

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Mantas Krugiškis

**Ryšių su klientais valdymo sistema metaduomenų
pagrindu**

Magistro darbas

Darbo vadovas
prof. dr. S. Gudas

Kaunas, 2008

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
INFORMACIJOS SISTEMŲ KATEDRA

Mantas Krugiškis

**Ryšių su klientais valdymo sistema metaduomenų
pagrindu**

Magistro darbas

Recenzentas
doc. dr. A. Lenkevičius
2008-01-

Vadovas
prof. dr. S. Gudas
2008-01-

Atliko
IFM-2/4 gr. stud.
Mantas Krugiškis
2008-01-

Kaunas, 2008

Turinys

1.	Įvadas.....	4
1.1.	Darbo tikslas.....	4
1.2.	Tyrimo sritis, objektas, problema ir naudota programinė įranga.....	4
1.3.	Tyrimo aplinka.....	8
1.4.	Rinkoje esantys vartotojai.....	8
2.	Portalų metamodelio ir šablonų metodologijos, tyrimo aplinka ir architektūra.....	9
2.1.	Panašios sistemos Lietuvoje.....	9
2.2.	Architektūra ir galimi įgyvendinimo priemonių variantai.....	11
2.3.	Metaduomenys.....	24
2.4.	MDE technologija.....	36
2.5.	UML koncepcija.....	39
2.6.	Ontologijos.....	42
2.7.	Modeliu paremta inžinerija.....	46
2.8.	Darbo tikslas ir siekiami privalumai.....	47
2.9.	Kompiuterizuojamos sistemos funkcijos.....	47
2.10.	Reikalavimai duomenims.....	50
2.11.	Nefunkciniai reikalavimai ir apribojimai.....	50
2.12.	Analizės išvados.....	52
3.	Projektinė dalis.....	53
3.1.	Įvadas.....	53
3.2.	Portalo dalykinės srities modelis.....	53
3.3.	Projektuojamo portalo projektas.....	60
3.4.	Išvados.....	72
4.	Realizacija.....	73
4.1.	DB ryšių schema.....	73
4.2.	Testavimo modelis bei duomenys, kontrolinis pavyzdys.....	79
4.3.	Vartotojo sąsaja.....	81
4.4.	Bandomosios eksploatacijos rezultatų aprašymas.....	88
4.5.	Išvados.....	94
5.	Sistemos vartotojo dokumentas.....	95
5.1.	Neregistruotas vartotojas.....	95
5.2.	Registruotas vartotojas.....	100
5.3.	Registruotos įmonės dalis.....	100
5.4.	Portalo administratorius.....	101
5.5.	Sistemos administravimo galimybės.....	102
5.6.	HTML redaktorius.....	106
6.	Išvados.....	107
7.	Literatūra.....	109
8.	Metadata based CRM system.....	111
8.1.	Summary.....	111
9.	Priedai.....	112
9.1.	Duomenų bazės lentelės.....	112
9.2.	HTML redaktorius.....	121
9.3.	Tinklalapių atvaizdai.....	126
9.4.	Straipsnis.....	129

1. Įvadas

Tobulėjant interneto sistemoms, tapo įprasta naudoti dinامينius tinklo puslapius, kurie reaguoja į vartotojo įvestį. Serveryje veikiančių puslapių kūrimas - Active Server Pages (ASP) ir Java Server Pages (JSP) leidžia kūrėjams tiesiogiai įdėti tekstus į HTML puslapius, tokiu būdu suprastinant programavimo modelį. Sudėtingesniems įmontuotų tekstų taikymams atsirado poreikis atskirti veiklos logiką nuo pateikties logikos puslapių lygyje. Deja, nėra vienos kūrimo strategijos, kuri tiktų visoms situacijoms. Dėl programinės įrangos kūrimo priemonių konkurencijos atsirado būtinybė pašalinti perdėtą įvairovę ir sudėtingumą.

1.1. Darbo tikslas

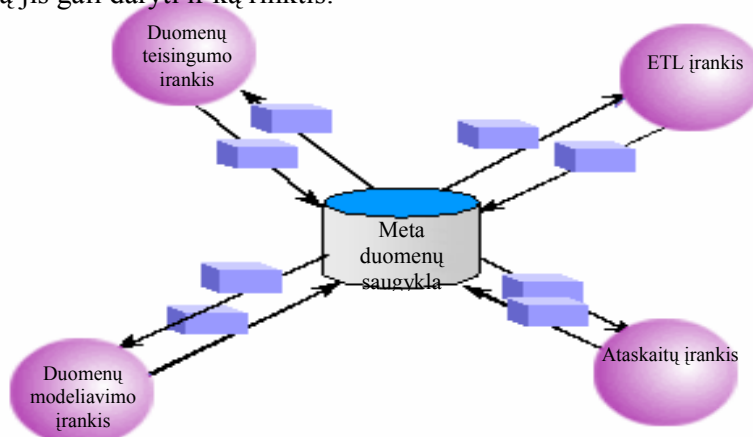
- ✚ Išanalizuoti sukurtų tinklalapių kūrimo metodologijas,
- ✚ Palyginti tinklalapius vartotojo atžvilgiu.
- ✚ Sukurti duomenų bazę, paremtą MySQL duomenų bazių valdymo sistema.
- ✚ Sukurti metaduomenų bazę, kurios pagalba būtų galima įterpti naują meniu punktą tiesiai per metabazę, kad vartotojas galėtų lengviau dalyvauti portale.
- ✚ Suprojektuoti metaduomenų lygį, kuris leistų patogiau portalą administruoti.
- ✚ Sukurti tokį metabazės lygį, kuris leistų greičiau pasiekti duomenis, greičiau perprojektuoti portalą ir jį plėtoti.

1.2. Tyrimo sritis, objektas, problema ir naudota programinė įranga

Sistemos objektas yra: metaduomenų taikymas informacijos sistemų funkcionalumui gerinti.

Norima sukurti tokią žinių bazę, kuri galėtų padėti puikiai orientuotis būsto statyboje ir remonte. Žinių bazės pagalba būtų galima spręsti iškilusias statybos problemas bei būtų galima sužinoti, kokias medžiagas pasirinkti pagal savo finansines galimybes bei stilių.

Norima sukurti metamodelį, per kurį būtų kreipiamasi į duomenų bazę. Kai vartotojas norėtų atlikti tam tikrus veiksmus, šis metamodelis padėtų jam geriau orientuotis ir pasiūlytų keletą galimybių, ką jis gali daryti ir ką rinktis.



1.1 pav. Metaduomenų architektūra.

Darbe panaudotos šios programinės priemonės:

- ✚ Macromedia DreamWeaver (vartotojo sąsajos realizacija bei puslapio struktūros koregavimui);
- ✚ PHPMyAdmin (įrankis, skirtas DB konfigūracijai keisti)
- ✚ Adobe Photoshop
- ✚ Zend Framework 1.0.0 – PHP programavimo aplinka
- ✚ MagicDraw ir MS Visio (UML diagramos);
- ✚ MS Word (pateikta dokumentacija);
- ✚ MS PowerPoint.

1.2.1. Macromedia DreamWeaver

"Macromedia Dreamweaver® 8" – pramoninis standartas kurti žiniatinklio priedus, kuris leidžia vartotojams veiksmingai projektuoti, kurti ir palaikyti interneto tinklalapius ir priedus. "Su "Dreamweaver 8" kūrėjai gauna galimybę kurti ir prižiūrėti tinklalapius bei priedus, kurie palaiko pačias šiuolaikiškiausias technologijas ir algoritmus. Naujos galimybės ir optimizavimo priemonės sutrumpina rutininio darbo laiką. Integravus XML duomenis galingu "drag-and-drop" mechanizmu, atsiranda galimybė keisti dizaino mastelį ir valdyti projektą, grupuoti kodo fragmentus, galimybė įdiegti į projektą "Flash® Video" viso labo penkiais pelės spragtelejimais.

Galimybė dirbti su geriausiais algoritmais.

"Dreamweaver 8" užtikrina precedento neturintį geriausių algoritmų palaikymą ir tai tapo pramoniniu standartu. Galimybė naudoti vizualias projektavimo priemones XML duomenų integravimui į interneto puslapius, naudojantis paprasta duomenų importavimo sąsaja "drag-and-drop" režimu. Galimybė pereiti į darbo su išeitiniu kodu režimą ir pasinaudoti jutikliniais XML ir XSLT patarimais dėl sintaksės. Su nauju suvienytu CSS pultu lengvai galima išnagrinėti ir dirbti su puslapiais kurti naudojama stilių lentelė. Visos CSS funkcinės galimybės sujungtos į vieną grupę ir išplėstos, kad darbas būtų lengvesnis ir našesnis. Nauja sąsaja palengvina stilių taikymą elementuose, o savybių lentelė pagreitina redagavimą. Galima pasinaudoti vaizdiniais patarimais, kad nustatyti stilių taikymo ribas, naudoti savo žymėjimą spalvomis, kad vizualiai kontroliuoti sudėtingų sistemų įdiegimo lygį. Iššokantys patarimai padeda geriau valdyti pakopinių lentelių stilių elementus projekte. Naudojantis stilių perjungimo pulteliu galima kontroliuoti, kaip atrodo projektas ekrane, ant popieriaus arba mobiliajame įrenginyje. Dabar "Dreamweaver" visiškai palaiko šiuolaikines CSS priemones, tokias kaip užklojimas, pseudoelementai ir formų elementai.

Įdiegtas audito įrankis dabar palaiko "Section 508" ir "WCAG Priority 1 bei 2" standartų suderinamumo patikrą. WebDAV palaiko "Digest Authentication" ir SSL protokolus, kad bylos būtų perduodamos saugiai ir būtų geresnės galimybės susijungti su dar didesniu serverių kiekiu.

Daugiau laiko kūrybai.

Naujos galimybės ir optimizavimo priemonės sutrumpina laiką rutininiams užduotims atlikti. Galimybė keisti projekto lango mastelį, kad geriau valdyti detales, atrinkti ir didinti vaizdus bei dirbti su įkrautais sluoksniais arba atskirti ir įvertinti, kaip tinklalapyje atrodys visas projektas. Susikoncentruoti tik į aktualų kodą. Grupuoti ir atnaujinti kodo blokus pagal išskyrimą ar žymas, kad kontroliuoti projekto struktūrą. Nebūtina laukti, kol "Dreamweaver 8" įkraus bylas į serverį - naujos priemonės suteikia galimybę dirbti su bylomis vietiniame kompiuteryje, kol jungiamasi su serveriu. Lyginti gauto puslapio išvaizdą su projektu, naudojantis maksimaliai tiksliomis liniuotėmis puslapio matmenims koreguoti. Vizuali kontrolė padeda tiksliai išmatuoti atstumus ir palaiko intelektualią sąsają su krepiamosiomis. Naujojo kodo skydelio meniu yra daugelis standartinių operacijų, išdėstytų redaktoriaus lange. Daugiau nereikia ieškoti meniu ar skydeliuose paruoštų blokų ar kodo fragmentų. Nauja galimybė leidžia komentuoti kodo blokus ir naikinti komentarus. Tobulesnis redaktorius suteikia galimybę geriau valdyti kodą dėl sintaksės patarimų ir automatiškai pabaigiamų žymų, išlaikydamas kodavimo stilių. Naujos dokumentų šaknelės "Mac®" platformai supaprastina vartotojo sąsają ir palengvina dokumentų paiešką bei atranką. Puikūs naujieji šablonai suteikia vartotojams galimybę praeiti kelią nuo paketo įdiegimo iki galutinio tinklalapio dizaino be kliūčių. Greito bylų sulyginimo funkcija leidžia rasti skirtumus nepaisant, ar šios bylos yra vietinės, viena vietinė, kita nuotolinė, ar abi iš jų yra nuotoliniame serveryje. Dabar "Dreamweaver" suderinamas su daugeliu lyginamųjų taikomųjų "Windows" ir "Mac" platformos programėlių. Specialaus įterpinio funkcija sutrumpina turinio importavimo iš elektroninio pašto ar "Microsoft® Word®" laiką. Nauja įterpinio funkcija leidžia išsaugoti pirminio teksto, sukurtą "Microsoft Word" programoje, formatavimą, arba įterpti tekstą ir panaudoti CSS šabloną, jau esantį prie puslapio.

Šiuolaikinių technologijų ir standartų palaikymas.

"Dreamweaver 8" sėkmingai įsisavina ir palaiko naujas technologijas. Galimybė įtraukti "Flash Video" į tinklalapio dizainą, įdėti ir derinti tinklalapiuose vaizdo pristatymus, naudojantis dialogine sąsaja penkiais pelės spragtelėjimais. "ColdFusion® MX 7" palaikymas suteikia naujų scenarijų serveryje ir patarimų dėl sintaksės. Glaudi "Dreamweaver" ir "ColdFusion" integracija leidžia vartotojams kurti ir naikinti duomenų bazes tiesiai iš "Dreamweaver" komponentų skydelio, taip pat naviguoti po CFCs elementus. Naujojoje versijoje atnaujintas PHP 5 palaikymas, tarp jų scenarijai serveryje ir patarimai dėl sintaksės. Įvykių įrašai ir pranešimai padeda sekti tinklalapio pokyčius. "Dreamweaver" perduoda įvykių sąrašą "Macromedia Web Publishing System" serveriui,

visus pokyčius registruoja ekosistema WPS. "Dreamweaver" pagalbos sistemoje yra išpūdinga "O'Reilly®" atnaujintų medžiagų kolekcija – naujos XML, XSLT ir XPath nuorodos ir atnaujinti ASP bei JSP dokumentai.

1.2.2. PhpMyAdmin

PhpMyAdmin yra valdymo įrankis parašytas PHP kalba, skirtas administruoti MySQL duomenų bazes naudojantis naršykle. Šiuo metu jis gali kurti ir šalinti duomenų bazes, manipuliuoti lentelėmis bei įrašais (kurti, šalinti, keisti lenteles bei įrašus, tvarkyti ryšius), vykdyti SQL sakinius.

1.2.3. Microsoft Office Visio2000

Naudojamas įvairių tipų schemų ir diagramų kūrimui ir vaizdavimui. Orientuotas plačiam naudotojų ratui, Visio2000 padeda optimizuoti darbą, eliminuoti nereikalingas operacijas, padidinti veiklos efektyvumą:

- ✚ greitas diagramų kūrimas naudojant jau paruoštas (Microsoft SmartShapes®) figūras;
- ✚ bendro pobūdžio diagramų sudarymas iš einamų duomenų;
- ✚ kontekstinės informacijos ir šablonų nuolatinis atnaujinimas iš Interneto.

