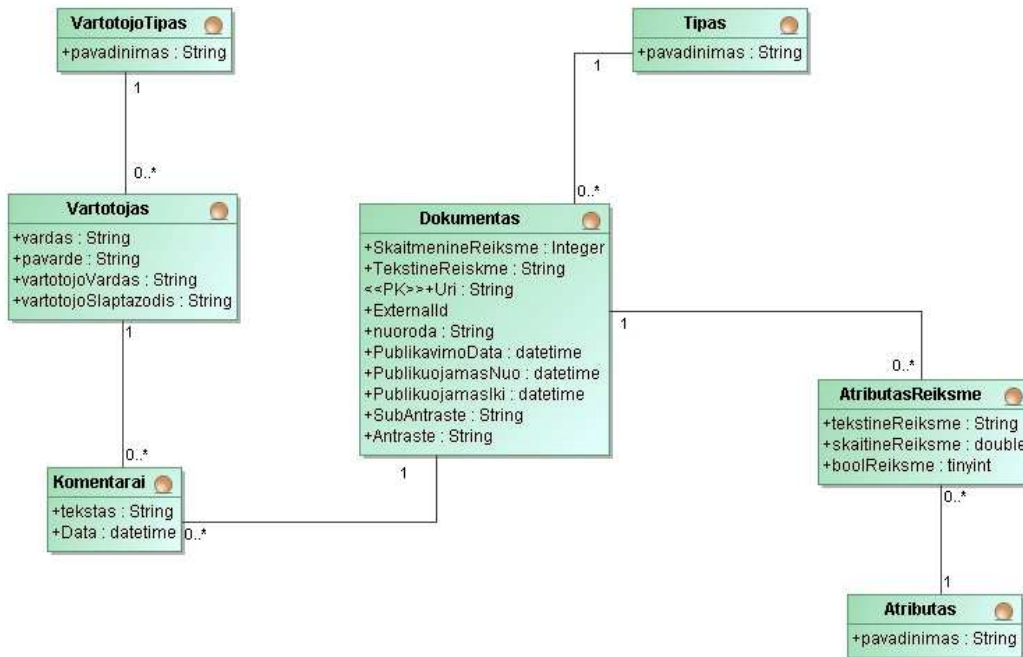
PA „Paprogramių peržiūra“	
Tikslas. Vartotojams svečiams ir registruotiems vartotojams peržiūrėti interaktyvias paprogrames, naudotis papildomomis funkcijomis.	
Aprašymas. Šis PA apima paprogramių peržiūrą.	
Prieš sąlyga	Integruotos paprogramės
Aktorius	Vartotojas svečias, registruotas vartotojas
Sužadinimo sąlyga	Vartotojas nori peržiūrėti naujienas
Susiję panaudojimo atvejai	Išplečia PA
	Apima PA
	Specializuoja PA
Pagrindinis įvykių srautas	Sistemos reakcija ir sprendimai
1. Vartotojas pasirenka kategoriją	Sistema pateikia naujienų sąrašą
2. Vartotojas pasirenka paprogramę	Sistema atveria paprogramės langą
3. Vykdyti paprogramės funkcijas	Sistema įvykdo paprogramės funkcijas
Po sąlyga:	
Alternatyvūs scenarijai	
1 alternatyvus scenarijus	



17 pav. „Paprogramių peržiūra“ veiklos diagrama

4.3. Dalykinės srities modelis

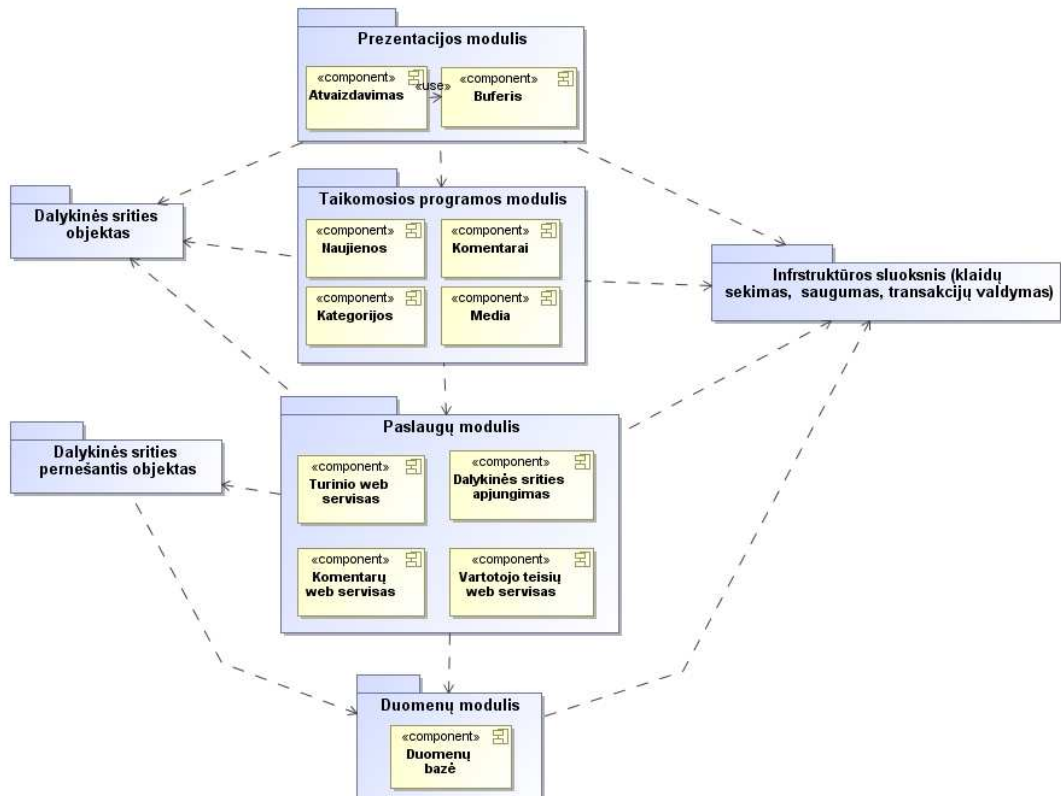
Dalykinės srities esybių klasių modelis pateiktas 18 pav. Šios klasės yra skirtos duomenims saugoti, jos neturi metodų.



18 pav. Naujių portalų dalykinės srities modelis

4.4. Sistemos architektūros projektas

4.4.1. Sistemos loginė architektūra

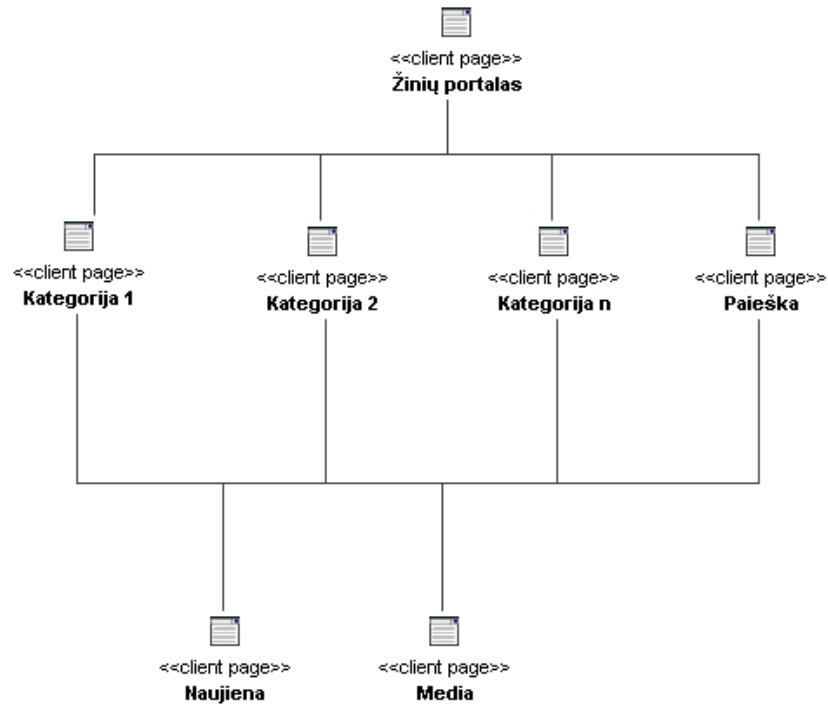


19 pav. Sistemos loginė architektūra

Šią sistemos loginę architektūrą (19 pav.) detalizuojant ir panaudojant projekto klases, gaunama detalizuota sistemos architektūra, joje papildomai įtraukiamos sistemos paslaugos.

4.4.2. Vartotojo paslaugos (sąsaja, sąsajos navigavimo planas)

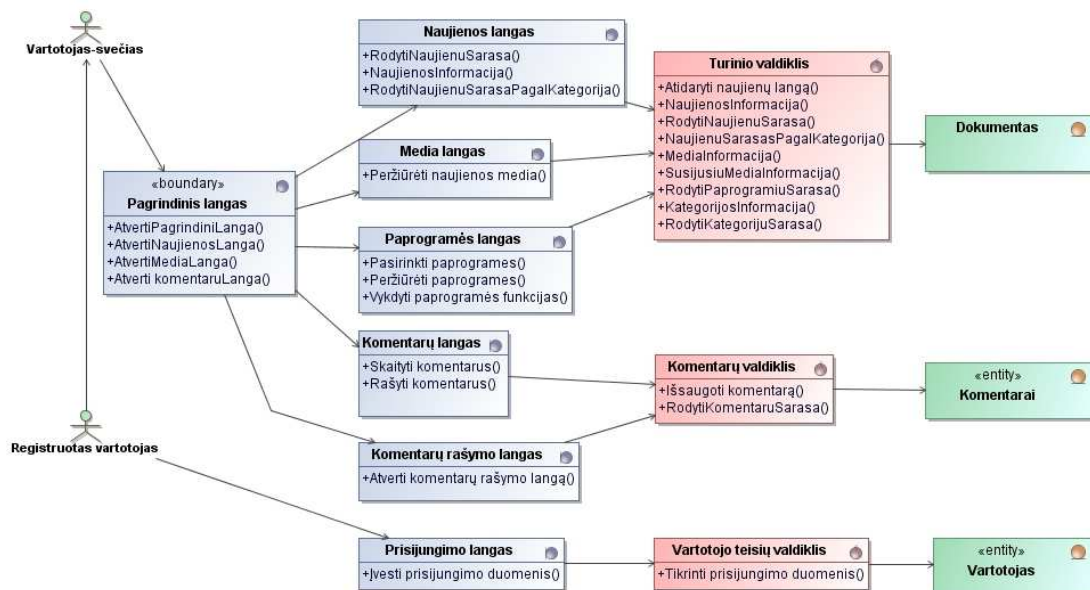
Navigavimo planas pateiktas 20 pav.