Vaizdinis procesu pristatymas pateikiant įvairių tipų ir sudėtingumo diagramas:

- ✚ vaizdžių diagramų kūrimas ir jų pristatymas auditorijai;
- ✚ galimybė dirbti ir taisyti diagramas darbinėje aplinkoje Microsoft Windows® SharePoint™ Services;

supaprastintas ranka rašytų pastabų prie diagramų įvedimas planšetinio kompiuterio pagalba;

- ✚ koregavimo režimas leidžia sekti ir taisyti komentarus;
- ✚ galimybė išsaugoti diagramas web formatu;
- ✚ diagramų importas ir eksportas SVG formatu (Scalable Vector Graphics), nauju XML

(Extensible Markup Language).

Yra galimybė naudoti integruotas ir automatizuotas sistemas ir procesus, vykdamas sudėtingesnius uždavinius, darbo našumui didinti siekiant veiklos sėkmės:

- ✚ biznio procesu ir sistemų integracija importuojant ar eksportuojant duomenis iš MS Visio diagramų į taikomąsias programas MS Access, MS Excel, MS Word, MS SQL Server, XML ir kitais formatais ;

- ✚ galimybė integruoti Visio2000 į galingus Microsoft .NET produktus, konkrečių biznio poreikių tenkinimui ;
- ✚ galimybė naudoti Visio2000 grafinius elementus kituose .NET produktuose.

1.3. Tyrimo aplinka




Lietuvoje sparčiai auga statybų mastai, todėl statybos sektorius yra vienu iš sparčiausiai besiplėtojančių Lietuvos ūkio šakų. Šalyje nemažėja susidomėjimas nekilnojamu turtu, naujos statybos būstais, administraciniais, pramoniniais pastatais. Šiais naujais technologijos laikais labai sparčiai plėtojamas elektroninis verslas, kuris yra neatsiejamas nuo interneto. Internetu vis daugiau statybos bendrovių teikia informaciją, bendrauja su klientais, vykdo projektus. Informacijos srutai labai stipriai auga, žmonėms tampa vis sunkiau ir sunkiau rasti reikiamą informaciją apie būsto statybą. Todėl atsiranda poreikis portalo, kuriame būtų susisteminta informacija apie būtiniausius statybos kaštus ir statybos bendrovių teikiamas paslaugas, jau esamus projektus.

1.4. Rinkoje esantys vartotojai

1.4.1. Vartotojų aibė, tipai ir savybės

Didžioji būsto vartotojų-savininkų dalis prioritetus atiduoda pigios statybos, pigios eksploatacijos patogiam būstui, mažiau kreipdami dėmesį į komfortabilumą ir dažnai neatsižvelgdami į kokybę bei medžiagų ekologiškumą. Tai sąlygoja nepakankamos būsto savininkų pajamos. Ši vartotojų dalis pasmerkia save pastoviam būsto rekonstravimui.

Mažesnė dalis vartotojų-savininkų prioritetą atiduoda brangiam būstui. Jų apsisprendimą sąlygoja:

-  Turėjimas pakankamai pinigų
-  Noras ar būtinybė įsigyti geresnį, kokybiškesnį būstą
-  Išsilavinimas ir nuo to priklausantis žmogaus darbas.

1.4.2. Vartotojų tikslai ir problemos

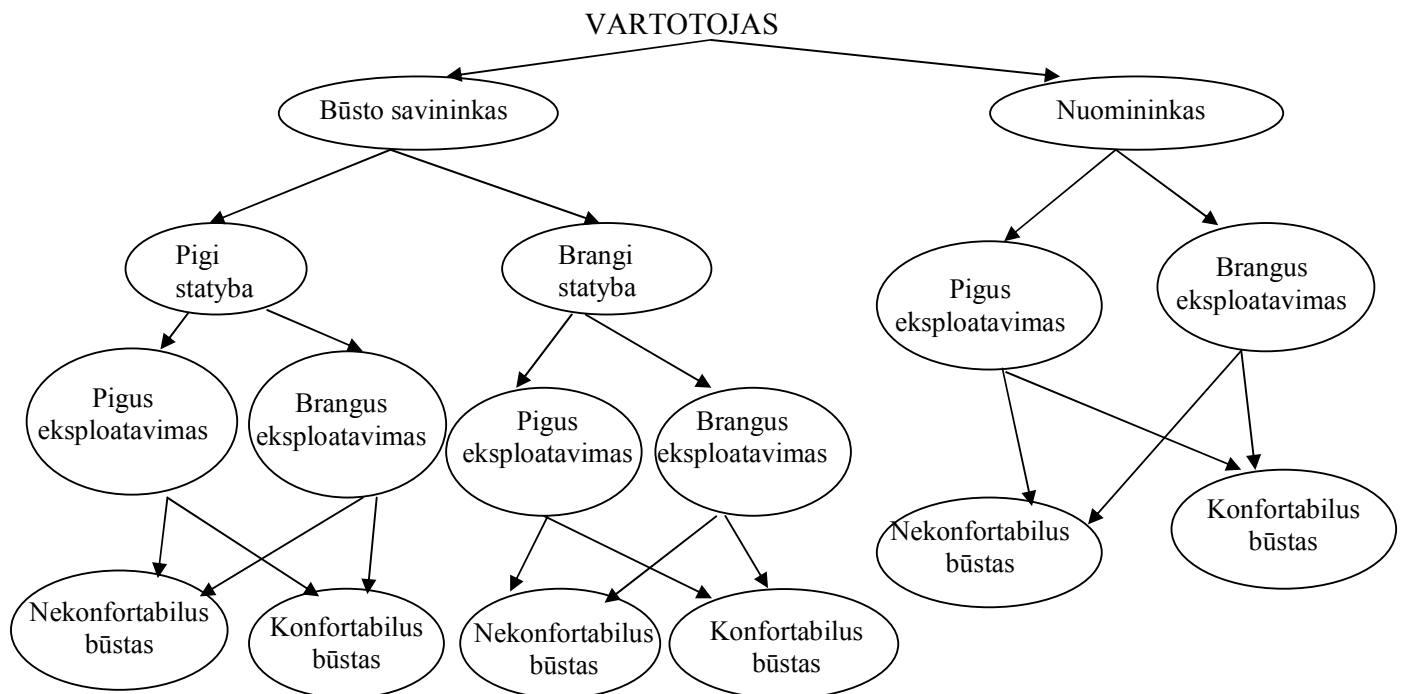
Vartotojai - nuomotojai prioritetą teikia pigiam būstui, nes tai laikinas būstas.

Neišvengiamai būsto savininkus lydi eksploataavimo problemos.

1.1 lentelė. Būsto eksploataavimo problemos

Problema	Jos sprendimo būdai
Būsto santchnikos gedimai	Sudaryti sutartis su paslaugas teikiančia firma, numatant atlikimo terminus.
Būsto elektros gedimai. Paslaugas atliekantys individualūs darbuotojai nepakankamai kompetentingi.	Sudaryti sutartis su paslaugas teikiančia firma, numatant atlikimo terminus.
Netinkamai ir nesistemiškai organizuojamas būsto būklės tikrinimas, todėl ne laiku pastebimi trūkumai ir neišvengiama didelių nuostolių įvykus avarijai.	Sudaryti sutartis su paslaugas teikiančia firma, numatant atlikimo terminus. Organizuoti inspekcinis būsto tikrinimus, saugoti ir sisteminti jų rezultatus, teikti firmoms (su kuriomis sudarytos sutartys) pasiūlymus būklei gerinti.

Taip pat vartotojo problemas galime apibrėžti grafiškai 1.2 pav.

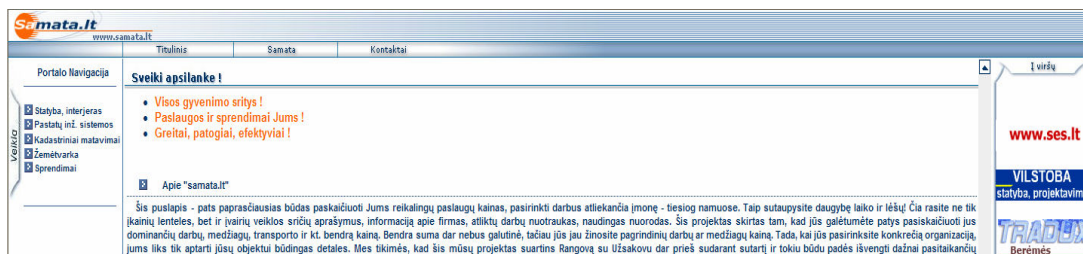


1.2 pav. Būsto portalo vartotojų poreikiai

2. Portalų metamodelio ir šablonų metodologijos, tyrimo aplinka ir architektūra.

2.1. Panašios sistemos Lietuvoje

Tinklapiu <http://www.samata.lt> privalumai ir trūkumai¹



2.1 pav. www.samata.lt tinklalapis

Šio lygio paskirtis pateikti vieningą vaizdą verslo procesų dalims. Vartotojas, dirbdamas kasdieninį savo darbą turi per dažnai persijunginėti į skirtingas programas. Organizacijos ir departamento lygio taik. programos reikalauja vartotojų vardo ir slaptažodžio, ir vartotojai netyčiomis sukuria potencialias saugumo problemas, naudodami tuos pačius slaptažodžius, užsirašant slaptažodžius ar neišsijungiant iš sistemos žinodamas kad vėli Šiame puslapyje galima paskaičiuoti reikalingų paslaugų kainas, pasirinkti darbus atliekančia įmonę. Neidentifikuotas

¹ Pilnas tinklapiu vaizdas parodytas prieduose

virtotojas gali peržiūrėti paslaugų asortimentą. Norint paskaičiuoti kainas nereikia registruotis. Šiame puslapyje trūksta paslaugų ir sąmatos išsaugojimo galimybės. Dominuoja tik kelios firmos, kurios atlieka tam tikrus darbus ir nėra galimybės naujai įmonei skelbi savo informaciją.

Tinklapiu <http://www.bukra.lt> privalumai ir trūkumai²



2.2 pav. www.bukra.lt tinklalapis

Šiame tinklapyje yra pateikta informacija apie technologijas naudojamas būsto statybai. Firms siūlo savo paslaugas projektuoti būstą pas juos nuo pradžios iki raktų. Šiame tinklapyje yra pateikti jau atlikti darbai. Pateikiami namo projektai su apytiksliais sąmatų skaičiavimais. Šiame puslapyje trūksta sąmatos išsaugojimo galimybės. Taip pat yra naudojamas šliaužiklis, kuris virtotojui sukelia papildomus veiksmus informacijos paieškai ir peržiūrai.

Tinklapiu <http://www.asa.lt> privalumai ir trūkumai³



2.3 pav. www.asa.lt tinklalapis

Šiame tinklapyje yra pateikiama informacija apie paslaugų tiekėjų paslaugas. (Apdailinės medžiagos, interjeras, komunalinės paslaugos, projektavimas ir t.t.). Čia taip pat nėra pateikta jokios duomenų bazės ar skaičiuoklės, naudojamas šliaužiklis.

² Pilnas tinklapiu vaizdas parodytas prieduose

³ Pilnas tinklapiu vaizdas parodytas prieduose

	Projektavimas	Konsultacija	Technologijos	Samata	Statybinės medž.	Apdailinės medž.	Komun. paslaugos	Inžinier. irengimai	Interjeras	Eksterjeras	Trūkumai
http://www.asa.lt	+	+	+	+	+	+	+	+	+	+	Nėra duomenų bazės. Naudojamas šliaužiklis
http://www.bukra.lt	+	+	+	+	+				+		Nėra galimybės išsaugoti duomenis. Naudojamas šliaužiklis
http://www.samata.lt				+							Dominuoja tik kelios firmos. galimybės naujai įmonei skelbi savo informaciją.

2.2. Architektūra ir galimi įgyvendinimo priemonių variantai

✚ Naudojama 3-jų lygių architektūra:

- ✚ pristatymo sluoksnis (angl. presentation layer),
- ✚ taikomųjų programų sluoksnis (angl. application layer)
- ✚ informacijos paslaugų sluoksnis (angl. information services layer).

✚ Veiklos pagal sluoksnius:

- ✚ parodymas (pristatymo sluoksnis),
- ✚ apdorojimo logika (taikomųjų programų lygis)
- ✚ duomenų servisai (organizacijos informacijos sistemos lygis).

2.2.1. Pristatymo lygis

Šio lygio paskirtis pateikti vieningą vaizdą verslo procesų dalims. Vartotojas, dirbdamas kasdieninį savo darbą turi per dažnai persijunginėti į skirtingas programas. Organizacijos ir departamento lygio taik. programos reikalauja vartotojų vardo ir slaptažodžio, ir vartotojai netyčiomis sukuria potencialias saugumo problemas, naudodami tuos pačius slaptažodžius, užsirašant slaptažodžius ar neišsijungiant iš sistemos žinodamas kad vėliau jomis reiks naudotis. Išsprendžiant šias problemas reikia vieningos bendros sąsajos daugeliui taikomųjų programų ir

sistemos valdyti kiekvieno vartotojo saugumo įrašus ir teikti priėjimą prie sistemos pagal poreikius pagrindo.

Vienas iš portalo privalumų yra integruotas plataus diapazono taik. programų įvairioms auditorijoms (vartotojams, darbdaviams, tiekimo grandinės partneriams, vykdomųjų sprendimų darytojams, bendrai visuomenei) suteikimas.

Šis turinys yra sumaišytas su statine informacija („banerinė“ reklama) ir dinamiškai sugeneruotu turiniu (reklama).

Portalo sąsajoje dažniausiai yra keletas elementų:

- ✚ „baneriais“ pabrėžiamas produktas ar paslaugos ženklas,
- ✚ produktų ar paslaugų hierarchinis meniu,
- ✚ galimybė prisijungti prie personalizuotos informacijos,
- ✚ bendra informacija ir reklama.

Pirmiausiai, kad prieitumėm prie informacijos yra prisijungiama prie portalo.

Kai vartotojas prisijungia pateikiamas dinamiškai sugeneruotas turinys:

- ✚ nuorodos į statines HTML ataskaitas,
- ✚ nuorodos į verslo intelekto priemones specialioms užklausoms ir analizei,
- ✚ trečių šalių naujienos perduodant pramonei būdingą informaciją,
- ✚ turinio valdymo priemonės dokumentų publikavimui ir pasidalinimui,
- ✚ priėjimas prie bendradarbiavimo įrankių tokių kaip diskusijų grupės.
- ✚ sąsaja yra sudaryta iš elementų, išdėliotų tam tikrose srityse.

Elementai gali būti įvairaus sudėtingumo nuo paprastų objektų (URL, paveikslukų) iki apletų (kalendoriai, kalkuliatoriai) ir komponentinių taik. programų (užklausos ir vizualizavimo priemonės). Dizaineriai turi nuspręsti kaip turinys bus pateikiamas naršyklėje. Naudojami metodai yra: HTML, Plug-in, appletai, portletai:

Portalo serveriai, esantys viduriniame lygyje suteikia pagrindinę aplinką generuojant turinį perduodant per portala.

Norint užtikrinti pastovią ir patikimą sistemą naudojant pasirenkamus elementus vieningoje portalo aplinkoje, kiekvienas komponentas, vadinamas portletu, turi atitikti tam tikras taisykles ir susitarimus.

Portalo serveris vykdo bazinius servigus, tokius kaip autentifikacija, priėjimo kontrolė ir taikomųjų uždavinių valdymą. Labiau specializuotos dalys, tokios kaip papildomos paieškos galimybės, naujienų grupės, hierarchinis meniu, ataskaitos ir grafikai, ir vartotojui pritaikyti servigus yra pateikiami per vartotojų taikomuosius uždavinius arba portletus.

2.2.2. Taikomųjų uždavinių serverio lygis

Dažniausiai tai portalo pagrindinė dalis. Šioje dalyje: vykdomi serverio pusės servletai, vykdomos verslo taisyklės, integruojami duomenys iš įvairių šaltinių, vykdomos pagrindinės portalo dalys (paieška ir navigacija)

Realiuose portaluose gali būti daug taikomųjų programų serverių ir gali būti daug taikomųjų programų serverių sluoksnių. Ne visi darbai gali būti atliekami vien tik šiuose serveriuose, yra išnaudojamos organizacijų informacinės sistemos.

2.2.3. Taikomųjų programų serverio uždaviniai

Taikomųjų programų serveriai valdo šiuos uždavinius:

- ✚ Duomenų iš įvairių šaltinių integravimą
- ✚ Verslo taisyklių taikomuosius uždavinius
- ✚ Turinio valdymą
- ✚ Bendradarbiavimą
- ✚ Turinio adaptavimą
- ✚ Paiešką ir skirstymą į kategorijas
- ✚ Apsaugą ir vartotojų administravimą

2.2.4. Duomenų integravimas iš įvairių šaltinių

Portalai dažnai naudojami integruotam, vieno taško priėjimui prie organizacijos įvairių informacinių sistemų. Kiekvienas taik. užd. vykdomas atskirais portletais ir veikia nepriklausomai vienas nuo kito. Vadinamas pristatymo lygio arba paviršutinė integracija. Privalumas, tokią integraciją lengva įvykdyti. Trūkumas. Neišnaudoja esamus ryšius tarp verslo procesų ir jų ryšių tarp taik. programų. Kai kurie verslo procesai priklauso nuo įvairių taik. progr. integracijos. Programos vykdomos taik. užd. serveryje gali būti jungiamoji dalis valdanti darbų sekas tarp reikiamų taik. užd. ir užtikrinanti, kad duomenys gauti vienos sistemos yra tinkamos formos ir nusiųsti į kitas taik. programas viso proceso metu. Serveryje vykdoma daug pranešimų ir tranzakcijų reikalingų užtikrinti, kad verslo procesai veiktų kaip neskaidomas vienetas.

2.2.5. Verslo taisyklių taikomieji uždaviniai

Dauguma sistemos projektuotojų išskirsto šias taisykles į vykdomuosius taik.užd. ir naudoja šias taisykles vykdant procesus, reikalingus daugeliui taik. užd.

Verslo taisyklės yra nurodymai, strategijos ir procedūros, kurios diktuoja, kaip verslas organizacijoje yra vykdomas.

2.2.6. Turinio valdymo pagrindiniai bruožai

Suteikia turinio valdymo servisus tvarkyti teksto, paveikslų, garso ir vaizdo medžiagos pateikimą visame portale. Apima priėjimo kontrolę skelbiant ir peržiūrint turinį, darbų sekas ir integraciją su el. komercijos uždaviniais.

Bendradarbiavimas

- ✚ El.paštas
- ✚ Diskusijos
- ✚ Pranešimai
- ✚ Ekspertų duomenų bazės
- ✚ Video konferencijos

Turinio adaptavimas

- ✚ Pritaikymas individualiam vartotojui, naudojant tiesioginį atgalinį ryšį iš vartotojų sukuriama domėjimosi profailai.
- ✚ Bendras filtravimas, remiantis tam tikrų vartotojų įpročiais, ieškomi panašūs vartotojai ir jiems siūlomas panašus turinys
- ✚ Pritaikymas asmeniniams poreikiams grupuojami ir analizuojami vartotojų tipai ir nustatomi produktai, kurie su didžiausia tikimybe bus parduoti.

Paieška ir skirstymas į kategorijas

Galimi du būdai organizuoti paiešką:

- ✚ Parsisiųsti turinį iš įvairių šaltinių ir kurti vieningą, centralizuotą rodyklę
- ✚ Naudoti daug paieškos sistemų, kurių kiekvienos rodyklė yra bendro turinio prieinamo vartotojui dalis. Kai gaunamas užklauso rezultatas ir sutampa su užklauso kitoms sistemoms rezultatais, sudaromi ir pateikiami vieningi rezultatai vartotojui.

Apsauga ir vartotojų administravimas

Portalų serveriai patys kontroliuoja priejimą prie turinio, portletų ir pritaikymo asmeniniams poreikiams.

Informacijos servisų sluoksnis

Šioje dalyje portalai ir kiti internetiniai uždaviniai susitinka su organizacijos informacine infrastruktūra.

Šiame sluoksnyje gali būti:

- ✚ Reliacinės duomenų bazės ERP sistemos
- ✚ CRM sistemos
- ✚ Automatizuotos pardavimų sistemos
- ✚ Dokumentų valdymo sistemos
- ✚ El. pašto sistemos
- ✚ Palikuoninės sistemos

Portalų kūrėjai turi įvertinti:

- ✚ Kaip informacija yra naudojama
- ✚ Kaip ji keičiasi per laiką

- ✚ Kaip ji gali būti prieinama ir valdoma

Verslo procesų palaikymas

Darbų sekų arba bendradarbiavimo sistemos gali eliminuoti specialius modulius struktūrizuotai informacijai paskleisti.

Apeinant darbų sekų sistemas patys vartotojai portale suranda ir parsisiunčia ataskaitas bei dokumentus, suspaudžia juos į vieną paketą ir siunčia el. paštu kitiems organizacijoje dirbantiems asmenims. Daug portalų sistemų turi darbų srautų valdymo servisus, kurie gali palaikyti verslo procesus.

Paprastesnis metodas būtų: vartotojai peržiūri turinį portale, patvirtina jį tolimesnei peržiūrai ir ši informacija persiunčiama kitiems asmenims naudojant darbų srautų sistemą.

Darbų srautai dažnai naudojami valdant informacijos-turinio pateikimą portale. Autoriai pateikia savo informaciją specifinei portalo sričiai, ji automatiškai persiunčiama atsakingam šios srities redaktoriui, kai redaktorius patvirtina, darbų srautų sistema automatiškai publikuoja informaciją portale.

Dalyvavimas sprendimų priėmimo

Organizacijų informacijos portalai paliečia daug organizacinių ir techninių organizacijos taškų, kurie priklauso centralizuotam sprendimų priėmimui. Tuo pačiu metu organizacija turi kontroliuoti infrastruktūros ir projektavimo plėtros klausimus.

Kiekviena organizacija yra skirtinga ir reikia surasti geriausią sprendimų priėmimo modelį, kuris atitiktų organizacijos tipą.

Yra siūlomi keli valdymo modeliai.

Pirmas sprendimų priėmimo valdymo modelis

Mažiausiai dėmesio kreipiamas į šiuos reikalavimus:

- ✚ Saugumo aprūpinimas
- ✚ Metaduomenų standartas
- ✚ Paieškos galimybės
- ✚ Katalogų ir sisteminimo išvystymas
- ✚ Navigacijos šablonai
- ✚ Panaudojimo klausimai

Šie servisai ir standartai yra patalpinti portale ir yra lengvai panaudojami

Antras sprendimų priėmimo valdymo modelis

- ✚ Neanalizuojama, kas yra publikuojama (tai autorių ir administratorių sritis)
- ✚ Nekreipiamas dėmesys kaip servisai yra tiekiami (turi būti palikta sistemos administratoriams ir projektuotojams)

Trečias sprendimų priėmimo valdymo modelis

Skaidoma į daugelį sluoksnių ir kiekvieno sluoksnio galimybės apribojamos sprendimo priėmimo platumu ir gyliu.

Formalus ir neformalus bendradarbiavimas

Gali būti naudojami du bendriausi bendradarbiavimo būdai diegiant portalą:

- ✚ Dokumentais pagrįstas būdas, kuriame informacija ir žinios yra užfiksuojamos, formaliai įrašomos ir sudedamos į turinio sandėlius.
- ✚ Būdas sutelkiant dėmesį į neformalų komunikavimą tarp žmonių.

Formalus bendradarbiavimas: Dokumentais pagrįstas būdas

Informacija yra saugojama per laiką, lengvai perduodama ir padalinama daugeliui žmonių, peržiūrima ir ištaisoma.

Šios charakteristikos yra reikalingos sėkmingam valdymui ir sudėtingų procesų vystymui.

Jei modulis yra gerai suprojektuotas, tai gali būti įtraukti neformalūs dokumentai, kaip el. pašto žinutės ir diskusijos.

Nauda:

- ✚ pagrindinė informacija ir žinios yra lengvai platinama organizacijoje.

Neigiamas aspektas:

- ✚ Nesvarbu kokia plati yra dokumentacija, ji gali neduoti pilno reikiamo vaizdo sprendimui priimti, nesurasti klaidų ir praleistų galimybių, ar neperteikti sunkios tiesos, kuri kieno nors nebuvo parašyta.
- ✚ Kartais reikia greito atsakymo, o ne didelės analizės.

Neformalus bendradarbiavimas

Diskusijų grupės yra sudaromos pagal temas ir jos skirtos konkrečiai su nagrinėjama problema susijusiai auditorijai, pvz. Kiekvienam kompanijoje ar projekto komandos nariams.

Neformalus bendravimas tarp diskusijų grupės narių gali padėti: spręsti įvairias technines problemas, gauti pranešimus apie kurio nors nario priimtą sprendimą.

Portalo vartotojai sukuria neformalius žinių pasidalinimo saugyklas.

Vartotojai gali surasti atsakymą greičiau tiesiogiai bendraudami su ekspertu negu naršydami ir ieškodami portale.

Viena iš portalo paskirčių suteikti profesionalius katalogus. Tačiau tokių katalogų sudarymui ir palaikymui reikia laiko ir resursų.

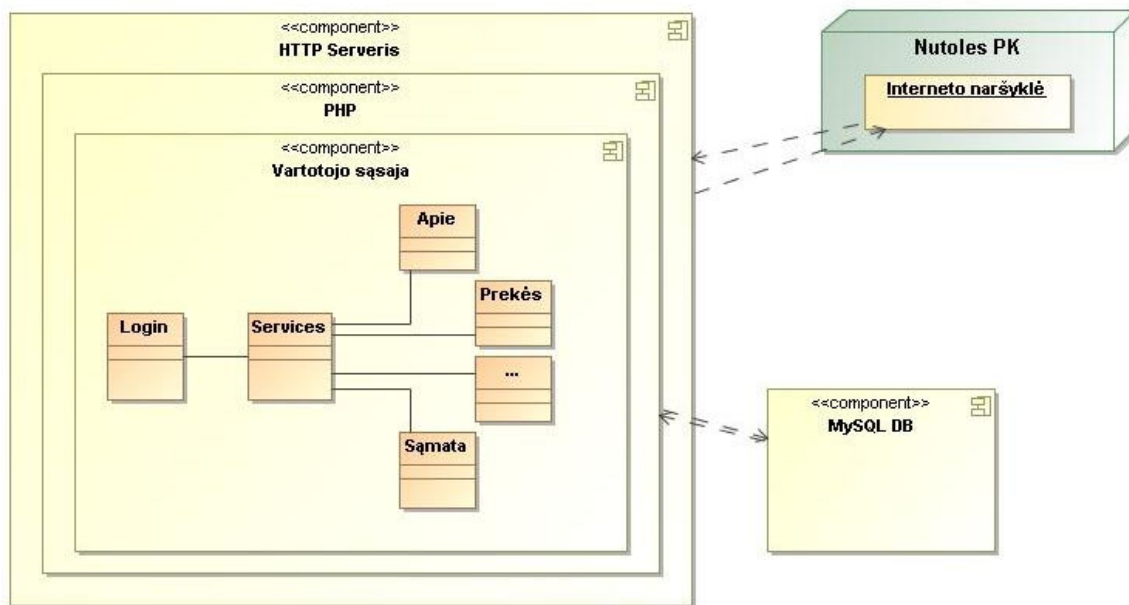
Kuriant tokius katalogus, susiduriama su nepageidaujamomis organizacinėmis problemomis.

Informacijos katalogus reikia pradėti kurti nuo pat pradžių.

- ✚ Patirtis turi būti aprašyta, sudėta į katalogus, kuriuose būtų lengva ieškoti.
- ✚ Pateikti nurodymus, šaltinius ir papildomą informaciją, bei laiku atnaujinti turinį

2.2.7. Įvadas ir architektūrinių sprendimų pagrindimas

Projekto grafinė sąsaja realizuota PHP ir MySQL kalba. Sistemoje dalyvauja keturios vartotojų grupės, kurios nutolusiam kompiuteryje mato vaizdą, o visi veiksmai ir skaičiavimai atliekami serveryje. Žemiau pateikta registruoto vartotojo sąsajos veikimo struktūra. Vartotojas iš nutolusio kompiuterio duoda užklausas, serveryje jos apdorojamos ir atlikus veiksmus grąžinamas atsakas į naršyklę.



2.4 pav. Vartotojo sąsajos veikimo struktūra

Kiekvienas registruotas sistemos vartotojas gali prisijungti prie sistemos per nutolusį kompiuterį ir, priklausomai nuo jam suteiktų teisių, valdyti pasirinktus paslaugas. Vartotojas prisijungęs prie sistemos mato grafinę sąsaja, kurios pagalba dirba su sistema. Serveryje įdiegta duomenų bazė, kurioje sukurta šiai sistemai pritaikyta duomenų bazė, kurioje saugoma visa su portalu susijusi informacija: vartotojai, vartotojų grupės, prekės, sąmatos, ir t.t. Įdiegtas Apache serveris, kuris palaiko vartotojo sąsajos veikimą.