20 pav. Navigavimo planas

4.5. Detalus projektas

Sistemos klasių diagrama su operacijomis (21 pav.):



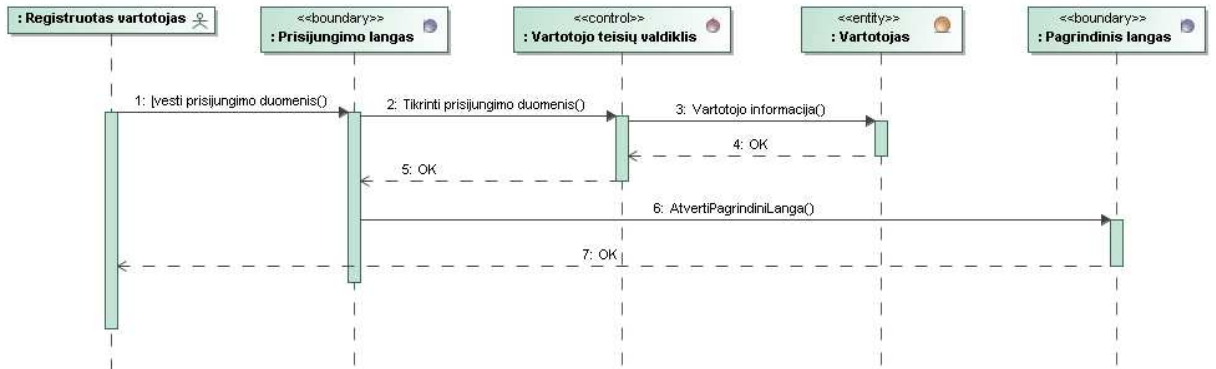
21 pav. Panaudojimo atvejų diagrama su operacijomis

Diagramoje dauguma valdiklių susisiekia su „Dokumentas“ esybe. Taip yra dėl to, kad duomenų bazė suprojektuota remiantis OWL kalba, kai kiekvienas artefaktas turi aibę atributų.

4.6. Sistemos elgsenos modelis

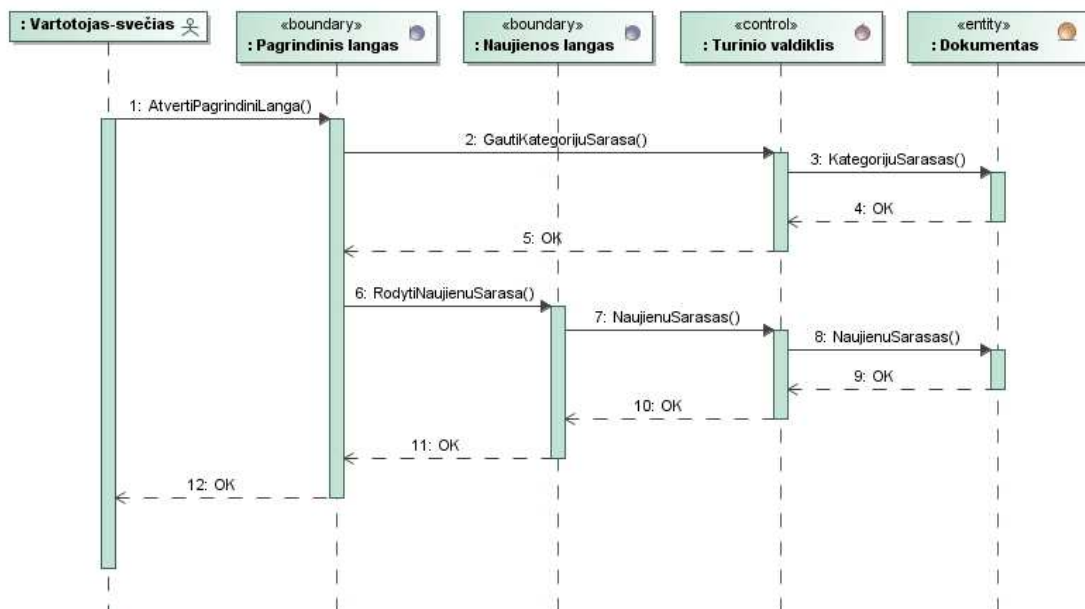
Sekų diagramos pateikiamos 22 – 27 pav.

PA „Prisijungti“ sekos diagrama (22 pav.)



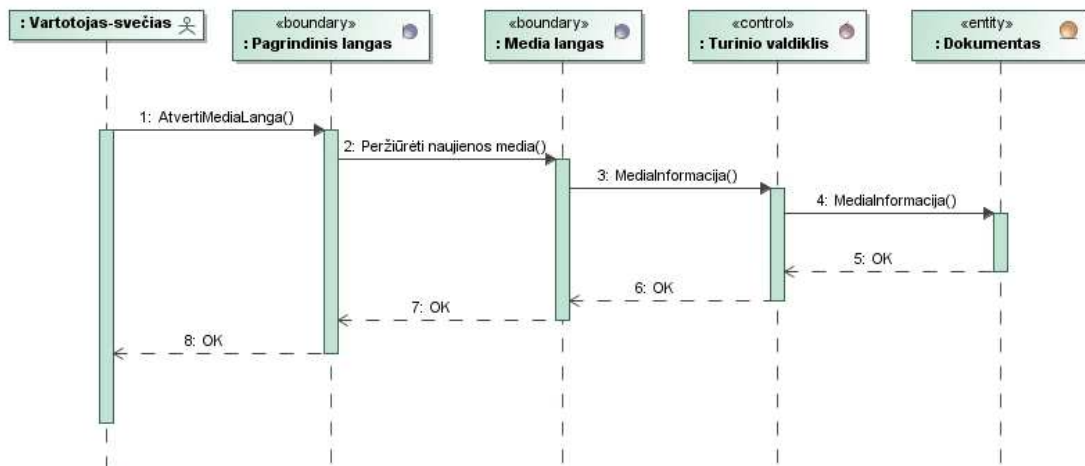
22 pav. „Prisijungti“ sekos diagrama

PA „Skaityti naujienas“ sekos diagrama (23 pav.)



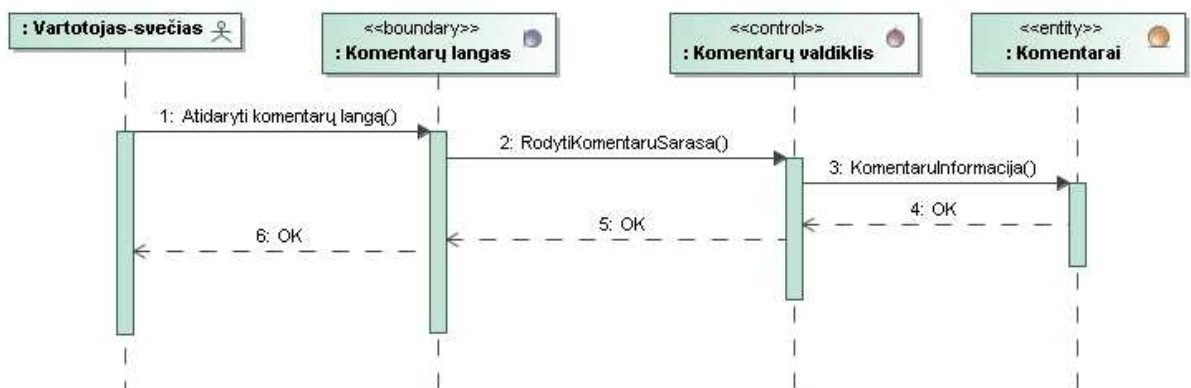
23 pav. „Skaityti naujienas“ sekos diagrama

PA „Peržiūrėti naujienos media“ sekos diagrama (24 pav.)



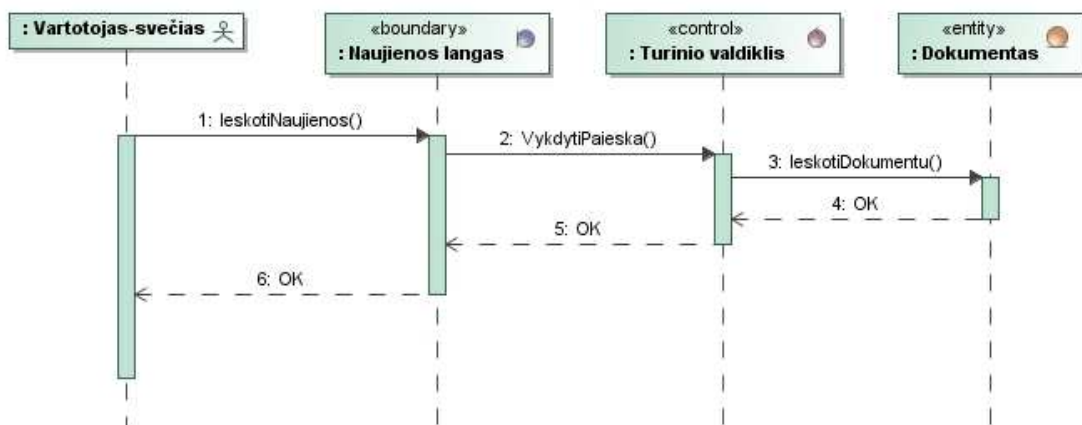
24 pav. „Peržiūrėti naujienos media“ sekos diagrama

PA „Skaityti komentarus“ sekos diagrama (25 pav.)