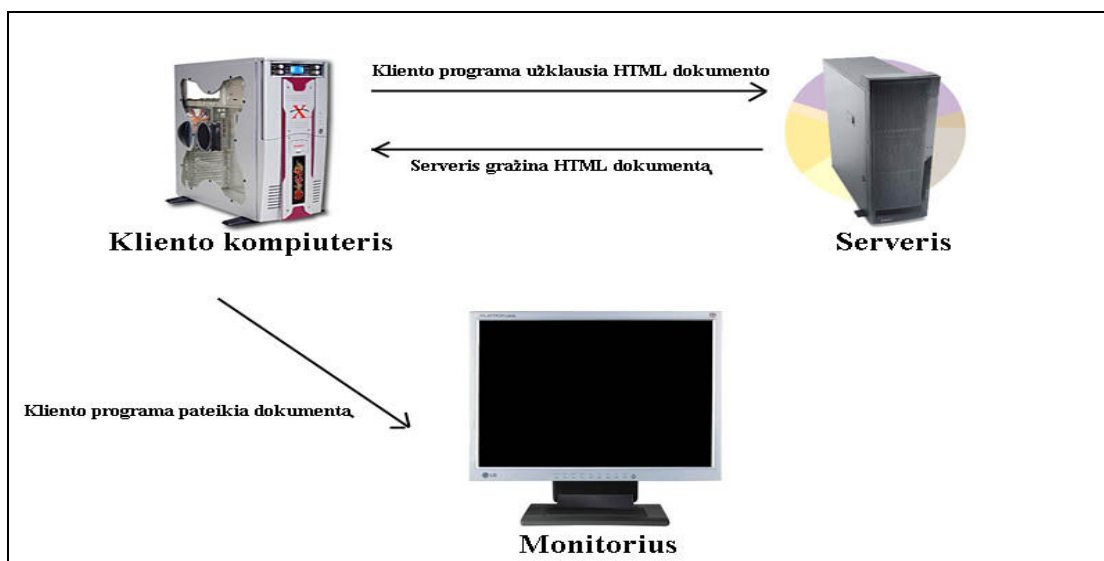
2.2.8. Sistemos dalys

WEB serveris, kuriame įdiegti Apache ir MySQL serveriai. Apache serveris, tai pasirinktas serveris, kuriame bus įdiegta kuriama vartotojo sąsaja ir kurio viduje bus atliekami visi vartotojų veiksmai. MySQL serveris, kuriame bus duomenų bazė sauganti visa su portalu susijusi informacija: vartotojai, vartotojų grupės, prekės, sąmatos, ir t.t. Vartotojo sąsaja – tai projektuojama grafinė sąsaja, kuri leis vartotojui atlikti visus, pagal jo teises leistinus veiksmus.

2.2.9. PHP programavimo kalba

Projekto įgyvendinimui pasirinkome PHP programavimo kalbą, nes ji pateikia daug priemonių turinčią platformą, skirtą žiniatinklio taikomosioms programoms. PHP siūlo platų programavimo sąsajų pasirinkimą ir galimybę susisiekti su daugybe atvirojo kodo ir komercinių duomenų bazių, kitaip tariant, ją lengva operuoti, nes specialiai buvo sukurta dirbti su dideliu technologijų asortimentu. PHP yra atviro kodo programa – sudarantis kodas prieinamas nemokamai, taip pat ji labai mobili ir tinka įvairiausiems žiniatinklio serveriams ir operacinėms sistemoms. Rinkdamiesi projektavimo platformą atsižvelgėme į pagrindinius kuriamos programos aspektus. Apsvarstėme ir įvertinome kiekvieną galimybę ir pasirinkome tinkamiausia. Kadangi PHP buvo parašyta C++ kalba, nors ir nedaug žiniatinklio taikomųjų programų buvo parašyta šia kalba, mums šis aspektas buvo svarbus. Daugelyje kalbų pažanga paaukota vardan paprastumo. PHP netrūksta pažangių priemonių, tačiau išsaugota ir galimybė nesunkiai jos mokytis ir ja naudotis. Tai vienas iš svarbiausių veiksnių, renkantis konkrečią programavimo kalbą taikomajai programai kurti, todėl atsiranda galimybės veiksmingai mokytis programavimo kalbos ir ja naudotis.

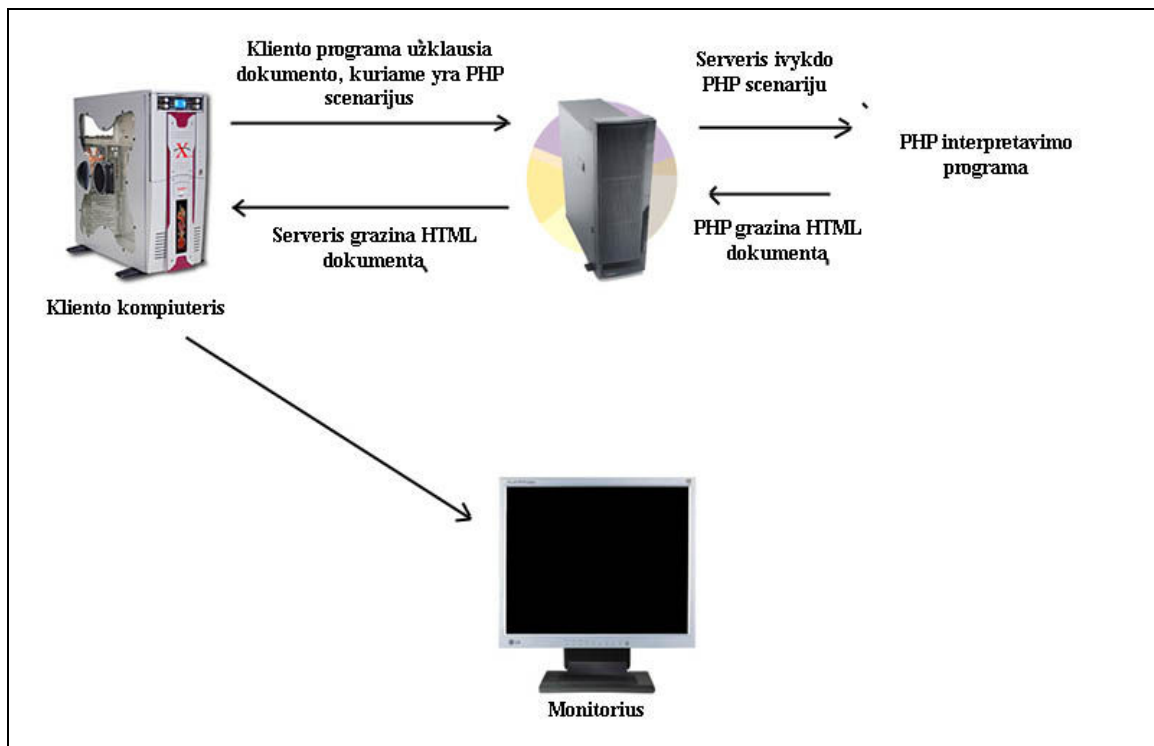
PHP veikia tik serveryje priešingai nei HTML. Kai vykdomas PHP scenarijus, jo vykdymas nepriklauso nuo žiniatinklio naršyklės, ir tik galutinis scenarijaus produktas, dažniausiai HTML dokumentas, gražinamas naršyklei. Jei naršyklė gautų neapdorotą PHP scenarijų, ji bandytų jį atvaizduoti kaip įprastą HTML dokumentą. Naršyklės negali vykdyti PHP scenarijų. HTML yra būtinas žiniatinklio programų kūrimo komponentas. PHP kodas gali būti įterptas tiesiai į HTML. Kai kliento programa serverio pareikalauja HTML dokumento, stotis jį siunčia klientui. Paveiksle 2.5 parodyta kliento HTML užklausa ir serverio atsakas.



2.5 pav. HTML dokumento užklausa

PHP scenarijaus užklausa veikia skirtingai. Prieš išsiunčiant dokumentą kliento programai, dokumentą peržiūri PHP, kuri vykdo dokumente rastą visą PHP kodą. Paveiksle

atvaizduota kliento PHP scenarijaus užklausa. Šiame 8 paveiksle PHP scenarijus gražina apdorotą HTML dokumentą.



2.6 pav. PHP scenarijaus užklausa

Šios užklauso parodo PHP ir HTML skirtumą: PHP vykdoma serveryje ir yra visavertė programavimo kalba, HTML skirta tik hiperteksto publikavimui ir vykdoma kliento programai. PHP (PHP hypertext preprocessor) tai „skriptinimo“ kalba (scripting language) kuri pačioje pradžioje buvo orientuota tik į Internetą, nors šiuo metu ją galima "drąsiai" pavadinti programavimo kalba, kadangi su ja pilnai galima programuoti ne tik internetui (web'ui). PHP skirtumas nuo JavaScript yra tai, jog PHP skriptai yra atliekami serverio pusėje o vartotojui yra gražinamas rezultatas (HTML'u ar XML'u). Rezultatą vartotojas mato savo naršyklėje. Tuo tarpu kai Java Script yra pilnai perduodamas kliento mašinai (kompiuteriui) ir atliekamas klientinėje dalyje. PHP sintaksės pagrindai yra paimti iš: C, Perl, Java, todėl programavusiems šiomis kalbomis yra lengva pradėti programuoti PHP.

2.2.10. PHP privalumai ir trūkumai

Privalumai: PHP - visiškai nemokamas, PHP - veikia įvairiose operacinėse sistemose: Win, *nix, MacOS, Solaris, HP-UX, AIX ir t.t. PHP - atviro kodo projektas, todėl jį naudoja didelė grupė žmonių, iškilusios klaidos yra greitai ištaisomos. Veikia ir ant daugelio WEB serverių: Apache, IIS, PWS, OmniHTTP, BadBlue ir t.t. Išmokti PHP programavimo pagrindų yra nesudėtinga. Pasižymi dideliu greičiu serverio pusėje, bei dirbant su duomenų bazėmis Kadangi PHP programuotojų yra

daug, daugumą jau parašytų skriptų galite rasti internete: HotScripts, FreeScripts, PHPClasses.upperdesign.com ir t.t. ir t.t.

Trūkumai: Dauguma pateikiamos naujausios medžiagos, kaip dirbti su PHP yra užsienio kalba, perskaitomi PHP „sourc'ai“

2.2.11. MySQL duomenų bazių valdymo sistema

MySQL - viena iš reliacinių duomenų bazių apdorojimo sistemų (RDBMS - relational database management system) . *Duomenų bazių sistema* yra skirta tvarkyti, rūšiuoti ir manipuluoti informaciją. Kadangi MySQL duomenų bazė leidžia įtraukti, keisti ir ištrinti informaciją iš kelių lentelių vienu metu pagal nurodytus kriterijus, ji priskiriama reliacinėms duomenų bazėms (angl. "relational"). Duomenys gali būti įvairūs – nuo paprasto sąrašo prekių iki didelio kiekio informacijos įmonių tinkle. Įrašymui, išrinkimui ir apdorojimui duomenų, saugomiems skaitmeniniuose duomenų bazėse, reikalinga duomenų bazės valdymo sistema, kokia yra kaip MySQL. Kadangi šiuolaikiniai kompiuteriai puikiai susitvarko su didelės apimties duomenimis, duomenų bazių valdymas vaidina pagrindinę rolę duomenų išrinkime.

Reliacinėje duomenų bazėje duomenys saugomi ne vienoje krūvoje, o atskirose lentelėse, kurių pagalba yra laimimas greitis ir lankstumas. Lentelės yra surišamos ryšiais, kurių dėka galima apjungti duomenis iš kelių lentelių atliekant užklausa. Kiekviena lentelė yra suskirstyta į eilutes ir stulpelius. Kiekviena eilutė atitinka vieną įrašą. Įrašas gali turėti kelių rūšių informaciją, kuri priklauso nuo stulpelių tipų ir pavadinimų. Pvz. jei duomenų bazėje esanti lentelė pavadinimu **naujienos** yra šios struktūros

Data	Pavadinimas	Tekstas
------	-------------	---------

2.7 pav. Lentelės struktūra

Tuomet kiekviena lentelėje laikoma naujiena turės tris informacijos atšakas: **data**, **pavadinimą** ir naujienos **tekstą**.

MySQL duomenų bazės sistemą, instaliuotą serveryje (UNIX, Windows ar pan.), sudaro šios dalys:

- **SQL serveris**, palaikantis MySQL ir suteikiantis priėjimą prie duomenų bazės ir joje laikomos informacijos.
- **Klientinės programos** (Client programs), naudojamos susisiekti su serveriu. Keleta iš tokių programų yra:
 - **mysql** - SQL užklausų (Queries) siuntimas serveriui;
 - **mysqldump** - duombazės trynimasis, back-up, informacijos perkėlimas į kitą SQL serverį (nebūtinai MySQL), lentelės struktūros gavimas,

- **mysqlimport** - informacijos perkėlimas iš failų į lentelę,
- **mysqladmin** - duombazės administravimas.
- **Administravimo irankiai;**
- **Programavimo interfeisas** skirtas bendravimo su SQL serveriu programų kūrimui.

MySQL veikia kliento-serverio principu. Naudojantis mysql klientu jūs rašote SQL užklausas. SQL (SQL - Structured Query Language) *struktūrizuota užklausu kalba*, yra standartinė beveik visų modernių duomenų bazių kalba. **mysql** nusiunčia jūsų parašytą užklausa SQL serveriui, kuris yra toje pačioje vietoje kur laikomi duomenų bazės duomenys. Serveris, gavęs užklausa iš klientų, atrenka informacija iš duomenų bazės pagal užklausoje nurodytus kriterijus. Vėliau serveris rezultatus persiunčia klientui, kuris juos parodo. Šiuo metu MySQL yra bene greičiausia duomenų bazių sistema. Ji lengvai instaliuojama ir administruojama. Duomenys joje gali būti manipuluojami keletu būdu: naudojantis klientinių programų komandine eilute, naršykle ar programavimo kalbomis, pavyzdžiui, PHP, C, Perl, Java, Python, ir t.t

2.2.12. Turinio valdymo sistema (TVS)

Anglų kalba turinio valdymo sistemos sutrumpinimas yra CMS (angl. *Content management system*), kas reiškia Content Management System, tačiau ši santrumpa naudojama ir daugeliui kitų, nebūtinai programinių produktų, pavadinimų sutrumpinimui. Pabandžius žodyne [<http://en.wikipedia.org/wiki/CMD>] paieškoti CMD sutrumpinimo, gauname gausybę variantų, kaip gali būti iššifruota CMD santrumpa. Akivaizdu, kad angliškoje literatūroje trumpinys CMS nėra unikalus, apibūdinantis tik turinio valdymo sistemą. Jis gali reikšti daugybę skirtingų sistemų, sąvokų, institucijų ar net mitologinių vardų, priklausomai nuo konteksto. Tai nieko nuostabaus, net literatūroje, susijusioje su informacinėmis technologijomis, turinio valdymo sistema neturi vieningo vertimo. Net tuo atveju kaip CMS naudojamas kaip TVS (turinio valdymo sistema) trumpinys, autoriaus suvokimas apie sistemą gali gerokai skirtis nuo skaitytojo. Bendru atveju tai apibūdinama kaip sistema, padedanti organizuoti ir palengvinanti susieto turinio kūrimą. Ši apibūdinimą atitinkanti turinio valdymo sistema gali būti skirta praktiškai bet kokio elektroninio turinio kūrimo ir tvarkymo sistema. Tad minint darbe CMS ar TVS (turinio valdymo sistema), bus kalbama būtent apie sistemas, skirtas interneto svetainių turinio valdymui. Turinio valdymo sistemos yra kelios:

- ✚ **Internetinės TVS** – valdyti internetinio puslapio informacija, jos kūrimą ir pristatymą.
- ✚ **Transakcijų TVS** – įmonėms dokumentams tvarkyti.
- ✚ **Publikavimo TVS** – ilgam laikui publikuoti įvairius leidinius.
- ✚ **Mokymo TVS** – nuotoliniam mokymui pritaikyti.
- ✚ **Įmonių TVS** – apima internetinės ir publikavimo turinio valdymo sistemas, gali būti skirtos įmonės vidiniams ryšiams valdyti.

Kiekviena sistema turi savo privalumų ir trūkumų.

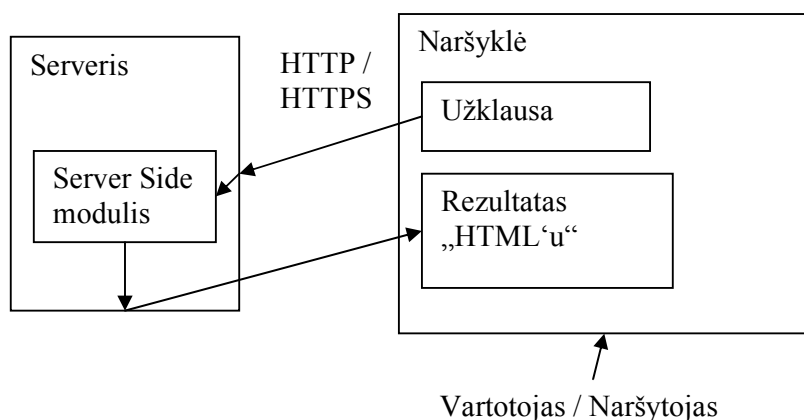
2.2.13. Turinio valdymo sistemos technologija

Kaip buvo minėta anksčiau, dauguma turinio valdymo sistemų yra sukurtos internetui pritaikytomis programavimo kalbomis. „Internetui pritaikytos programavimo kalbos“ turima omenyje, tokias programavimo kalbas kaip PHP ar ASP.

Svarbu paminėti, jog scenarijų rašymo kalbos dar smulkiau gali būti skirstomos į „ServerSide“ ir „ClientSide“. Pirmųjų veikimas yra pagrįstas serverio, kuriame talpinama interneto svetainė, įdiegtos įrangos / platformos galimybėmis, o kitos – vartotojo kompiuteryje esančia ir naudojama programine įranga. Žinoma, turinio valdymo sistemos kurti yra naudojamos „ServerSide“ scenarijų kūrimo kalbos, kadangi tokia kalba parašyta turinio valdymo sistema vienodai gerai veiks ir kompiuteriuose, kure turi įdiegtas skirtingas naršyklės, operacines sistemas ir papildomus priedus.

2.2.14. „ServerSide“ kalbos

Šioms kalboms priskiriamos tokios šiuo metu populiarios scenarijų rašymo kalbos kaip PHP (Hypertext Prie Processor – hiperteksto priešprocesorius) ir Microsoft sukurta ASP (Active Server Page – aktyvus serverio puslapis) technologija. Šios dvi kalbos, ko gero šiuo metu pačios populiariausios. Visų, šio tipo kabų veikimo principas yra panašus. Schematinis veikimo procesas pateikiamas 9 pav.



2.8 pav. Schematinis „ServerSide“ kalbos veikimo procesas

Iš pateiktos schemos matome, jog ServerSide kalbų naudojimo privalumas yra tai, jog visą „darbą atlieka“ serveris, o galutinis vartotojas mato tik suformuotus ir paruoštus peržiūrėjimui HTML (Hyper Tekst Markup Language) puslapius.

To pasekoje:

- ✚ Vartotojo kompiuteris nėra apkraunamas skaičiavimais, tam, kad būtų parodomas internetinis puslapis.
- ✚ Klaidos atsiradimo, atverčiant puslapį, tikimybės sumažėjimas.

- ✚ Vienodas puslapio atvaizdavimas visiems vartotojams.
- ✚ Nereikia papildomos programinės įrangos ar išplėtimų.

2.2.15. Turinio valdymo sistemos pasirinkimo etapai

Renkantis turinio valdymo sistemą, kaip vieną iš daugelio, yra svarbu, kad pasirinktas produktas veiktų su jūsų kompiuterine įranga, tačiau funkcinės savybės taip pat yra svarbios. Šiame skyriuje pateikiami turinio valdymo sistemos pasirinkimo kriterijai.

Nustatyti suinteresuotus asmenis ar grupes.

- ✚ Sukurti ir surikiuoti pagal prioritetą savo reikalavimus
- ✚ Sukurti veiklos scenarijų
- ✚ Patikimumas
- ✚ Išsiaiškinti sistemos saugumo lygį
- ✚ Pamėginti nors vieną prototipą kaip įmanoma anksčiau

2.2.16. Pagrindinės TVS savybės/įrankiai:

- ✚ **Integruota turinio tvarkyklė.** Šiuolaikinė TVS įgalina tvarkyti ne tik tekstą, bet ir braižyti lenteles, įterpti įvairus objektus (pvz., paveikslėlius);

- ✚ **Daugelio vartotojų palaikymas.** Įgalina kelis nepriklausomus autorius kurti turinį bei jį pasiekti visiems norintiems - visa tai galima daryti vienu metu;

- ✚ **Turinio atskyrimas nuo vaizdavimo.** Reikia, kad ta pati informacija galėtų būti panaudota neribotą skaičių kartų įvairiomis formomis (HTML, WAP, Adobe PDF ir pan.);

- ✚ **Turinio pakartotinis panaudojimas.** Labai naudinga savybė, leidžianti tą patį elementą naudoti keliose vietose (pavyzdžiui, tą pačią nuotrauką keliuose straipsniuose);

- ✚ **Metaduomenų valdymas.** Metaduomenys leidžia suteikti informacijai papildomus aprašus (autorius vardas, raktiniai žodžiai). Tai įgalina vykdyti efektyvesnę paiešką;

- ✚ **Išvystytas nuorodų mechanizmas.** Keli autoriai gali norėti susieti savo informaciją nuorodomis (pavyzdžiui, iš vieno internetinio puslapio į kito autoriaus panašų puslapį). Šiuo atveju nepakeičiamas yra Metaduomenų bei išvystyto nuorodų mechanizmo junginys;

- ✚ **Nereikia specialių techninių žinių.** Ši savybė leidžia autoriui, visiškai neturinčiam programavimo patirties, lengvai valdyti turinį;

- ✚ **Bendras sistemos suprantamumas.** Subjektyvus reikalavimas. Jums teks pakankamai daug laiko praleisti naudojantis TVS, todėl jos veikimas turi būti visiškai aiškus ir nereikalauti per daug susikaupimo.

✚ **Versijų kontrolė ir archyvavimas.** Naudingos savybės, leidžiančios atkurti dalinai ar pilnai prarastus duomenis arba grąžinti ankstesnę turinio versiją, jei naujoji pasirodys netinkama;

✚ **Saugumas.** Jei jūsų TVS valdo ne tik viešą, bet ir konfidencialią informaciją, būtina, kad pastaroji nebūtų prieinama pašaliniam asmeniui;

✚ **Integralumas.** Labai naudinga savybė, leidžianti per TVS prisijungti ne tik prie internetinio tinklapio, bet ir prie vidinio ir/ar išorinio kompanijos tinklų (intranet/extranet) - tai leidžia pasiekti visą reikiamą informaciją ir išvengti darbo su keliomis programomis vienu metu;

✚ **Statistika.** TVS gali būti galingu statistikos apdorojimo įrankiu.

Pagal savo darbo pobūdį buvo pasirinkta Internetinė TVS. Pagrindinis turinio valdymo sistemos principas – būtų nesudėtingas informacijos pateikimo įrankis, kuriuo nesunkiai būtų galima organizuoti, struktūrizuoti bei valdyti duomenis, net neturint specialiųjų programavimo žinių.

2.3. Metaduomenys

Bet koks elementas yra kažkieno apibūdinimas. Metaduomenys - duomenų tipas kur kas nors yra apibūdinama duomenimis. Kitaip sakant metaduomenys yra duomenys apie duomenis. Jei mes nuspręsimė palyginti metaduomenis realiame pasaulyje, tai jie gali būti apibūdinami daugeliu duomenų tipų, pavyzdžiui: 1. “E83BJ”, 2 “17”, 3 “Saulėta”.

Norint surasti šio duomenų tipo prasmę, reikia surasti formą, kuria yra apibūdinami duomenys. Taigi, pavyzdžiui, metaduomenų apibūdinimai, prieš tai paminėtiems duomenims gali būti:

1.1 “Pašto indeksas” – Tai yra duomenų elemento “E83BJ” apibūdinimas.

1.2 “Unikalus pašto srities identifikatorius” – Tai kitoks elemento “E83BJ” apibūdinimas

1.3 “2007 birželio 27a” – Tai dar vienas elemento “E83BJ” apibūdinimas

2 “Vidutinė temperatūra Celsijaus laipsniais” – Tai elemento “17” apibūdinimas

3 “Vakarykštis oras”. – Tai elemento “Saulėta” apibūdinimas

Duomenų elementas yra duomenys ir taip pat gali būti savo paties metaduomenys. Taigi, pavyzdžiui, “pašto indeksas” gali turėti metaduomenis: “duomenų elemento pavadinimas”, “8 simboliai, prasidedantys A – Z”, “2007 birželio 27a” gali turėti metaduomenis: “Paskutinio pakeitimo data” .

Duomenų hierarchija, metaduomenys, meta-metaduomenys ir .t.t gali tęstis be galo. Laimei mes turime pakankamai žinių, taigi mes paprastai galime jausti duomenų elementų naudojimo teisingumą, pavyzdžiui: su “pašto ženklas” metaduomenimis “8 simboliai, prasidedantys A-Z” gali būti naudojami nusakyti pašto indekso formatui.

2.3.1. Lygiai

Kaip nurodyta, yra duomenų ir metaduomenų hierarhijos. Vis dėlto betkoks duomenų elementas gali būti apibūdintas skirtingais hierarhijos lygiais, priklausomai nuo konteksto. Taigi, pavyzdžiui, kai apibūdinamas geografinis elementas “Kaunas”, tai “E83BJ” gali būti duomenys ir “pašto kodas” gali būti metaduomenys. Bet kada apibūdinamas duomenų valdymas, automatinės sistemos, kuri tvarko geografinius duomenis, tai “pašto kodas” gali būti duomenys ir “duomenų tipo pavadinimas” ir “8 simboliai, prasidedantys A-Z” gali būti metaduomenys.

Bet kuriame kontekste, metaduomenys turi būti didesnio abstrakcijos lygio nei duomenys, kurie yra apibūdinami. Taigi, ryšium su “E83BJ”, duomenų elementas “yra Kaune” yra busimas apibūdinimas vietos realiame pasaulyje, kuris turi pašto indeksą “E83BJ” ir yra tame pačiame abstrakcijos lygyje. Taigi, jeigu yra numatyta informacija apie “E83BJ” (tai mums pasako, kad tai yra Kauno pašto indeksas).

2.3.2. Metaduomenų apibrėžimai

Terminas buvo įvestas intuityviai, be jokių formalių apibrėžimų. Daugiausiai naudojamas apibūdinimas yra:

✚ Duomenys apie duomenis

Pavyzdžiui: "12345" yra duomenys ir be konteksto yra beverčiai. Kada "12345" įgauna pavadinimą (metaduomenis) – “Pašto indeksas”, galima suprasti koks tai pašto kodas yra.

Daugumai žmonių skirtumai tarp duomenų ir metaduomenų yra filosofinis, taigi paprasčiausi metaduomenų apibūdinimai būtų:

✚ Metaduomenys yra informacija apie duomenis.

✚ Metaduomenys yra informacija apie informaciją.

Yra ir sudėtingesni metaduomenų apibūdinimai:

✚ " Metaduomenys yra struktūrizuoti, išskoduoti duomenys, kurie apibūdina informacijos būsenų charakteristikas tam, kad padėtų identifikuoti, atrasti ir valdyti apibrėžtus elementus. "[1]

✚ "[Metaduomenys yra dalis] optimaliausių sustruktūrintų apibūdinimų, kurie yra laisvai prieinami lokaliuose objektuose. "[2]

Tai yra naudojama retai, nes jie yra linkę susikoncentruoti į vieną metaduomenų tikslą, kad surastų – “objektus”, “būsenas” arba išteklius ir atmesti kitus, taip pat kaip naudojami metaduomenys optimizuoti kompresijos algoritams .

Metaduomenų koncepcija išplito pasaulio sistemose kaip “duomenys apie duomenis”: lentelių pavadinimai, stulpeliai, programos ir t.t.

Tarptautiškai, metaduomenys yra “duomenys kurie apibūdina struktūrą ir organizacijos darbą, kuri naudoja informaciją, ir kuri naudoja metaduomenis apibūdinti sistemai.”.

2.3.3. Metaduomenų hierarchijos

Kai sutraukiami į hierarchinį klasifikavimą, metaduomenys dažniausiai yra vadinami ontologija arba schema. Abu terminai apibūdina "kas egzistuoja" tam tikru tikslu arba įgalina tam tikrus veiksmus. Pavyzdžiui, objektų pavadinimų bibliotekos kataloge klasifikacija tarnauja netik kaip padėjėjas randant knygas.

Metaduomenys yra dažniausiai saugomi pagrindinėje lokacijoje ir naudojami padėti organizacijoms standartizuoti jų duomenis. Ši informacija tipiškai yra saugoma metaduomenų registre.

2.3.4. Skirtumai tarp paprastų duomenų ir metaduomenų

Paprastai yra neįmanoma atskirti duomenis ir metaduomenis, nes:

- ✚ Daug kas gali būti duomenys ir metaduomenys tuo pačiu metu. Straipsnio antraštė yra kartu ir pavadinimas (metaduomenys) ir dalis jos teksto (duomenys).
- ✚ Duomenys ir metaduomenys gali apsikeisti vaidmenimis. Poema, gali būti suprantama kaip duomenys, bet jeigu joje yra daina, tai visa poema gali būti prikabinta, kaip dainos audio failas, kaip metaduomenys. Todėl pavadinimai priklauso nuo požiūrio į juos.
- ✚ Įmanoma sukurti meta-meta-...-metaduomenis. Tada, priklausomai nuo įprasto aprašymo, metaduomenys gali būti patys duomenys., įmanoma sukurti metaduomenis apie metaduomenis ir t.t. Iš pirmo žvilgsnio tai atrodo nenaudinga, tai gali būti svarbu archybuojant metaduomenis apie metaduomenis, pavyzdžiui, kad palaikyti metaduomenų kelią iš kur metaduomenys ateina, kai sujungiam du dokumentus.