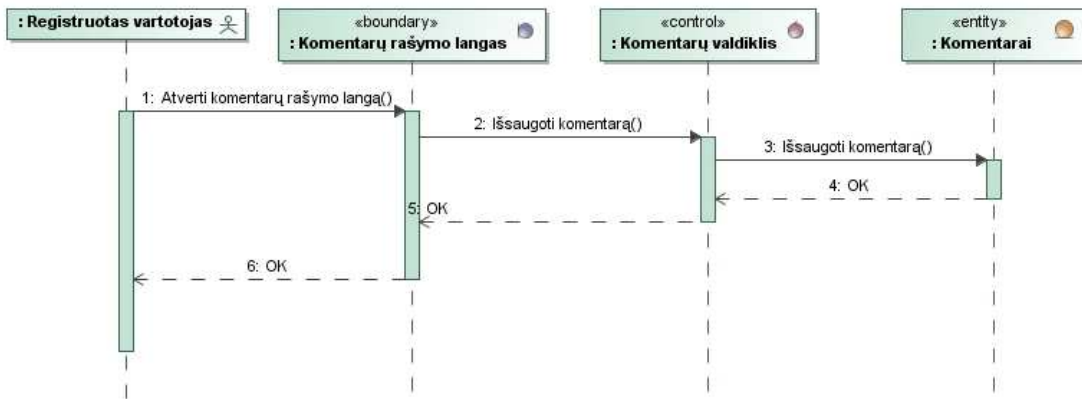
25 pav. „Skaityti komentarus“ sekos diagrama

PA „Vykdėti naujienų paiešką“ sekos diagrama (26 pav.)



26 pav. „Vykdėti naujienų paiešką“ sekos diagrama

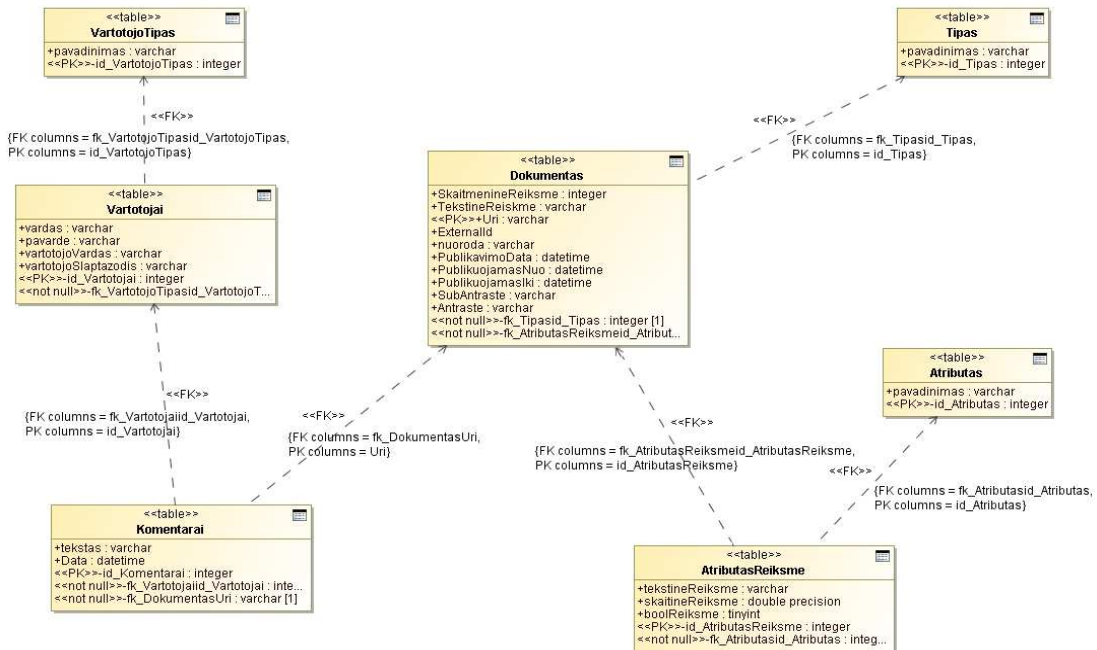
PA „Rašyti komentarus“ sekos diagrama (27 pav.)



27 pav. „Rašyti komentarus“ sekos diagrama

4.7. Duomenų bazės schema

Duomenų bazės schema, naudota realizuoti informacinę sistemą (28 pav.)

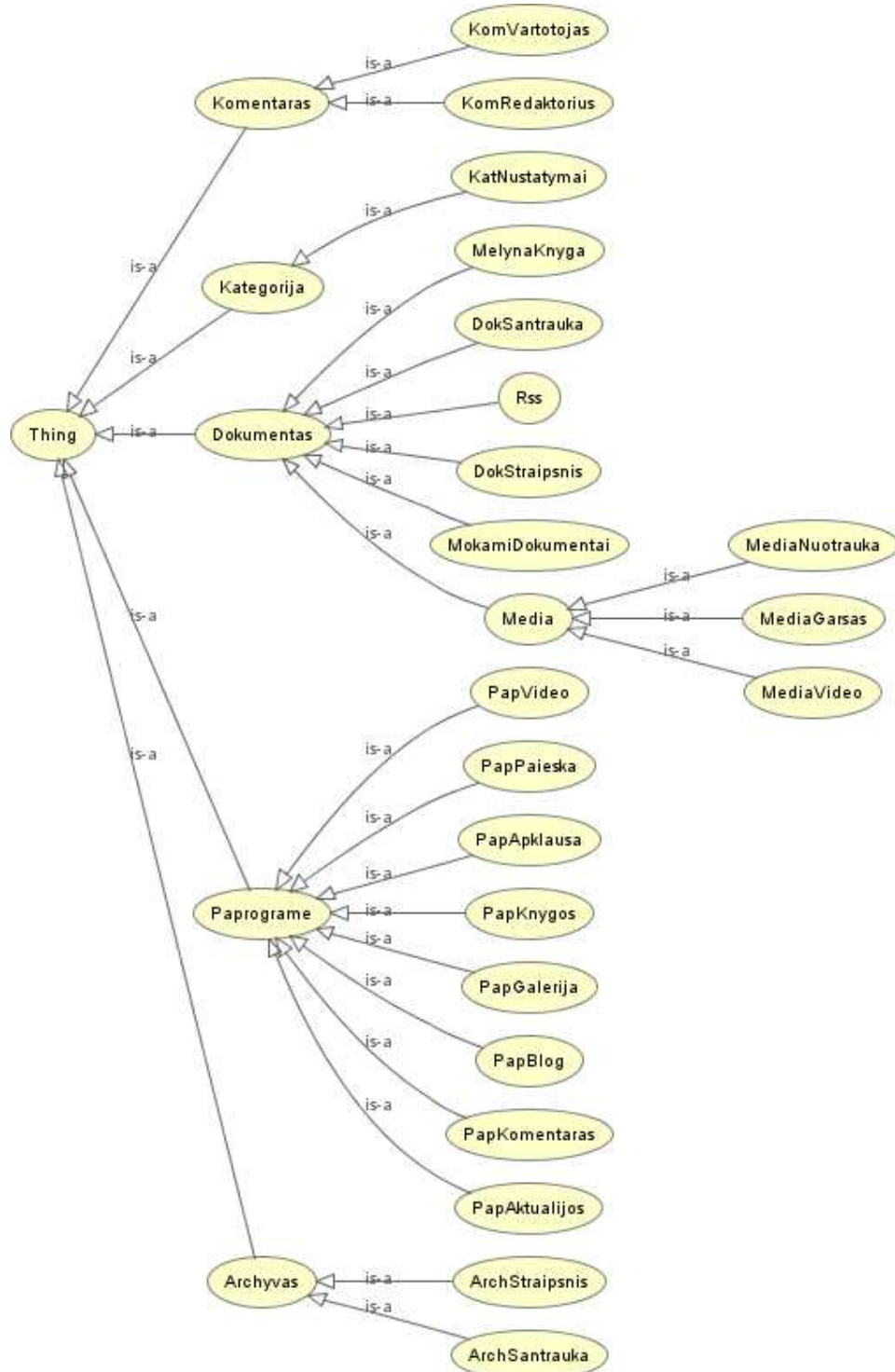


28 pav. Duomenų bazės schema

Duomenų bazės schema skiriasi nuo pavyzdinės schemos, aprašytos punkte „Ontologijos transformavimas į duomenų bazės schemą“. Dėl didelio duomenų kiekio (> 500 000) įrašų DB lentelėje „Dokumentas“ ir gerokai daugiau įrašų DB lentelėje „AtributasReiksme“ MS SQL duomenų bazėje užklausos pradėjo užtrukti ilgiau nei nustatytas laiko limitas, reikalingas įvykdyti užklausą, teko pagrindinius atributus iškelti į lentelę „Dokumentas“. Pavyzdinė duomenų bazės schema galėtų būti pasiekama labiau

optimizuojant užklausas, bet įtraukiant tik pagrindinius atributus į duomenų bazės schemos lenteles, labiau išnaudojamos duomenų bazės privalumai.

Naudojantis „Protege“ programa sukurtas ontologijos aprašas, aprašantis žinių portalo dalykinę sritį (29 pav.).



29 pav. Naujienų portalo OWL ontologija

Naudojantis automatizuotu įrankiu sukuriami pagrindiniai architektūriniai failai, įrankio langai yra primityvūs ir pateikia tik tekstinę informaciją apie procesų įgyvendinimus (30-32 pav.).

```
CLASS STRUCTURE
DokSantrauka : Dokumentas
  Subantraste : string
  Antraste : string
-----
Dokumentas :
  SukurimoData : dateTime
  GaliojaIki : dateTime
  ModifikavimoData : dateTime
  GaliojaNuo : dateTime
-----
DokStraipsnis : Dokumentas
  Antraste : string
  Autorius : string
  Tekstas : string
-----
```

30 pav. Klasių hierarchijos sukūrimas ir vaizdavimas

```
-----
Create DOs
-----
Project Directory: D:/NaujienuPortalas
DOs Directory: D:/NaujienuPortalas/DO
The directory is already created.
using System;

public class DokSantrauka : Dokumentas {
    public string Subantraste {get; set;}
    public string Antraste {get; set;}
}

using System;

public class Dokumentas {
    public DateTime SukurimoData {get; set;}
    public DateTime GaliojaIki {get; set;}
    public DateTime ModifikavimoData {get; set;}
    public DateTime GaliojaNuo {get; set;}
}

using System;

public class ArchStraipsnis : Archyvas {
    public boolean ArMokamas {get; set;}
    public string Tekstas {get; set;}
}
-----
```

31 pav. Dalykinės srities objektų sukūrimas ir vaizdavimas

```

public class ArchyvasMapper : BaseMapper {
    public Archyvas GetMappingValuesFromDTO(string[] predicateValues) {
        Archyvas doObj = new Archyvas();

        doObj.SukurimoData = GetValuedateTime(predicateValues, "SukurimoData", "date
Time");

        return doObj;
    }

    public string[] ResultPredicates() {
        string[] result = null;

        result = new string[] {
            "SukurimoData",
        };
        return result;
    }
}

public class DokumentasMapper : BaseMapper {
    public Dokumentas GetMappingValuesFromDTO(string[] predicateValues) {
        Dokumentas doObj = new Dokumentas();

        doObj.SukurimoData = GetValuedateTime(predicateValues, "SukurimoData", "date
Time");
        doObj.Galiojalki = GetValuedateTime(predicateValues, "Galiojalki", "dateTime
");
        doObj.ModifikavimoData = GetValuedateTime(predicateValues, "ModifikavimoData
", "dateTime");
        doObj.Galiojanauo = GetValuedateTime(predicateValues, "Galiojanauo", "dateTime
");

        return doObj;
    }

    public string[] ResultPredicates() {
        string[] result = null;

        result = new string[] {
            "SukurimoData",
            "Galiojalki",
            "ModifikavimoData",
            "Galiojanauo",
        };
        return result;
    }
}

public class ArchStraipsnisMapper : BaseMapper {
    public ArchStraipsnis GetMappingValuesFromDTO(string[] predicateValues) {
        ArchStraipsnis doObj = new ArchStraipsnis();

        new ArchyvasMapper().GetMappingValuesFromDTO(predicateValues);
        doObj.ArMokamas = GetValueboolean(predicateValues, "ArMokamas", "boolean");

        doObj.Tekstas = GetValuestring(predicateValues, "Tekstas", "string");

        return doObj;
    }

    public string[] ResultPredicates() {
        string[] result = null;

        result = new string[] {
            "SukurimoData",
            "ArMokamas",
            "Tekstas",
        };
        return result;
    }
}

```

32 pav. Jungiamųjų objektų sukūrimas ir vaizdavimas

Kuriant žinių portalą, pakartotinai panaudoti komponentų nebuvo galima, kadangi tai buvo pirmasis internetinis projektas pagal DDD metodą, tad viskas sukurta būtent šiam projektui. Žinių portalo pagrindinis puslapis pateiktas 33 pav. Pagrindiniam puslapyje matomos kategorijos ir aktualiausias naujienos, dešinėje pusėje matyti paprogramės. Paprogramės rankiniu būdu priskiriamos kiekvienai kategorijai.

FAZJOB.NET FAZFINANCE.NET Märkte : F.A.Z.-Archiv : e-paper : Abo : Blogs : Services Mein FAZ.NET Login Suche

Frankfurter Allgemeine **FAZ.NET** Home Politik : Wirtschaft : Feuilleton : Sport : Gesellschaft : Finanzen : 24. Januar 2010 : Reise : Wissen : Auto : Computer : Beruf & Chance : Kunstmarkt : Immobilien : Rhein-Main

Haiti Chaos und Verwüstung

Elf Tage nach dem Beben Rettung aus den Trümmern
Bild für Bild Elf Tage nach dem Beben gelang es einem französischen Team einen 24-Jährigen aus den Trümmern eines Hotels zu retten. Offiziell hat die Regierung die Suche nach Überlebenden bereits für beendet erklärt. Die Zahl der Toten erhöhte sich auf mehr als 112.000. >

Frankreichs Rolle in Haiti
Wo die Republik ihre Ideale verriet
 Frankreich engagiert sich demonstrativ für den Wiederaufbau Haitis - auch als Sühne für die Verbrechen der Kolonialzeit. Ein gutes Jahrhundert bis zur von den einstigen Sklaven blutig erkämpften Unabhängigkeit 1804 hat Frankreich in Saint-Domingue, dem heutigen Haiti, geherrscht.
 Von Michaela Wisegal, Paris >

Adoption: „Haiti steckt ihm im Blut“
 Nach dem Beben ziehen viele Haitianer aufs Land
 Video: 24 Jahre alter Haitianer aus Trümmern gerettet >

Der Kommentar Ohne Lafontaine
 Von Mechthild Küpper
 Ohne Oskar Lafontaine wird sichtbar, was der Starkult angerichtet hat: Mit ihrem SED-Erbe hat sich die Linkspartei nicht ernsthaft auseinandergesetzt. >

Videos Serge Gainsbourg
Der raue Barde mit den schönsten Frauen

Martin Winterkorn
Der Auto-Star
 Der VW-Chef übernimmt Porsche, steigt bei Suzuki ein und will Toyota als weltgrößter Autohersteller ablösen. Wenn das mal gutgeht. Doch Anzeichen von Müdigkeit lässt Winterkorn freilich

FAZ.NET-RSS-Services

33 pav. Pagrindinis portalo langas

Pasirinkus naujieną, atidaromas naujienos pilnas aprašymas (34 pav.). Puslapyje išskiriami du stulpeliai. Kairiniame tekstas, dešiniame paprogramės.

Aktuell > Gesellschaft > **Katastrophen**

Elf Tage nach dem Beben Artikel-Services

Rettung aus den Trümmern

24. Januar 2010 Das Erdbeben in Haiti hat mindesten 112.000 Menschen das Leben gekostet und ist damit eine der zerstörerischsten Naturkatastrophen der vergangenen 100 Jahre. Die Suche nach Überlebenden wurde zwar offiziell am Freitag eingestellt - und doch gab es danach noch ein kleines Wunder: Ein französisches Team rettete am Samstag, elf Tage nach der Katastrophe, einen 24-Jährigen aus den Trümmern eines Hotels. Insgesamt konnten die Rettungskräfte 133 Überlebende aus zerstörten Gebäuden ziehen.

In der Umgebung der Hauptstadt Port-au-Prince seien noch immer etwa 609.000 Menschen ohne jegliches Obdach, berichtete die UN-Organisation für Migration (IOM) in Genf unter Berufung auf die Regierung Haitis. Es würden dringend Zelte benötigt, sagte IOM-Sprecher Jean-Philippe Chauzy am Sonntag in Genf. Andererseits verließen immer mehr Bewohner die Hauptstadt: Mehr als 130.000 hätten bereits das Angebot der Regierung angenommen und seien in den Norden und den Südwesten gegangen.

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 21

Gesellschaft RSS
 Menschen > Katastrophen
 Umwelt > Kriminalität
 Mode > Gesundheit
 Jung

Anzeige
 Google Anzeigen
Emissionshandel: A bis Z
 Das Handbuch als PDF-Dokument Alles was Sie hierzu wissen müssen!
www.clando.com/emissionshandel
Nie wieder XXL Mode:
 Bis 16 kg in 4 Wochen, schnell und schonend abnehmen. Sofort Erfolge!
abnehmen.slim-line.info

Videos Rubriken Blättern
 Elf Tage nach dem Beben
24 Jahre alter Haitianer aus Trümmern gerettet

34 pav. Naujienos pilnas aprašymas

Naujiena, prie kurios prisegta paveikslėlių, video arba garso įrašų, turi media puslapį (35 pav.)

Aktuell > Sport > Wintersport

← zurück 1 | 2 2 3 Zum Beitrag >



Leer, weit, schön: Zweieinhalbtausend Kilometer Loipen gibt es rund um die Olympiastadt Lillehammer, und nichts stört die Harmonie von Schnee und Mensch.

Alex Westhoff

Drucken Versenden Speichern Lesezeichen

35 pav. Naujienos media puslapis

Registruoti vartotojai gali komentuoti naujieną (36 pav.)

Mein FAZ.NET > **Lesermeinungen**

Krankenkasse	Zahl der Versicherten	Zahl der Beitragszahler
DAK	1,4 Mio	1,4 Mio
VKM Allianz	2 Mio	1,8 Mio
BKK Gesundheit	1,4 Mio	1,4 Mio
Deutsche Bkk	1 Mio	750 000
Kneller BKK	300 000	240 000
BKK für Profisportler	120 000	120 000
VVa BKK	110 000	110 000
BKK Westfalen	100 000	100 000

**DAK gibt Startschuss
Acht Kassen verlangen höhere Beiträge**

Mit der DAK hat erstmals eine große gesetzliche Krankenkasse Zusatzbeiträge für ihre Versicherten angekündigt. Ab Februar soll der Beitrag von acht Euro pauschal von den Mitgliedern erhoben werden. Sieben weitere gesetzliche Krankenkassen wollen folgen. Experten glauben, dass noch sehr viel mehr Kassen nachziehen werden. >

Betreff (max. 80 Zeichen)

 Noch 80 Zeichen frei

Ihr Kommentar (max. 1250 Zeichen)

 Noch 1250 Zeichen frei

> Richtlinien für Lesermeinungen Absenden GO

Suche in Lesermeinungen

Suche in Beitrag Lesername oder Login

Datum bis

Suche GO

Anzeige

PRIVATE Krankenkasse 59€
TOP-Testsieger Private Krankenkasse ab nur 59,-Euro/mtl.

SIXT-Autos ab €9,99/Tag!
Günstige SIXT Autos in über 80 Ländern mieten. Online extra günstig, ab € 9,99/Tag!

Time for Change
Investieren Sie in einen neuen Markt – mit VoIP und sipkom!

Lesermeinungen Ressorts Blättern

23:36 Sicherheit durch SMS-Information ?

36 pav. Komentarų rašymas naujienai

Paieška vykdoma įvedus raktažodį į paieškos lauką. Paieškos rezultatai rodomi 37 pav.

Aktuell > **Suche**

Artikel (4277) Archiv (8770) Rezensionen (122) Lesermeinungen (1015)

1 - 10 von 4277 Suchergebnissen in FAZ.NET

1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |  

Der Fahrtbericht: Mercedes-Benz E 350 CDI Der kantige Leitwolf in der Auto-Elite 

Der Kollege hat sich einen Mercedes gekauft. Sofort wird darüber gesprochen, als hätte er sechs Richtige im Lotto gehabt. Dass es nur eine gebrauchte C-Klasse für weniger als 8000 Euro war, spielt keine Rolle. >

F.A.Z.

25. Januar 2010, 01:26

[Autographen](#)

Liebesworte eines Gangsters

Jacques Mesrine war jahrelang Staatsfeind Nr. 1 in Frankreich. Aus dem Gefängnis schrieb er seiner großen Liebe. Jetzt werden seine Briefe versteigert. >

F.A.Z.

25. Januar 2010, 05:55

[Martin Winterkorn](#)

Der Auto-Star

Der VW-Chef übernimmt Porsche, steigt bei Suzuki ein und will Toyota als weltgrößter Autohersteller ablösen. Wenn das mal gutgeht. Doch Anzeichen von Müdigkeit lässt Winterkorn freilich nicht erkennen. >

F.A.S.