2.3.5. Panaudojimas

Metaduomenys turi daug skirtingų pritaikomumų; ši dalis aprašo keletą iš svarbiausių. Metaduomenys yra naudojami pagreitinti resursų paieškai. Iš esmės paieškos užklausos, naudojančios metaduomenis, gali apsaugoti vartotojus nuo kompleksinių filtrų naudojimo. Tai šiuo metu yra labai svarbu interneto naršyklėse.

Metaduomenys taip pat gali būti susiejami rankiniu būdu su failais. Tai dažnai yra naudojama repozitoriuose FileNet arba Documentum. Kai tik dokumentai yra perverčiami į elektroninį formatą vartotojas juos pamato naudojamoje programoje.

Metaduomenys teikia papildomą informaciją apie duomenis vartotojams. Ta informacija gali būti apibūdinama ("Šios nuotraukos buvo fotografuotos vaikų trečioje klasėje. ") arba algoritmu (angl. "Checksum=139F").

Metaduomenys padeda sujungti semantinius intervalus. Pasakant kompiuteriui kaip šie duomenų įrankiai yra susiję ir kaip jų sąsajos gali būti įvertintos automatiškai, tampa įmanoma valdyti daug sudėtingesnius filtrus ir paieškos operacijas. Pavyzdžiui, jeigu paieškos sistema supranta, kad "Van Gogas" buvo "Olandų tapytojas", ji gali veikti ir su paieška "Olandų tapytojai"

ir įtraukti nuorodą į Vincento Van Gogo puslapį. Šis pagerinimas, vadinamas žinių vaizdavimu, yra specialiai aprašomas kaip dirbtinis protas.

Pagrindiniai metaduomenys yra kuriami optimizuoti praradimų kompresijos algoritmus. Pavyzdžiui, jeigu video medžiaga turi metaduomenis, kurie leidžia kompiuteriui atskirti dabartį nuo praeities, tai praeitis gali būti suspaudžiama labiau,

Kiti metaduomenų apibūdinimai gali būti naudojami automatizuoti darbų sekas. Pavyzdžiui, jeigu “protinga” programa žino duomenis ir duomenų struktūrą, tai ji gali konvertuoti duomenis ir perduoti kitai “protingai” programai gautus rezultatus automatiškai. Rezultatas, vartotojai gali sutaupyti daug copy-paste operacijų.

Metaduomenys tampa vis labiau svarbesne dalimi elektroniniuose atradimuose (angl. electronic discovery)^[3]. Programos ir failų sistemos metaduomenys, gauti iš elektroninių dokumentų ir failų gali būti svarbūs.

Metaduomenys tampa svarbūs ir pasauliniame voratinklyje (angl. www – World Wide Web), nes daugumai reikia surasti reikiamą informaciją iš didelio kiekio informacijos. Jeigu internetinis tinklalapis apie tam tikrą temą talpina žodį ar frazę tada visi puslapiai apie šią temą turėtų talpinti tą patį žodį ar frazę. Metaduomenys taip pat garantuoja pasirinkimą, taigi jeigu pavadinimas sudarytas iš dviejų žodžių ir abu žodžiai naudojami paieškoje. Pavyzdžiui, straipsnis apie sporto reikmenis gali būti pavadintas “keturi ratai” arba “4ratai” tai visvien paieška atliekama naudojantis abiem žodžiais.

2.3.6. Metaduomenų tipai

Metaduomenys gali būti klasifikuojami pagal :

- ✚ Turinį. Metaduomenys gali būti apibūdinami kaip šaltinis (pvz. failo vardas ir dydis) arba šaltinio turinys (pvz. „šiam video rodomas berniukas žaidžiantis futbolą).
- ✚ Kintamumą. Sąsajoje su visu šaltiniu metaduomenys gali būti nekintami (pvz. video „pavadinimas“ nesikeičia ėmus rodyti filmą) ir kintami („scenos apibūdinimas“ keičiasi).
- ✚ Loginė funkcija. Yra trys loginės funkcijos sluoksniai: pačioje apačioje subsimbolinis sluoksnis susidedantis iš žaliavinių/plikų duomenų, tuomet simbolinis sluoksnis su metaduomenimis apibūdinančiais žaliavinius duomenis, ir pačiame viršuje loginis sluoksnis susidedantis iš metaduomenų suteikiančių loginį pagrindą simboliniam sluoksniui.

2.3.7. Svarbūs klausimai

Norint sėkmingai plėtoti ir naudoti metaduomenis reikia atkreipti dėmesį į šiuos svarbius klausimus:

2.3.8. Metaduomenų gyvavimo ciklas

Netgi ankstyvojoje planavimo ir kūrimo stadijoje yra būtina sekti visų sukurtamų metaduomenų eigą. Neekonomiška pradėti metaduomenų prisegimą tik pasibaigus produkto kūrimo

procesui. Pvz. jeigu duomenys sukurti skaitmenine kamera nėra tučtuojau išsaugomi, juos atkurti prireiks didelių pastangų rankiniu būdu. Taigi būtina skirtingiems šaltinių kūrimo grupėms kooperuotis naudojant atitinkamus metodus ir standartus.

Manipuliavimas. Metaduomenys turi adaptuotis, jeigu jas apibūdinantis šaltinis keičiasi. Jie turi būti suliejami, jeigu du šaltiniai yra sujungiami. Šios operacijos yra retai atliekamos šiuolaikinės programinės įrangos; pvz., vaizdą redaguojančios programos neatseka metaduomenų eigos, sukurtos skaitmenine kamera.

Sunaikinimas. Gali būti naudinga išlaikyti metaduomenis netgi tuomet, kai jų šaltinis yra sunaikinamas, pvz. tekstinių dokumentų pasikeitimuose arba archivuoti failų naikinimus, atsiradusius dėl skaitmeninių teisių vadybos. Nei vieni šiandieniniai metaduomenų standartai neapima šios fazės.

2.3.9. Saugojimas

Metaduomenys gali būti saugomi arba vidiniai, tame pačiame faile kaip ir duomenys, ir išoriškai, atskirame faile. Abu būdai turi savo privalumų ir trūkumų:

Vidinis saugojimas leidžia perduoti metaduomenis kartu su duomenimis, kuriuos jie apibūdina; tokiu būdu metaduomenys yra visada po ranka ir gali būti laisvai manipuluojami. Šis metodas sukuria atsargumą ir neleidžia laikyti metaduomenis kartu.

Išorinis saugojimas leidžia surišti metaduomenis pvz. duomenų bazėje, tikslesnėms paieškoms. Nelieta jokie dubliavimo ir metaduomenys gali būti perduodami tuo pačiu metu. Vis dėlto kadangi dauguma formatų naudojami URI (angl. Uniform Resource Identifier) šiam tikslui, metodas kaip metaduomenys yra susieti su savo duomenimis turi būti atidžiai peržiūrimas. Kas nutinka jeigu, šaltinis neturi URI (vietinio kietojo disko šaltiniai arba svetainės sukurtos turinio valdymo sistema). Kas nutinka, jeigu metaduomenys gali būti įvertinami tik esant ryšiui su tinklu, ypač naudojantis RDF (angl. Resource Description Framework)? Kaip suvokti, kad šaltinis yra pakeičiamas kitu paliekant tą patį vardą, bet suteikiant skirtingą turinį?

Ir dar išlieka duomenų formato klausimas: metaduomenį saugojimas žmogui perskaitomame formate, tokiame kaip XML, gali būti naudingas, kadangi vartotojai gali suvokti ir redaguoti be specifinių įrankių. Iš kitos pusės šie formatai nėra optimizuoti saugojimo talpai, jie gali būti naudingi saugojant metaduomenis binariniame žmogui neperskaitomame formate pagreitinti perdavimą ir taupant atmintį.

2.3.10. Metaduomenų kritika

Nors dauguma kompiuterių mokslininkų mato metaduomenis kaip būdą geresniam operavimui, kai kurie kritikai ginčijasi dėl šių dalykų:

Metaduomenys yra per daug brangūs ir užimantys laiką. Argumentuojama, kad kompanijos negamins metaduomenų be reikalo, nes tai kainuoja papildomus pinigus ir privatūs vartotojai nekurs

kompleksinių metaduomenų, nes jų sukūrimas yra itin daug laiko užimantis procesas. Taigi nėra naudinga kurti formatus ir standartus, kuomet jais niekas nesinaudos.

Metaduomenys yra per daug sudėtingi. Privatūs vartotojai nekurs metaduomenų, nes egzistuojantys formatai, ypač MPEG-7, yra per daug sudėtingi. Kol nebus automatinių įrankių metaduomenų kūrimui, jie nebus kuriami.

Metaduomenys yra subjektyvūs ir priklauso nuo turinio. Greičiausiai du asmenys prisegtų skirtingus metaduomenis tam pačiam šaltiniui dėl skirtingų požiūrių. Dar gi metaduomenys gali būti neteisingai interpretuojami dėl jų priklausomybės nuo turinio. Pvz. ieškant „postmodernistinio meno“ galima praleisti tam tikrus objektus, nes šis terminas nebuvo naudojamas šių meno kūrinų sukūrimo metu, arba ieškant „nuotraukos, nufotografuotos 1 val.“ galima gauti trikdančius duomenis dėl vietinio laiko skirtumų.

Metaduomenims nėra pabaigos. Pvz. anotuojant futbolo varžybas metaduomenimis, vieni gali apibūdinti visus žaidėjus ir jų veiksmus laike ir ten sustoti. Taipogi galima apibūdinti reklamą fone ir rūbus, kuriuos dėvi žaidėjai. Kiti gi gali apibūdinti kiekvieną tribūnos sirgalių ir jų aprangą. Šie visi metaduomenys gali būti įdomūs vienai ar kitai žmonių grupei–žiūrovams, sponsoriams ar policijos greito reagavimo būriui — netgi paprastam šaltiniui įmanomų metaduomenų kiekis gali būti milžiniškas.

Metaduomenis yra nenaudingi. Dauguma šiandieninių paieškos sistemų leidžia labai efektyviai ieškoti tekstų. Kiti būdai ieškoti paveikslėlių, video ar muzikos taps vis efektyvesni ateityje. Taigi nėra reikalo ir poreikio metaduomenims.

Metaduomenų priešininkai kartais naudoja terminą metašiukšlės (angl. metacrap), siedami tai su neišspręstomis metaduomenų problemomis.

2.3.11. Tipai

Iš esmės egzistuoja dvi atskiros metaduomenų klasės: struktūriniai arba kontroliniai metaduomenys ir nukreipiamieji metaduomenys.^[4] Struktūriniai metaduomenys yra naudojami apibūdinti kompiuterių sistemoms tokioms kaip lentelės, stulpeliai ir indeksai. Nukreipiamieji metaduomenys yra naudojami padėti žmonėms rasti specifinius dalykus ir yra dažniausiai apibūdinami kaip rinkinys raktinių žodžių natūralia kalba.

2.3.12. Sąsajiniai duomenų bazių metaduomenys

Kiekviena sąsajinių duomenų bazių sistema turi savo mechanizmą saugoti metaduomenis. Sąsajinių duomenų bazių metaduomenys apima: visų duomenų bazės lentelių lentelės, jų vardai, dydžiai ir eilučių skaičius kiekvienoje lentelėje. Kiekvienos duomenų bazės stulpelių lentelės, kokiose lentelėse yra naudojami ir duomenų, saugomų juose tipas.

Duomenų bazės terminologijoje šis metaduomenų rinkinys yra vadinamas katalogu. SQL standartas specifikuoja unifikuotas priemones valdyti katalogą, vadinamą Informacijos Schema, bet

ne visos duomenų bazės ją taiko, net jei taiko kitus SQL standarto aspektus. Pvz. ieškant specifinių duomenų bazių valdymo pavyzdžių.

2.3.13. Duomenų sandėliavimo metaduomenys

Duomenų sandėliavimo metaduomenų sistemos yra kartais atskiriamos į dvi sekcijas:

- ✚ **galinio kambario metaduomenys**, kurie yra naudojami ištraukimo, transformavimo ir pakrovimo funkcijoms gauti.
- ✚ **priekinio kambario metaduomenys**, kurie yra naudojami įvardinti ekranus ir sukurti pranešimą.

Kimball ^[5] nurodo sekančius metaduomenų tipus duomenų sandėliavime (taip pat^[6]):

- ✚ šaltinio sistemos metaduomenys
- ✚ šaltinio specifikacijos, tokios kaip saugyklos ir šaltinio schemos
- ✚ šaltinio apibūdinanti informacija, tokia kaip savininko informacija, atnaujinimo dažnis, teisiniai suvaržymai ir valdymo metodai
- ✚ proceso informacija, tokia kaip darbo grafikai ir kilmės kodai.
- ✚ duomenų susstatymo metaduomenys
- ✚ duomenų įgijimo informacija, tokia kaip duomenų perdavimo aprašas (grafikas) ir rezultatai, bei failų naudojimas
- ✚ dimensijos lentelės valdymas, toks kaip dimensijų apibūdinimas ir surogatinio rakto paskyrimas
- ✚ transformacija ir sukaupimas, tokie kaip duomenų padidinimas ir pažymėjimas, DBMS pakrovimo scenarijai ir agregatų apibūdinimai.
- ✚ Auditas, darbo registras ir dokumentacija, tokia kaip duomenų linijos įrašai, duomenų transformavimo registras

DBMS metaduomenys, tokie kaip:

- ✚ DBMS sistemos lentelių turinys
- ✚ Procesinės užuominos

Michael Bracket apibūdina metaduomenis (juos jis vadina „duomenų šaltinio duomenys“) kaip „bet kokie duomenys apie organizacijos duomenų šaltinį“.^[7] Adrienne Tannenbaum apibūdina metaduomenis kaip „detalius atskirų duomenų apibūdinimus; patalpintų atskirų duomenų formatus ir charakteristika: atskiri atvejai ir vertės, priklausomas nuo metaduomenų vaidmens respondentas“.^[8] Šie apibūdinimai yra charakteristikos „duomenų apie duomenis“ apibrėžimo.

2.3.14. Verslo informacijos metaduomenys

Verslo informacija yra procesas, kuris analizuoja didelius kiekius kolektyvinių duomenų, dažniausiai saugomų didelėse duomenų bazėse, tokiose kaip Duomenų sandėliai, sekant verslo eigą, nustatant modelius ir tendencijas, ir padedant kompanijos darbuotojams dėl geresnių sprendimų.

Verslo informacija metaduomenys apibūdina, kaip duomenys yra užklaunami, filtruojami, analizuojami ir pateikiami verslo informacijos programinės įrangos reikmenimis, tokiais kaip ataskaitų įrankiai, OLAP įrankiai ir duomenų gavimo įrankiai.

Pavyzdžiai:

- ✚ OLAP metaduomenys: dimensijų, kubų, matavimų(metrinių), hierarchijų, lygių ir veikimo būdų apibūdinimas ir struktūros.
- ✚ Ataskaitos metaduomenys: pranešimų, diagramų, veikimo būdų, duomenų rinkinių, filtrų, variacijų ir išraiškos apibūdinimas ir struktūros.
- ✚ Duomenų gavimo metaduomenys: duomenų rinkinių, algoritmų ir veikimo būdų apibūdinimas ir struktūros.