24. Januar 2010, 11:40

Aktuelle Sucheinstellungen

Suchbegriffe
Suchbereich
Themengebiete
Suchzeitraum

[Suche ändern](#) 

 Beiträge mit vorangestelltem Schlüssel-Symbol sind für eingeloggte F.A.Z.-Zeitungsubonnenen frei zugänglich.

 Auch wenn Sie die F.A.Z. nicht abonniert haben, können Sie diesen Beitrag zum ausgewiesenen Preis für 24 Stunden nutzen.

[Suchhilfe](#) >

FAZ.NET-Ticker

 [Ressorts](#)  [Blättern](#)

Traunstein: Felsbrocken stürzt auf Wohnhaus ...

Zweite Fußball-Bundesliga: Duisburg untermauert Aufstiegsambitionen

DFB-Leistungstest: Van Gaal bleibt ein einsamer Kritiker

37 pav. Paieškos rezultatai

Paprogramė „FAZ.NET-Ticker“ (38 pav.) ir jos peržiūrėjimas (39 pav.).

FAZ.NET-Ticker  [Ressorts](#)  [Blättern](#)

Traunstein: Felsbrocken stürzt auf Wohnhaus ...

Zweite Fußball-Bundesliga: Duisburg untermauert Aufstiegsambitionen

DFB-Leistungstest: Van Gaal bleibt ein einsamer Kritiker

Top-Themen: Heute Woche Monat

38 pav. Paprogramė „FAZ.NET-Ticker“

■ Alle Beiträge □ Top-Themen

Ressorts gruppieren Ressort wählen

Heute
 Woche
 Monat

25.01.2010

1 2 3 4 5 6 7 8 9 10 11 12

22:32	Landkreis Traunstein Felsbrocken stürzt auf Wohnhaus >	Gesellschaft
22:24	Zweite Fußball-Bundesliga Duisburg untermauert Aufstiegsambitionen >	Sport
21:52	DFB-Leistungstest Van Gaal bleibt ein einsamer Kritiker Von Michael Horeni, Stuttgart >	Sport
21:29	Blog „Stützen der Gesellschaft“ Nach oben schlafen mit Tullia d’Aragona Von Don Alphonso >	Feuilleton
21:11	Früherer hessischer Minister Hoff wird Mitglied im Opel-Vorstand Von Ralf Euler, Wiesbaden >	Rhein-Main-Zeitung
19:56	Berlin Heidi Klum enthüllt ihre Wachfigur >	Video
19:51	Nach Lafontaine-Rückzug Linkspartei diskutiert über künftige Parteiführung >	Politik
19:28	Privat-öffentliche Partnerschaft 91 Schulen binnen fünf Jahren saniert Von Eberhard Schwarz >	Rhein-Main-Zeitung
19:21	Führungswechsel Bericht: Häusler soll Bayern LB leiten >	Wirtschaft
19:09	Flugzeugabsturz vor libanesischer Küste Kaum noch Hoffnung auf Überlebende Von Thomas Scheen und Hans-Christian Rißler >	Gesellschaft

Bilder des Tages Sportbilder des Tages

Blättern

F.A.Z.-Blogs

Aktuell Meistgelesen Themen

Zweiter Klasse
200 Minuten Verspätung
Von Nadine Bös

Zur Sicherheit
Substantielle Folgefragen
Von Stephan Löwenstein

Biopolitik
Englischer Autor wünscht
Suizid-Häuschen an den
Straßenecken
Von Oliver Tolmein

39 pav. Paprogramės „FAZ.NET-Ticker“ peržiūra

4.7.1. Portalo testavimas

Testuojant portalą tikimasi išsiaiškinti ar kokybiškas portalo veikimas. Ontologijų analizė šiame testavime nenaudojama kadangi prezentacijos lygis kuriamas programuotojų. Dalykinės srities sukūrimas/perkūrimas tik pagreitina portalo kūrimą.

Šiame testavime norima patikrinti ar portalas atitinka keliamus funkcinis reikalavimus.

NAUJIENOS PERŽIŪRA

1. Atsidaryti portalą
2. Naujienos atvaizduojamos
3. Uždaryti portalą

NAUJIENOS MEDIA PUSLAPIO PERŽIŪRA

1. Atsidaryti portalą
2. Atidaryti naujieną
3. Paspausti ant media elemento
4. Peržiūrėti media puslapį
5. Uždaryti portalą.

PRISIJUNGIMAS PRIE PORTALO

1. Atsidaryti portalą
2. Įvesti prisijungimo duomenis į prisijungimo duomenų laukelius

3. Prisijungti prie portalo
4. Peržiūrėti ar pateikti vartotojo duomenys ir prisijungimas įvykdytas sėkmingai
5. Uždaryti portalą.

KOMENTARO RAŠYMAS

1. Atsidaryti portalą
2. Prisijungti prie portalo
3. Atidaryti naujieną
4. Atverti komentavimo langą
5. Įvesti komentarą
6. Uždaryti portalą

KOMENTARO PERŽIŪRĖJIMAS

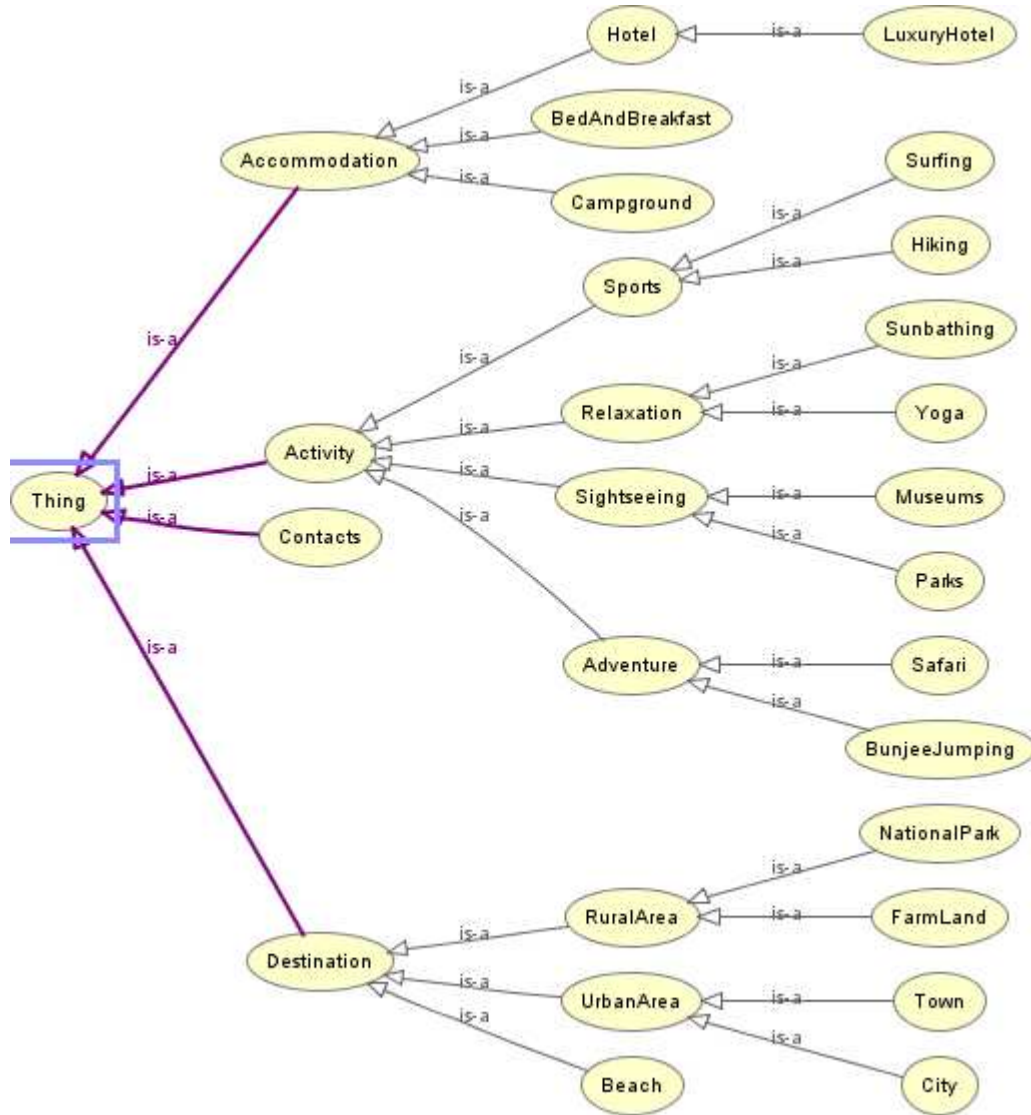
1. Atidaryti portalą
2. Atverti naujieną, kuri turi komentarą
3. Peržiūrėti komentarą
4. Uždaryti portalą

VYKDYTI NAUJIENŲ PAIEŠKĄ

1. Atidaryti portalą
2. Portalo paieškos lange įvesti frazę ir vykdyti paiešką
3. Paieškos rezultatuose ieškoti pateiktos frazės
4. Uždaryti portalą

5. Apyvėdinimo portalo projektas

Ontologija, nusakanti veiklos ir poilsio vietas padeda labiau suprasti OWL struktūrą. Sudarant ontologiją tik apie žinių sritį, per mažai skirtingų aibių su skirtingais parametrais, apyvėdinimo OWL struktūra sudėtingesnė už žinių portalo (40 pav.).



40 pav. Paprogramės „FAZ.NET-Ticker“ peržiūra

6. Eksperimentas ir metodo įvertinimas

6.1. Eksperimento projektas

Eksperimento tikslas - išanalizuoti automatizuoto įrankio panaudojimą, siekiant sukurti ir palyginti naujo portalo architektūrinės dalies pagrindinių komponentų kūrimo trukmę su pradinio portalo pagrindinių komponentų kūrimo trukme programuotojo požiūriu.

Įrankis naudoja ontologijas projektuoti ir rekonstruoti portalo struktūrą. Eksperimento metu bus bandoma įrodyti, kad automatizuotas įrankis pagrindinius architektūros failus sukuria greičiau nei projektuotojas rankiniu būdu, taip sutaupomas laikas, skirtas viso projekto įgyvendinimui. Automatizuoto įrankio nauda ta, kad sutaupyta laiką, galima paskirstyti kitiems projekto ciklo etapams.

Eksperimente priklausomas kintamasis yra laikas ir naudojama faktinė laiko skalė. Kadangi eksperimentas suskirstytas į dvi dalis, t.y. dalykinės srities kūrimo eksperimentą ir dalykinės srities pakeitimų eksperimentą. Pirmuoju atveju bus kuriami du nauji eksperimentai, antruoju atveju bus vykdoma 27 bandymai, kai projekto reikalavimai ir dalykinė sritis yra keičiama. Eksperimentui apibendrinti bus skaičiuojami vidurkiai.

Visus eksperimentus vykdo programuojas, kurio patirtis 3 metai, dirbantis pilną darbo dieną.

6.2. Dalykinės srities kūrimo eksperimentas

Kūrimo eksperimentas – tai eksperimentas, kurio metu sukuriama pradiniai architektūriniai failai. Eksperimento metu matuojami automatizuoto įrankio ir rankiniu būdu sukurtų objektų trukmė. Reikalingo laiko ontologijos perkūrimui neįskaičiuojamas, kadangi prieš naudojant automatinį įrankį ar rankiniu būdu kuriant dalykinės srities objektus, ontologija jau turi būti sukurta.

6.2.1. Žinių portalo eksperimentas

Žinių portalo ontologija sudaryta iš 27 objektų.

Eil. Nr.	Objektas*	Automatinio įrankio trukmė, min	Rankiniu būdu trukmė, min
1.	Archyvas	0.002	6
2.	Komentaras	0.002	6
3.	Paprograme	0.002	7
4.	Dokumentas	0.002	8
5.	Kategorija	0.002	6
6.	ArchSantrauka	0.002	7
7.	ArchStraipsnis	0.002	7
8.	KomVartotojas	0.002	6
9.	KomRedaktorius	0.002	6

10.	Apklausa	0.002	6
11.	Galerija	0.002	7
12.	PapKnygos	0.002	9
13.	PapBlog	0.002	6
14.	Komentarai	0.002	6
15.	PapVideo	0.002	6
16.	PapAktualijos	0.002	6
17.	PapPaieska	0.002	6
18.	DokSantrauka	0.002	7
19.	DokStraipsnis	0.002	7
20.	Media	0.002	7
21.	MelynaKnyga	0.002	6
22.	MokamiDokumentai	0.002	8
23.	Rss	0.002	7
24.	KatNustatymai	0.002	6
25.	Nuotrauka	0.002	8
26.	Video	0.002	8
27.	Garsas	0.002	8

* objektą sudaro dalykinės srities objektas, jungiamasis objektas

Automatinio įrankio vidurkis: 0.1 s = 0.002 min.

Rankiniu būdu vidurkis: 6,77 min.

6.2.2. Apgyvandinimo portalo eksperimentas

Apgyvandinimo portalo ontologija sudaryta iš 27 objektų. Struktūra šiek tiek sudėtingesnė už naujienu portalą, ontologija turi didesnę gylį.

Eil. Nr.	Objektas*	Automatinio įrankio trukmė, min	Rankiniu būdu trukmė, min
1.	Accommodation	0.002	6
2.	Contacts	0.002	4
3.	Destination	0.002	6
4.	Activity	0.002	7
5.	Hotel	0.002	6
6.	BedAndBreakfast	0.002	7
7.	Campground	0.002	7
8.	Beach	0.002	5
9.	RuralArea	0.002	6
10.	UrbaArea	0.002	6
11.	Sightseeing	0.002	6
12.	Sports	0.002	7
13.	Adventure	0.002	7
14.	Relaxation	0.002	6
15.	LuzuryHotel	0.002	7
16.	FamLand	0.002	7
17.	NationalPark	0.002	8
18.	City	0.002	7
19.	Town	0.002	7

20.	Museums	0.002	6
21.	Parks	0.002	7
22.	Surfing	0.002	8
23.	Hiking	0.002	8
24.	Safari	0.002	6
25.	BunjeeJumping	0.002	6
26.	Sunbathing	0.002	7
27.	Yoga	0.002	6

* objektą sudaro dalykinės srities objektas, jungiamasis objektas

Automatinio įrankio vidurkis: 0.1 s = 0.002 min.

Rankiniu būdu vidurkis: 6,52 min.

6.3. Dalykinės srities pakeitimų eksperimentas

Pakeitimų eksperimentas – tai eksperimentas, kurio metu pakeičiami ir atnaujinami jau sukurti pradiniai architektūros failai.

6.3.1. Naujienų portalo dalykinės srities pakeitimų eksperimentas

Naujienų portalas ontologija sudaryta iš 27 objektų.

Eil. Nr.	Objektas*	Veiksmas	Automatinio įrankio trukmė, min	Rankiniu būdu trukmė, min
1.	Archyvas	Pridedama parametru 1	0.002	4
2.	Komentaras	Pašalinama parametru 1	0.002	3
3.	Paprogramė	Pridedama parametru 1	0.002	3
4.	Dokumentas	Pašalinama parametru 1	0.002	2
5.	Kategorija	Pridedama parametru 1	0.002	4
6.	ArchSantrauka	Pridedama parametru 2	0.002	4
7.	ArchStraipsnis	Pašalinama parametru 1	0.002	3
8.	KomVartotojas	Pridedama parametru 2	0.002	5
9.	KomRedaktorius	Pridedama parametru 1	0.002	4
10.	Apklausa	Pašalinama parametru 1	0.002	4
11.	Galerija	Pridedama parametru 2	0.002	4
12.	PapKnygos	Pašalinama parametru 2	0.002	2
13.	PapBlog	Pašalinama parametru 1	0.002	3
14.	Komentarai	Pridedama parametru 2	0.002	4
15.	PapVideo	Pašalinama parametru 3	0.002	4
16.	PapAktualijos	Pašalinama parametru 1	0.002	3
17.	PapPaieska	Pridedama parametru 3	0.002	4
18.	DokSantrauka	Pašalinama parametru 1	0.002	3
19.	DokStraipsnis	Pašalinama parametru 1	0.002	3
20.	Media	Pridedama parametru 2	0.002	3
21.	MelynaKnyga	Pašalinama parametru 3	0.002	4
22.	MokamiDokumentai	Pašalinama parametru 1	0.002	4
23.	Rss	Pridedama parametru 3	0.002	5
24.	KatNustatymai	Pašalinama parametru 1	0.002	3

25.	Nuotrauka	Pašalinama parametru 2	0.002	4
26.	Video	Pridedama parametru 3	0.002	5
27.	Garsas	Pridedama parametru 2	0.002	2

Automatinio įrankio vidurkis: 0.1 s = 0.002 min.

Rankiniu būdu vidurkis: 4,89 min.

Parametru pridėta/pašalinta: 46.

Vidutinė trukmė pridėti/pašalinti parametru: 0,11 min.

6.3.2. Apgyvandinimo dalykinės srities pakeitimų eksperimentas

Eil. Nr.	Objektas*	Veiksmas	Automatinio įrankio trukmė, min	Rankiniu būdu trukmė, min
1.	Accommodation	Pašalinama parametru 1	0.002	3
2.	Contacts	Pridedama parametru 2	0.002	4
3.	Destination	Pridedama parametru 1	0.002	3
4.	Activity	Pridedama parametru 3	0.002	4
5.	Hotel	Pridedama parametru 2	0.002	3
6.	BedAndBreakfast	Pašalinama parametru 1	0.002	4
7.	Campground	Pašalinama parametru 2	0.002	4
8.	Beach	Pašalinama parametru 2	0.002	4
9.	RuralArea	Pridedama parametru 1	0.002	3
10.	UrbaArea	Pridedama parametru 1	0.002	4
11.	Sightseeing	Pridedama parametru 3	0.002	4
12.	Sports	Pašalinama parametru 2	0.002	3
13.	Adventure	Pridedama parametru 2	0.002	4
14.	Relaxation	Pridedama parametru 3	0.002	5
15.	LuzuryHotel	Pašalinama parametru 2	0.002	3
16.	FamLand	Pridedama parametru 3	0.002	5
17.	NationalPark	Pašalinama parametru 1	0.002	3
18.	City	Pridedama parametru 1	0.002	4
19.	Town	Pašalinama parametru 1	0.002	4
20.	Museums	Pridedama parametru 2	0.002	5
21.	Parks	Pašalinama parametru 2	0.002	3
22.	Surfing	Pridedama parametru 3	0.002	4
23.	Hiking	Pašalinama parametru 1	0.002	4
24.	Safari	Pridedama parametru 2	0.002	4
25.	BunjeeJumping	Pašalinama parametru 2	0.002	3
26.	Sunbathing	Pridedama parametru 1	0.002	4
27.	Yoga	Pašalinama parametru 2	0.002	3

Automatinio įrankio vidurkis: 0.1 s = 0.002 min.

Rankiniu būdu vidurkis: 3.75 min.

Parametru pridėta/pašalinta: 49.

Vidutinė trukmė pridėti/pašalinti parametru: 0,08 min.

6.4. Eksperimento išvados

Nenaudojant automatizuoto įrankio vidutinis dalykinės srities objekto kūrimo laikas yra 6,81 min, vidutinis perkūrimo laikas yra 4,32 min, naudojant automatizuotą įrankį kūrimas ir perkūrimas užtrunka 0.002 min.

Eksperimento metu pastebėta, kad rankiniu būdu vykdant dalykinės srities kūrimą arba perkūrimą, trukmė turi tendenciją trumpėti tolesniuose eksperimentuose. Sukurti apgyvendinimo portalo dalykinę sritį užtruko trumpiau nei naujienu portalo dalykinę sritį. Įtakos tam turėjo padidėjusi programuotojo patirtis skaityti ontologijos failą ir kurti dalykinės srities objektus. Deja trukmė sumažėja palyginti mažai negu užtrunka automatinio įrankiui atlikti analogišką darbą.

Naudojant automatizuotą įrankį išvengiama klaidų, kurias programuotojas gali padaryti reikalavimus įvedant rankiniu būdu.

Įrankį taikyti ypač verta, jei pakeitimai daromi dažnai.

7. Išvados

1. Šiame darbe iškeltas tikslas - pritaikyti DDD metodą, patobulintą automatizuotu dalykinės srities sudarymą iš ontologijos failo – buvo pasiektas išanalizuojant DDD metodą ir sukuriant automatizuotą įrankį.