Verslo informacijos metaduomenys gali būti naudojami suvokti, kaip skaičiuojamos korporacijų finansinės ataskaitos, pateikiamos Wall Street'ui, kaip biudžeto pajamos, išlaidos ir pelnas yra sukaupti iš atskirų pardavimų perdavimų, saugomų duomenų sandėlyje. Gilus verslo informacijos metaduomenų suvokimas yra reikalingas išspręsti kompleksines problemas, tokias kaip laikymasis korporacijos valdymo standartų, tokių kaip Sarbanes Oxley (sutr. SOX) arba Basel II.

2.3.15. Bendriniai IT metaduomenys

Priešingai nei David Marco, kitas metaduomenų teoretikas apibrėžia metaduomenis kaip „visus fizinius duomenis ir žinias organizacijos viduje ir išorėje, įskaitant informaciją apie fizinius duomenis, techninius ir verslo procesus, taisykles ir duomenų suvaržymus, ir duomenų struktūrą, naudojamą kompanijos“^[9]. Kiti dar įtraukia tinklo paslaugas, sistemas ir sandūras. Iš esmės visa Zachman sistema gali būti pristatomi kaip metaduomenys..^[10]

Pažymėtina, kad tokie apibūdinimai praplečia metaduomenų sampratą, apimančią didžiąją dalį arba visus duomenis, reikalingus informacijos valdymo sistemos galimybėms. Šia prasme metaduomenų koncepcija turi žymų sutapimą su ITIL koncepcija ir duomenų bazės valdymo konfigūracija (sutr. CMDB), ir taip pat su sąvokomis Įmonės architektūra ir IT portfelio valdymu.

Šis metaduomenų apibrėžimas turi precedentą. Trečios kartos kolektyviniai informacijos šaltinio produktai (tokie kaip pastaruoju metu sulieti į CA privalumų liniją) ne tik saugo informaciją apie duomenų apibūdinimus (COBOL kopijavimo knygos, DBSM schema), bet taip pat ir apie programas aptarnaujančias šiuos duomenis ir darbo kontrolės kalbą, ir priklausomybes nuo darbo infrastruktūros paketus. Šie produktai (kai kurie iš jų yra vis dar gamyboje) gali suteikti pilną poveikslą pagrindinės kompiuterinės įrangos komplekto, palaikančio tikslų analizės poveikį reikalingą ITIL paremtus procesus, tokius kaip incidento ir pokyčio valdymą. ITIL galinis katalogas apima duomenų valdymo paketą, kuris apibūdina šių metaduomenų produktų vaidmenį pagrindinės kompiuterinės įrangos komplektui, nustatant CMDB kaip paskirstytą kompiuterinį ekvivalentą.

CMDB pardavimas vis tik iš esmės nepraplėtė savo galimybių įtraukti duomenų apibūdinimus, ir metaduomenų sprendimai yra vis dar prieinami distribuciniam pasaulyje. Nustatant kiekvienam tinkamus vaidmenis ir galimybes yra iššūkis didelėms IT organizacijoms reikalaujančioms dvigubo serviso.

Kadangi metaduomenys yra sklindantys, centralizuoti bandymai juos atsekti reikalauja dėmesį susikoncentruoti ties labiausiai įtaką darančiomis lėšomis. Kompanijos lėšos gali būti tik maža dalimi skiriamos nuo viso IT portfelio.

Kai kurie praktikai sėkmingai įgyvendina IT metaduomenų valdymą naudodamiesi Dublino pagrindo metamodeliu.^[11]

2.3.16. IT metaduomenų valdymo produktai

Pirminės kartos duomenų žinyno/metaduomenų informacijos kaupyklos įrankiai būtų pastarieji, palaikantys DBMS specifiką, tokią kaip IDMS's IDD (integruotų duomenų žinynas), IMS duomenų žinyną ir Adabas.

Antroji karta apimtų ASG's DATAMANAGER (duomenų valdymo) produktas, galintis palaikyti daug skirtingų failų ir DBMS tipus.

Trečios kartos informacijos šaltinių produktai greitai tapo populiarūs X-ajame dešimtmetyje kartu su pakilimu plataus RDBMS variklio naudojimu, tokiu kaip IBM's DB2.

Ketvirtos kartos produktai susieja informacijos šaltinį su daugiau išrinkimo, transformavimo, pakrovimo įrankių ir gali būti susieti su architektūriniais modeliavimo įrankiais. Pvz. Adaptuojantis metaduomenis valdytojas iš Adaptive, Informacinė bibliotekinė metaduomenų integravimo sistema ir Trous Technologies Metis Server produktas.

2.3.17. Failų sistemos metaduomenys

Beveik visos failų sistemos saugo metaduomenis apie failus nesuvienytai. Kai kurios sistemos saugo metaduomenis katalogų įvestyse; kiti – specializuotose struktūrose, arba net failo varde. Metaduomenys gali varijuoti nuo paprastų laiko ženklimų, bitų modų ir kitos specialios paskirties informacijos, naudojamos įgyvendinime, iki ikonų ir beteksčių komentarų, bei sutartinių vertei būdingų porų.

Kalbant apie sudėtingesnius ir begalinius metaduomenis, tampa naudinga ieškoti failų, paremtų metaduomenų turiniu. Unix paieškos naudingumas būtų ankstyvasis pavyzdys, nors neefektyvus skenuojant šimtus tūkstančių failų moderniose kompiuterinėse sistemose. Pastaroji Apple Computer versija su jos Mac OS X operacine sistema (angl. tiger) palaiko katalogizavimą ir metaduomenų failų paiešką per įrangą, žinomą kaip Spotlight. Microsoft, vystant tolygų funkcionalumą kartu su momentinės paieškos sistema Windows Vista operacinėje sistemoje. Linux įdiegia failų metaduomenis naudodamasis praplėstais failų priedais.

2.3.18. Vaizdų metaduomenys

Pavyzdžiai vaizdo failų, turinčių metaduomenis apima tarpusavyje besikeičiančių failų vaizdais formatus ir prisegtų vaizdų failų formatus.

Turint metaduomenis apie vaizdus įtvirtintus TIFF arba EXIF failuose yra vienas iš būdų reikalaujantis papildomų duomenų apie vaizdą. Vaizdo metaduomenys yra gaunami per prisegimus. Prisegant paveikslėlius su objektais, susijusiomis emocijomis ir kitomis apibūdinančiomis frazėmis padeda interneto vartotojams rasti lengvai paveikslėlius nei ieškant per visą paveikslėlių kolekciją. Vaizdo prikabinimo paslaugos pavyzdys yra Flickr, kur vartotojai įkelia vaizdus ir tuomet apibūdina turinį. Kiti tinklapio vartotojai gali tada ieškoti tų prisegimų. Flickr naudojasi folksonomija: betekstė klaviatūrinė sistema, kurioje bendruomenė apibrėžia žodyną per naudojimąsi, o ne per kontroliuojamą žodyną.

Skaitmeninė fotografija vis daugiau naudojasi metaduomenų prisegimais. Fotografai besinaudojantys Camera RAW failų formatais gali naudotis tokiais pritaikymais kaip Adobe Bridge arba Apple Computer's Aperture dirbti su kameros metaduomenimis vėlesniame apdirbime. Vartotojai taip pat gali prisegti nuotraukas organizavimo tikslais naudojantis Adobe's praplėčiamos metaduomenų platformos kalba.

2.3.19. Programų metaduomenys

Metaduomenys yra paprastai naudojami apibūdinti kontroliuojamus duomenis, naudojamus programinės įrangos kūrime, kas yra labiau abstraktu ir konfigūruojama. Dauguma vykdomųjų failų formatų apima tai, kas gali būti vadinama „metaduomenimis“ ir kas specifikuoja tam tikras, dažniausiai konfigūruojamas, funkcionuojamas veikimo laiko charakteristikas. Vis tik yra sunku, jei netgi neįmanoma tiksliai atskirti „metaduomenis“ nuo išsaugomų programų programavimo kūrimo bendrinių aspektų; jeigu mašina juos perskaito ir pagal juos veikia, tai yra apskaičiavimo instrukcija ir priesaga „meta“ turi su tuo mažai ryšio.

Java programavimo kalboje klasės failo formatas turi savyje metaduomenis naudojamus kompiliatoriaus ir Java virtualioje mašinoje (angl. Java virtual machine) dinamiškai susieti klases ir palaikyti refleksiją. J2SE 5.0 Javos versija turi metaduomenų įtaisą, leidžiantį papildomas anotacijas naudojamas vystymo priemonių.

MS-DOS COM failo formatas neturi savyje metaduomenų, kuomet juos turi EXE failas ir Windows PE formatai. Šie metaduomenys gali apimti kompaniją, kuri išleido programą, datą, kuomet buvo sukurta programa, versijos numerį ir kitą informaciją.

Microsoft.NET vykdomajame formate extra metaduomenys yra įtraukiami, kad leistų vyksti refleksijai paleidimo metu.

2.3.20. Egzistuojantys programinės įrangos metaduomenys

Objektų valdymo grupė (sutr. OMG) apibūdino metaduomenų formatą kaip skirtą reprezentavimui visų egzistuojančių pritaikymų tikslu kurti, modernizuoti ir užtikrinti programinę įrangą. Šis patikslinimas, vadinamas OMG žinių atradimo metamodeliu (sutr. KDM) yra OMG pagrindas "atgaliniam modeliavimui". KDM yra bendrinis nuo kalbos nepriklausomas tarpinis pristatymas, kuris suteikia integruotą požiūrį viso kompanijos pritaikomumo, įskaitant elgseną (programos tėkmę), duomenis, ir struktūrą. Vienas iš KDM taikomumų yra verslo taisyklių kūrimas.

Žinių atradimo metamodelis apima tiksliai susmulkintą žemo lygio pristatymą (vadinamą „micro KDM“), tinkamą atlikti statinę programų analizę.

2.3.21. Dokumentų metaduomenys

Daugybė programų, kuriančių dokumentus, įskaitant Microsoft SharePoint, Microsoft Word ir kitus Microsoft Office produktus, išsaugo metaduomenis kartu su dokumentų failais. Šie metaduomenys gali apimti asmens, sukūrusio failą, vardą (gautą iš operacinės sistemos), vardą žmogaus, paskiausiai atnaujinusio failą, kiek kartų failas buvo atspausdintas, ir netgi netgi kiek pakeitimų buvo atlikta faile. Kita išsaugota medžiaga, tokia kaip ištrintas tekstas (išsaugotas tuo atveju, jei yra komanda atgaminti ištrintą), dokumentų komentarai ir panašiai, yra taipogi vertinami kaip „metaduomenys“ ir neapdairus šios medžiagos įtraukimas į distribucinius failus kartais priveda prie nemalonių demaskavimų.

2.3.22. Metamodeliai

Metaduomenų modeliai yra vadinami metamodeliais. Modeliu paremtoje inžinerijoje (sutr. MDA), modelis turi atitikti duotą metamodelį. Pagal MDA (angl. Model Drive Architecture) gidą, metamodelis yra modelis ir kiekvienas metamodelis atitinka duotą metamodelį. Metamodeliavimas leidžia tikslų ir judrų modelių ir metamodelių automatinį vyksmą.

Objektų valdymo grupė (OMG) apibrėžia 4 metamodeliavimo sluoksnius. Kiekvienas modeliavimo sluoksnis yra apibrėžiamas ir tikrinamas sekančio sluoksnio:

- ✚ M0: atskiras objektas, duomenų eilutė, įrašas -> "John Smith"
- ✚ M1: modelis, schema -> "Klientas" UML Class or database Table
- ✚ M2: metamodelis -> Unifikuota modeliavimo kalba (UML), Bendras sandėliavimo metamodelis (CWM), Žinių atradimo metamodelis (KDM)
- ✚ M3: meta-metamodel -> Meta objektų įtaisa (MOF)

2.3.23. "Keisti" metaduomenys

Kadangi metaduomenys yra taip pat ir duomenys, yra įmanoma turėti metaduomenų metaduomenis-, „meta-metaduomenys“. Mašinų sukurti meta-metaduomenys, tokie kaip atgalinis indeksas sukurtas betekstės paieškos sistemos, vis tik nėra laikomi metaduomenimis iš esmės.

Metaduomenys įtvirtinti su turiniu yra vadinami pritvirtinti metaduomenys. Duomenų informacija tipiškai išsaugo metaduomenis atskirtus nuo duomenų.

2.3.24. Skaitmeninės bibliotekos metaduomenys

Yra trys metaduomenų kategorijos, dažniausiai naudojamos apibūdinti objektus skaitmeninėje bibliotekoje. ^{[12][13]}:

- ✚ **apibūdinantys** – informacija, apibūdinanti objekto intelektualinį turinį, tokį kaip katalogizavimo įrašai, atrandantys pagalbas ar panašias schemas. Tai tipiškai naudojama bibliografiniams tikslams, paieškai ir atitaisymams.
- ✚ **struktūriniai** – informacija, kuri suriša kiekvieną objektą prie kitų sudaryti loginius vienetus (pvz. Informacija, kuri susieja individualius vaizdus iš knygos puslapių su kitais, sudarančiais knygą)
- ✚ **administruojantys** – informacija, naudojama valdyti objektą ir kontroliuoti prieigą prie jo. Tai gali apimti informaciją, kaip objektas buvo nuskaitytas, jo saugojimo formatą, kopijavimo teises ir licenzijos informaciją, ir informaciją reikalingą ilgalaikiam skaitmeninių objektų saugojimui.

2.3.25. Geoerdviniai metaduomenys

Metaduomenys, apibūdinantys geografinius objektus (tokius, kaip duomenų rinkiniai, žemėlapiai, bruožai ar įprasti dokumentai su geoerdviniais komponentais) datuojami nuo mažiausiai 1994 metų (pagal MIT Library page on FGDC Metadata ^[14]). Ši metaduomenų klasė yra plačiau apibūdinama Geospatial metadata ^[15] puslapyje.

Dažniausiai metamodeliavimas yra analizė, konstarvimas ir vystymas rėmų, taisyklių, suvaržymų, modelių ir teorijų, tinkamų ir naudingų modeliavimui nustatytos klasės problemoms. Ši koncepcijos definicija yra susieta su paminėjimu terminų tokių, kaip meta ir modeliavimas.

Dėl metamodeliavimo meta pobūdžio šis veiksmas ir metamodeliai yra esmė meta mokslo domėjimosi, metafilosofijos ir meta teorijos bei sistematikos, taipogi susiję su meta sąmone.