2. Atlikta MDA, AOP, DDD metodų ir ontologijų kalbos OWL analizė parodė, kad lankstiems portalams kurti tikslinga pasirinkti DDD ir papildyti jį ontologijos aprašo taikymu. Naudojant DDD kūrimo metodą, pasiekiamas greitas ir patogus naujų reikalavimų įvedimas į projektą. Išanalizavus ontologijos kūrimo įrankius pasirinktas patogiausias įrankis, ontologijos kūrimui.

3. Nustatomi projekto įgyvendinimui reikalingi procesai, jų įgyvendinimui reikalingos priemonės. Aprašomi baziniai elementai.

4. Sukurto automatizuoto įrankio testavimas patvirtino, kad įrankis geba automatiškai atlikti reikiamas funkcijas:

- nuskaityti ontologijos failą, sukurtą programa „Protégé“.
- sukurti dalykinės srities architektūrinius failus.

5. Realizuoti du žinių portalai patvirtino metodo tinkamumą. Žinių portalo realizacija parodė, kad pasirinktas metodas ir portalo kūrimo žingsniai įgyvendinti. Panaudojimas DDD architektūrinių komponentų galimas, jei jau sukurta analogiškų projektų ir galimas tų komponentų panaudojimas. Automatizuotas įrankis padeda greičiau sukurti pradinius architektūrinius failus.

6. Eksperimentas parodė, kad automatizuotas įrankis sutrumpina portalo modifikavimo laiką. Ypač jis naudingas tada, kai modifikavimai vykdomi dažnai

8. Literatūra

1. Fred Brooks, “No Silver Bullet – Essence and Accident in Software Engineering”, Proceedings of the IFIP Tenth World Computing Conference, pp. 1069-1076, 1986
2. John D. Poole, „Model-Driven Architecture: Vision, Standards And Emerging Technologies“, http://www.omg.org/mda/mda_files/Model-Driven_Architecture.pdf
3. Aspect-Oriented Software Development, <http://www.aspectprogramming.com/aosd>, http://en.wikipedia.org/wiki/Aspect-oriented_programming
4. Eric Evans / edited by Abel Avram and Floyd Marinescu, “Domain-Driven Design Quickly”, December 2006, <http://www.infoq.com/minibooks/domain-driven-design-quickly>
5. CORCHO, O.; FERNANDEZ-LOPEZ, M.; GOMEZ-PEREZ, A. Methodologies, tools and languages for building ontologies. Where is their meeting point? Facultad de Informatika, Universidad Politecnica de Madrid. Madrid, Spain, 2002.
6. A comparative analysis of methodologies, tools and languages used for building ontologies. CCLRC, ESC.
7. GANDON, F. Ontology Engineering: A survey and a return on experience. Institut de Recherche en Informatique et Automatique, 2002.

9. Priedai

9.1. Žinių portalo ontologijos failo turinys

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY NaujienuPortalas4
"http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#" >
]>
<rdf:RDF
xmlns="http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#"

xml:base="http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl"
  xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

xmlns:NaujienuPortalas4="http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:owl="http://www.w3.org/2002/07/owl#">
  <owl:Ontology rdf:about=""/>
  <!--

////////////////////////////////////
////////////////////////////////////
//
// Data properties
//

////////////////////////////////////
////////////////////////////////////
--> <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Antraste
--> <owl:DatatypeProperty rdf:about="#Antraste">
  <rdfs:domain rdf:resource="#DokSantrauka"/>
  <rdfs:domain rdf:resource="#DokStraipsnis"/>
  <rdfs:domain rdf:resource="#Rss"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#ArMatoma
--> <owl:DatatypeProperty rdf:about="#ArMatoma">
  <rdfs:domain rdf:resource="#KatNustatymai"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#ArMokamas
s --> <owl:DatatypeProperty rdf:about="#ArMokamas">
  <rdfs:domain rdf:resource="#ArchSantrauka"/>
  <rdfs:domain rdf:resource="#ArchStraipsnis"/>
  <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Autorius
--> <owl:DatatypeProperty rdf:about="#Autorius">
  <rdfs:domain rdf:resource="#DokStraipsnis"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>
```

```

    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#GaliojaI
ki -->    <owl:DatatypeProperty rdf:about="#GaliojaIki">
        <rdfs:domain rdf:resource="#Dokumentas"/>
        <rdfs:range rdf:resource="&xsd;dateTime"/>
    </owl:DatatypeProperty>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#GaliojaN
uo -->    <owl:DatatypeProperty rdf:about="#GaliojaNuo">
        <rdfs:domain rdf:resource="#Dokumentas"/>
        <rdfs:range rdf:resource="&xsd;dateTime"/>
    </owl:DatatypeProperty>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Komentar
as -->    <owl:DatatypeProperty rdf:about="#Komentaras">
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Lygis --
>    <owl:DatatypeProperty rdf:about="#Lygis">
        <rdfs:domain rdf:resource="#KatNustatymai"/>
        <rdfs:range rdf:resource="&xsd;int"/>
    </owl:DatatypeProperty>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#MenuK
ategorija -->    <owl:DatatypeProperty rdf:about="#MenuKategorija">
        <rdfs:domain rdf:resource="#KatNustatymai"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Modifika
vimoData -->    <owl:DatatypeProperty rdf:about="#ModifikavimoData">
        <rdfs:domain rdf:resource="#Dokumentas"/>
        <rdfs:range rdf:resource="&xsd;dateTime"/>
    </owl:DatatypeProperty>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Redaktor
ius -->    <owl:DatatypeProperty rdf:about="#Redaktorius">
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Subantra
ste -->    <owl:DatatypeProperty rdf:about="#Subantraste">
        <rdfs:domain rdf:resource="#DokSantrauka"/>
        <rdfs:domain rdf:resource="#Rss"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Sukurimo
Data -->    <owl:DatatypeProperty rdf:about="#SukurimoData">
        <rdfs:domain rdf:resource="#Archyvas"/>
        <rdfs:domain rdf:resource="#Dokumentas"/>
        <rdfs:domain rdf:resource="#Komentaras"/>
        <rdfs:range rdf:resource="&xsd;dateTime"/>
    </owl:DatatypeProperty>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Tekstas
-->    <owl:DatatypeProperty rdf:about="#Tekstas">
        <rdfs:domain rdf:resource="#ArchStraipsnis"/>
        <rdfs:domain rdf:resource="#DokStraipsnis"/>
        <rdfs:domain rdf:resource="#MelynaKnyga"/>
        <rdfs:domain rdf:resource="#MokamiDokumentai"/>
        <rdfs:range rdf:resource="&xsd;string"/>
    </owl:DatatypeProperty>

```

```

<!--

////////////////////////////////////
////////////////////////////////////
//
// Classes
//

////////////////////////////////////
////////////////////////////////////
--> <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#ArchSant
rauka --> <owl:Class rdf:about="#ArchSantrauka">
    <rdfs:subClassOf rdf:resource="#Archyvas"/>
</owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#ArchStra
ipsnis --> <owl:Class rdf:about="#ArchStraipsnis">
    <rdfs:subClassOf rdf:resource="#Archyvas"/>
</owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Archyvas
--> <owl:Class rdf:about="#Archyvas"/>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#DokSantr
auka --> <owl:Class rdf:about="#DokSantrauka">
    <rdfs:subClassOf rdf:resource="#Dokumentas"/>
</owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#DokStrai
psnis --> <owl:Class rdf:about="#DokStraipsnis">
    <rdfs:subClassOf rdf:resource="#Dokumentas"/>
</owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Dokument
as --> <owl:Class rdf:about="#Dokumentas"/>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#KatNusta
tymai --> <owl:Class rdf:about="#KatNustatymai">
    <rdfs:subClassOf rdf:resource="#Kategorija"/>
</owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Kategori
ja --> <owl:Class rdf:about="#Kategorija"/>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#KomRedak
torius --> <owl:Class rdf:about="#KomRedaktorius">
    <rdfs:subClassOf rdf:resource="#Komentaras"/>
</owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#KomVarto
tojas --> <owl:Class rdf:about="#KomVartotojas">
    <rdfs:subClassOf rdf:resource="#Komentaras"/>
</owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Komentar
as --> <owl:Class rdf:about="#Komentaras"/>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Media --
> <owl:Class rdf:about="#Media">
    <rdfs:subClassOf rdf:resource="#Dokumentas"/>
</owl:Class>

```

```

<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#MediaGarsas -->
  <owl:Class rdf:about="#MediaGarsas">
    <rdfs:subClassOf rdf:resource="#Media"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#MediaNuotrauka -->
  <owl:Class rdf:about="#MediaNuotrauka">
    <rdfs:subClassOf rdf:resource="#Media"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#MediaVideo -->
  <owl:Class rdf:about="#MediaVideo">
    <rdfs:subClassOf rdf:resource="#Media"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#MelynaKnyga -->
  <owl:Class rdf:about="#MelynaKnyga">
    <rdfs:subClassOf rdf:resource="#Dokumentas"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#MokamiDokumentai -->
  <owl:Class rdf:about="#MokamiDokumentai">
    <rdfs:subClassOf rdf:resource="#Dokumentas"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#PapAktualijos -->
  <owl:Class rdf:about="#PapAktualijos">
    <rdfs:subClassOf rdf:resource="#Papprograme"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#PapApklausa -->
  <owl:Class rdf:about="#PapApklausa">
    <rdfs:subClassOf rdf:resource="#Papprograme"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#PapBlog -->
  <owl:Class rdf:about="#PapBlog">
    <rdfs:subClassOf rdf:resource="#Papprograme"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#PapGalerija -->
  <owl:Class rdf:about="#PapGalerija">
    <rdfs:subClassOf rdf:resource="#Papprograme"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#PapKnygos -->
  <owl:Class rdf:about="#PapKnygos">
    <rdfs:subClassOf rdf:resource="#Papprograme"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#PapKomentaras -->
  <owl:Class rdf:about="#PapKomentaras">
    <rdfs:subClassOf rdf:resource="#Papprograme"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#PapPaieska -->
  <owl:Class rdf:about="#PapPaieska">
    <rdfs:subClassOf rdf:resource="#Papprograme"/>
  </owl:Class>
<!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#PapVideo -->
  <owl:Class rdf:about="#PapVideo">
    <rdfs:subClassOf rdf:resource="#Papprograme"/>

```

```

    </owl:Class>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Paprograme -->
    <owl:Class rdf:about="#Paprograme"/>
    <!--
http://www.semanticweb.org/ontologies/2010/3/NaujienuPortalas4.owl#Rss -->
    <owl:Class rdf:about="#Rss">
        <rdfs:subClassOf rdf:resource="#Dokumentas"/>
    </owl:Class>
</rdf:RDF><!-- Generated by the OWL API (version 2.2.1.1138)
http://owlapi.sourceforge.net -->

```

9.2. Apgyvandinimo portalo ontologijos failo turinys

```

<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
    <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
    <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
    <!ENTITY owl2xml "http://www.w3.org/2006/12/owl2-xml#" >
    <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
    <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
    <!ENTITY Accommodation
"http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#" >
]>

<rdf:RDF
xmlns="http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#"

xml:base="http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl"
    xmlns:owl2xml="http://www.w3.org/2006/12/owl2-xml#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"

xmlns:Accommodation="http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#">
    <owl:Ontology rdf:about=""/>

    <!--

////////////////////////////////////
////////////////////////////////////
//
// Data properties
//
////////////////////////////////////
////////////////////////////////////
-->
    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#AccommodationRating -->

    <owl:DatatypeProperty rdf:about="#AccommodationRating">
        <rdfs:domain rdf:resource="#Accommodation"/>
        <rdfs:range rdf:resource="&xsd:int"/>
    </owl:DatatypeProperty>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Adress -->

    <owl:DatatypeProperty rdf:about="#Adress">

```

```

        <rdfs:domain rdf:resource="#Contacts"/>
        <rdfs:range rdf:resource="&xsd:string"/>
    </owl:DatatypeProperty>

    <!--
    http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Animals -->

    <owl:DatatypeProperty rdf:about="#Animals">
        <rdfs:domain rdf:resource="#Parks"/>
        <rdfs:domain rdf:resource="#RuralArea"/>
        <rdfs:range rdf:resource="&xsd:string"/>
    </owl:DatatypeProperty>

    <!--
    http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Cultures --
    >

    <owl:DatatypeProperty rdf:about="#Cultures">
        <rdfs:domain rdf:resource="#Parks"/>
        <rdfs:domain rdf:resource="#RuralArea"/>
        <rdfs:range rdf:resource="&xsd:string"/>
    </owl:DatatypeProperty>

    <!--
    http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Density -->

    <owl:DatatypeProperty rdf:about="#Density">
        <rdfs:domain rdf:resource="#UrbanArea"/>
        <rdfs:range rdf:resource="&xsd:float"/>
    </owl:DatatypeProperty>

    <!--
    http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Description
    -->

    <owl:DatatypeProperty rdf:about="#Description">
        <rdfs:domain rdf:resource="#Accommodation"/>
        <rdfs:domain rdf:resource="#Activity"/>
        <rdfs:domain rdf:resource="#Destination"/>
        <rdfs:range rdf:resource="&xsd:string"/>
    </owl:DatatypeProperty>

    <!--
    http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#EMail -->

    <owl:DatatypeProperty rdf:about="#EMail">
        <rdfs:domain rdf:resource="#Contacts"/>
        <rdfs:range rdf:resource="&xsd:string"/>
    </owl:DatatypeProperty>

    <!--
    http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Entertainme
    nts -->

    <owl:DatatypeProperty rdf:about="#Entertainments">
        <rdfs:domain rdf:resource="#Adventure"/>
        <rdfs:domain rdf:resource="#Sunbathing"/>
        <rdfs:range rdf:resource="&xsd:string"/>
    </owl:DatatypeProperty>

```



```

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Popularity
-->

<owl:DatatypeProperty rdf:about="#Popularity">
  <rdfs:domain rdf:resource="#UrbanArea"/>
  <rdfs:range rdf:resource="&xsd:int"/>
</owl:DatatypeProperty>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Price -->

<owl:DatatypeProperty rdf:about="#Price">
  <rdfs:domain rdf:resource="#Accommodation"/>
  <rdfs:domain rdf:resource="#Activity"/>
  <rdfs:domain rdf:resource="#Destination"/>
  <rdfs:range rdf:resource="&xsd:double"/>
</owl:DatatypeProperty>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#RearAnimals
-->

<owl:DatatypeProperty rdf:about="#RearAnimals">
  <rdfs:domain rdf:resource="#NationalPark"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#RiskLevel -
->

<owl:DatatypeProperty rdf:about="#RiskLevel">
  <rdfs:domain rdf:resource="#Activity"/>
  <rdfs:range rdf:resource="&xsd:double"/>
</owl:DatatypeProperty>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Squar -->

<owl:DatatypeProperty rdf:about="#Squar">
  <rdfs:domain rdf:resource="#RuralArea"/>
  <rdfs:domain rdf:resource="#UrbanArea"/>
  <rdfs:range rdf:resource="&xsd:double"/>
</owl:DatatypeProperty>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Subtitle --
>

<owl:DatatypeProperty rdf:about="#Subtitle">
  <rdfs:domain rdf:resource="#Accommodation"/>
  <rdfs:domain rdf:resource="#Activity"/>
  <rdfs:domain rdf:resource="#Destination"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Telephone -
->

```

```

<owl:DatatypeProperty rdf:about="#Telephone">
  <rdfs:domain rdf:resource="#Contacts"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Title -->

<owl:DatatypeProperty rdf:about="#Title">
  <rdfs:domain rdf:resource="#Accommodation"/>
  <rdfs:domain rdf:resource="#Activity"/>
  <rdfs:domain rdf:resource="#Destination"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Width -->

<owl:DatatypeProperty rdf:about="#Width">
  <rdfs:domain rdf:resource="#Beach"/>
  <rdfs:range rdf:resource="&xsd:float"/>
</owl:DatatypeProperty>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#YearOfEstablishment -->

<owl:DatatypeProperty rdf:about="#YearOfEstablishment">
  <rdfs:domain rdf:resource="#Museums"/>
  <rdfs:domain rdf:resource="#UrbanArea"/>
  <rdfs:range rdf:resource="&xsd:dateTime"/>
</owl:DatatypeProperty>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#ZipCode -->

<owl:DatatypeProperty rdf:about="#ZipCode">
  <rdfs:domain rdf:resource="#Contacts"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<!--
////////////////////////////////////
////////////////////////////////////
////////////////////////////////////
//
// Classes
//
////////////////////////////////////
////////////////////////////////////
-->
<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Accommodation -->

<owl:Class rdf:about="#Accommodation"/>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Activity -->
<owl:Class rdf:about="#Activity"/>

```

```

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Adventure -
->
    <owl:Class rdf:about="#Adventure">
        <rdfs:subClassOf rdf:resource="#Activity"/>
    </owl:Class>
    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Beach -->
    <owl:Class rdf:about="#Beach">
        <rdfs:subClassOf rdf:resource="#Destination"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#BedAndBreak
fast -->
    <owl:Class rdf:about="#BedAndBreakfast">
        <rdfs:subClassOf rdf:resource="#Accommodation"/>
    </owl:Class>
    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#BunjeeJumpi
ng -->

    <owl:Class rdf:about="#BunjeeJumping">
        <rdfs:subClassOf rdf:resource="#Adventure"/>
    </owl:Class>
    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Campground
-->
    <owl:Class rdf:about="#Campground">
        <rdfs:subClassOf rdf:resource="#Accommodation"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#City -->

    <owl:Class rdf:about="#City">
        <rdfs:subClassOf rdf:resource="#UrbanArea"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Contacts --
>
    <owl:Class rdf:about="#Contacts"/>
    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Destination
-->

    <owl:Class rdf:about="#Destination"/>
    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#FarmLand --
>

    <owl:Class rdf:about="#FarmLand">
        <rdfs:subClassOf rdf:resource="#RuralArea"/>
    </owl:Class>
    <!-- http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Hiking
-->

    <owl:Class rdf:about="#Hiking">
        <rdfs:subClassOf rdf:resource="#Sports"/>
    </owl:Class>

    <!-- http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Hotel
-->

```

```

    <owl:Class rdf:about="#Hotel">
      <rdfs:subClassOf rdf:resource="#Accommodation"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#LuxuryHotel
-->
    <owl:Class rdf:about="#LuxuryHotel">
      <rdfs:subClassOf rdf:resource="#Hotel"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Museums -->

    <owl:Class rdf:about="#Museums">
      <rdfs:subClassOf rdf:resource="#Sightseeing"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#NationalPar
k -->

    <owl:Class rdf:about="#NationalPark">
      <rdfs:subClassOf rdf:resource="#RuralArea"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Parks -->

    <owl:Class rdf:about="#Parks">
      <rdfs:subClassOf rdf:resource="#Sightseeing"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Relaxation
-->

    <owl:Class rdf:about="#Relaxation">
      <rdfs:subClassOf rdf:resource="#Activity"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#RuralArea -
->

    <owl:Class rdf:about="#RuralArea">
      <rdfs:subClassOf rdf:resource="#Destination"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Safari -->

    <owl:Class rdf:about="#Safari">
      <rdfs:subClassOf rdf:resource="#Adventure"/>
    </owl:Class>

    <!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Sightseeing
-->

    <owl:Class rdf:about="#Sightseeing">
      <rdfs:subClassOf rdf:resource="#Activity"/>
    </owl:Class>

```

```

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Sports -->

<owl:Class rdf:about="#Sports">
  <rdfs:subClassOf rdf:resource="#Activity"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Sunbathing
-->

<owl:Class rdf:about="#Sunbathing">
  <rdfs:subClassOf rdf:resource="#Relaxation"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Surfing -->

<owl:Class rdf:about="#Surfing">
  <rdfs:subClassOf rdf:resource="#Sports"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Town -->

<owl:Class rdf:about="#Town">
  <rdfs:subClassOf rdf:resource="#UrbanArea"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#UrbanArea -
->

<owl:Class rdf:about="#UrbanArea">
  <rdfs:subClassOf rdf:resource="#Destination"/>
</owl:Class>

<!--
http://www.semanticweb.org/ontologies/2009/11/Accommodation.owl#Yoga -->

<owl:Class rdf:about="#Yoga">
  <rdfs:subClassOf rdf:resource="#Relaxation"/>
</owl:Class>
</rdf:RDF>

<!-- Generated by the OWL API (version 2.2.1.1138)
http://owlapi.sourceforge.net -->

```