Iš kompiuterinės perspektyvos ši koncepcija yra naudojama matematikoje ir yra praktiškai naudojama kompiuterių moksle ir kompiuterių inžinerijoje/programinės įrangos inžinerijoje

Kompiuterių moksle ir su tuo susijusiose disciplinose metamodeliavimas – koncepcijų kolekcijos konstravimas (daiktų, terminų ir t.t.) tam tikrame domene. Modelis yra fenomeno abstrakcija realiame pasaulyje, ir metamodelis yra dar kita abstrakcija, apšviečianti paties modelio savybes. Šis modelis, yra sakoma, susijęs su metamodeliu kaip programa susijusi su programavimo kalbos gramatika, kuria jis parašytas. Metamodeliai yra dažniausiai naudojami:

- ✚ Kaip schema semantiniams duomenims, kuriais reikia apsikeisti arba išsaugoti
- ✚ Kaip kalba, palaikanti tam tikrą metodą arba procesą
- ✚ Kaip kalba, leidžianti išreikšti papildomą egzistuojančios informacijos semantiką.

2.4. MDE technologija

2.4.1. Apibūdinimas

Sekanti diskusija gali būti aptinkama kaip detalus metamodeliavimo technikos taikymas, susijęs su modeliu paremta inžinerija. Duomenų inžinerijoje ir programinės įrangos inžinerijoje modeliu naudojimas yra vis labiau rekomenduojamas. Tai turi būti priešpastatoma klasikinei, kodu paremtai, vystymo technologijai. Modelis visuomet yra sietinas su unikaliu metamodeliu. Viena iš šiuolaikinių aktyvių modeliu paremtos inžinerijos šakų yra metodas, pavadintas modeliu paremtas kūrimas, pasiūlytas OMG. Šis metodas yra paremtas kalbos utilizavimu parašyti metamodelius vadinamus Meta Object Facility arba MOF. Tipiški metamodeliai pasiūlyti OMG yra UML, SPEM ar CWM. ISO taipogi turi išleidę standartinius metamodelius ISO/IEC 24744.

2.4.2. Metamodelių tipai

Programinės įrangos inžinerijoje keletas modelių tipų ir su jais susiję modeliavimo veiksmai gali būti apibūdinami kaip:

- ✚ Metaduomenų modeliavimas (angl. Meta Data Model)
- ✚ Metaproceso modeliavimas (angl. MetaProcess Model)
- ✚ Vykdomasis metamodeliavimas (apimantis abu viršutinius)

Panašių metamodelių biblioteka yra vadinama metamodeliu zoologijos sodu popieriuje. Keletas metamodelių zoologijos sodų gali būti randami AtlanticZoo^[16]. Kai kurie yra išreikšti ECore. Kiti yra parašyti MOF 1.4 - XMI 1.2. Metamodeliai išreikšti UML-XMI1.2 gali būti įkeliami į Poseidon UML, ar UML CASE įrankius.

2.4.3. Modelių transformavimas

Svarbus postūmis modeliu paremtoje inžinerijoje yra sisteminis naudojimas modelį transformuojančią kalbą. OMG pasiūlė standartą vadinamą QVT užklausoms, peržiūroms ir transformacijoms. QVT yra paremtas metaobjektų įtaisais arba MOF. Tarp daugelio kitų modelių transformuojančių kalbų pavyzdžiai standartų taikymo būtų AndroMDA, VIATRA, Tefkat.

- ✚ QVT (Transformacijos Modelis). MOF/QVT atveju, modelio transformacija yra taip pat modelis. Tai reiškia, kad transformavimo kalba turi būti apibrėžta tikslaus modelio. Pavyzdys modeliu paremtos transformavimo kalbos, paremtos tikslu metamodeliu yra ATL.

2.4.4. Ryšys su ontologija

Metamodeliai yra tampriai susiję su ontologija. Abu yra dažnai naudojami apibūdinti ir analizuoti santykius tarp koncepcijų [Söderström2002].

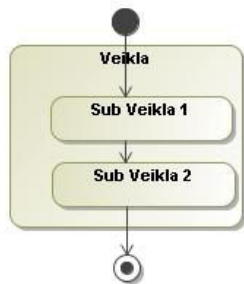
Ontologijos išreiškia kažką reikšmingo specifinėje erdvėje arba domene diskurso utilizuojant gramatiką žodynėlio naudojimui. Gramatika patikslina tai, kas yra gerai išreikštas teiginys, sprendimas, užklausa ir t.t (formalus apribojimas) ir kaip ontologija kontroliuojamo žodyno terminai gali būti naudojami kartu. [Metamodel-b]

Metamodeliavimas gali būti laikomas tiksliu apibūdinimu (konstrukcijų ar taisyklių) kaip yra sudarytas domeno specifinis modelis. Konkrečiai, tai formalizuoja sistemos specifiką. Tipiškai metamodeliai yra – ir visada turi laikytis - griežtų taisyklių rinkinio. [Metamodel-a]. “Galiojantis modelis yra ontologija, bet ne visos ontologijos yra tiksliai sumodeliuotos kaip metamodeliai” [Metamodel-b].

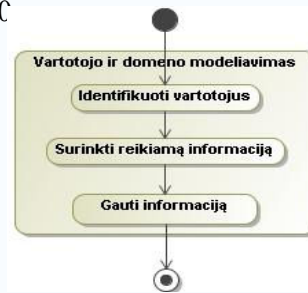
2.4.5. Metamodeliavimo technikos

Nuoseklūs veiksmai

Nuoseklūs veiksmai yra veiksmai, kurie turi būti atliekami apibrėžta tvarka. Veiksmai yra susieti rodykle, numanant, kad jie turi būti atliekami tokia tvarka. Abu veiksmai ir poveiksmiai gali būti modeliuojami nuosekliajū būdu. Paveikslėlyje 2.9 veiksmų diagrama yra iliustruota vienu veiksmu ir dviem nuosekliais poveiksmiais. Tam tikra rūšis nuoseklių poveiksmių yra pradžios ir pabaigos stadijos, taip pat iliustruojama paveikslėlyje 2.10



2.9 pav. Nuoseklūs veiksmai

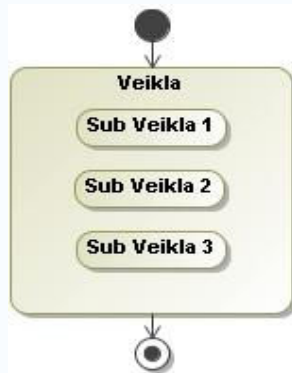


2.10 pav. Pavyzdys nuosekliųjų veiksmų

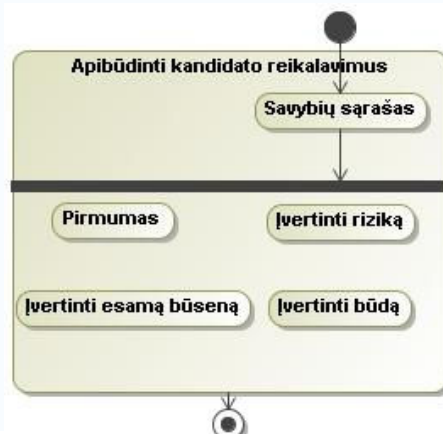
Paveikslėlyje 11 yra pavyzdys iliustruotas iš praktikos. Pavyzdys yra paimtas iš reikalavimų darbo eigai UML paremtoje tinklo inžinerijoje. Pagrindinis veiksmas, vartotojo ir domeno modeliavimas, susideda iš trijų veiksmų, privalomų atlikti apibrėžta tvarka.

Veiksmai be tikslios tvarkos

Veiksmai be tikslios tvarkos yra naudojami, kuomet veiksmo poveiksmiai neturi apibrėžtos tvarkos, kuria jie turi būti atliekami. Be tikslios tvarkos gali būti tik poveiksmiai. Veiksmai be tikslios tvarkos yra pristatomi kaip poveiksmiai be transmisijos veiksmė, kaip parodyta paveikslėlyje 2.11.



2.11 pav. Veiksmai be tikslios tvarkos

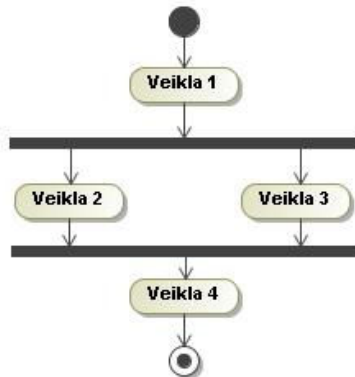


2.12 pav. veiksmų be tikslios tvarkos pavyzdys

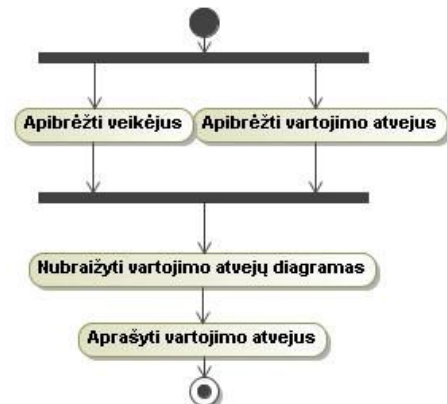
Tam tikrais specifiniais atvejais veiksmas egzistuoja kaip nuosekli ir be tikslios tvarkos veikla. Šio modeliavimo klausimo sprendimas yra padalinti pagrindinį veiksmą skirtingomis dalimis. Paveikslėlyje 2.12 iliustruotas pavyzdys, įrodantis būtinumą gebėti modeliuoti veiksmus be tikslios tvarkos. Pavyzdys yra paimtas iš darbo eigos reikalavimų analizės unifikuoto proceso. Pagrindinis veiksmas, „apibūdinti kandidato reikalavimus“, yra padalintas į dvi dalis. Pirmasis veiksmas yra nuoseklusis. Antrasis susideda iš keturių veiksmų, kuriems nereikia jokio nuoseklumo, kad būtų teisingai atlikti.

Vykstantys tuo pačiu metu veiksmai

Veiksmai gali vykti tuo pačiu metu. Tai yra tvarkoma atsišakojant ir jungiant. Piešiant veiksmų paralelę diagramoje, susijusioje su sinchronizacijos ruožu, galima iššakoti keletą veiksmų. Vėliau prie tų iššakotų veiksmų galima prijungti naudojantis tuo pačiu sinchronizacijos ruožu. Abu veiksmi ir poveiksmai vyksta tuo pačiu metu. Pavyzdyje 2.13, veiksmas 2 ir veiksmas 3 yra tuo pačiu metu vykstantys veiksmi.



2.13 pav. Tuo pačiu metu vykstantys veiksmi



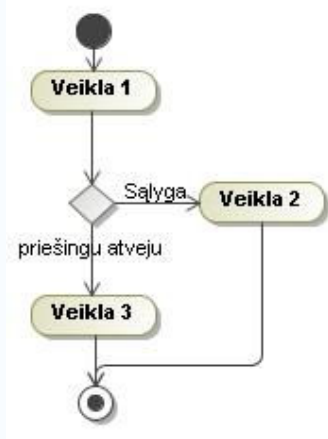
2.14 pav. Pavyzdys tuo pačiu metu vykstančių veiksmų

Paveikslėlyje 2.14, vaizduojamas fragmentas reikalavimų procesui. Du veiksmi, apibūdinantys aktorius ir vartojimo atvejus, yra atliekami tuo pačiu metu. Priežastis, kodėl tie veiksmi turi vykti tuo pačiu metu, yra tai, kad aktorių apibūdinimas ir vartojimo atvejai įtakoja vienas kitą.

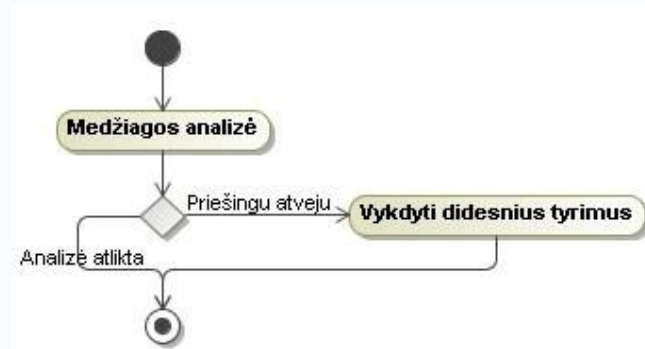
Sąlyginiai veiksmi

Sąlyginiai veiksmi yra veiksmi, kurie yra atliekami, jeigu atitinka konkrečią sąlygą. Tai grafiškai atvaizduojama išsišakojimu. Šakos yra iliustruojamos deimanto struktūra ir gali turėti įvesties ir išvesties perėjimus. Kiekvienas išėjties perėjimas turi apsauginę išraišką, sąlygą. Ši apsauginė išraiška yra „Boolean“ išraiška, naudojama atlikti pasirinkimą, kurią kryptį pasirinkti.

Abu veiksmi ir poveiksmi gali būti modeliuojami kaip sąlyginiai veiksmi. Paveikslėlyje 2.15 iliustruojami du sąlyginiai veiksmi.



2.15 pav. Sąlyginiai veiksmi



2.16 pav. Pavyzdys sąlyginių veiksmų

Paveikslėlyje 2.16 iliustruojamas pavyzdys iš praktikos. Reikalavimų analizė prasideda nuo medžiagos analizės. Remiantis šia studija priimamas sprendimas, ar reikalinga tolimesnė ir platesnė analizė, ar ne. Sąlyga neatlikti šios stadijos yra parodoma kairėje šakos pusėje. Jei ši sąlyga nepriimtina, yra sekama kita rodykle.

Metaduomenų modeliavimas

Diagramos metaduomenų pusė susideda iš koncepcinės diagramos. Tai yra iš esmės pritaikyta klasės diagrama apibūdinta Booch, Rumbaugh ir Jacobson (1999). Svarbiausi klausimai joje yra koncepcija, apibendrinimas, daugialypumas ir konglomeracija.

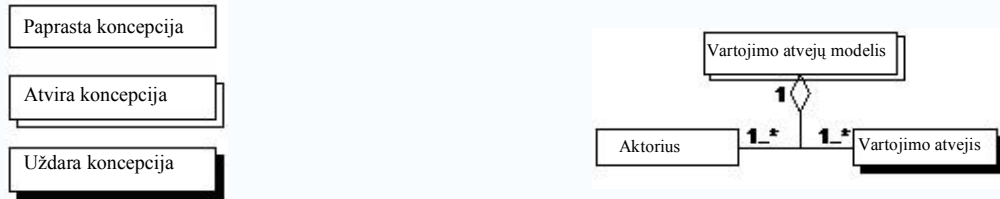
2.5. UML koncepcija

Pirmiausia, koncepcija yra paprasta UML klasės versija. Klasės apibūdinimas pagal Booch, Rumbaugh ir Jacobson (1999) yra pasirenkamas apibrėžti koncepcija, konkrečiai: rinkinys objektų, kurie dalinasi tais pačiais atributais, operacijomis, ryšiais ir semantika.

Sekantys koncepcijų tipai yra patikslinami kaip:

- ✚ Standartinė koncepcija: koncepcija, neturinti tolimesnių koncepcijų. Ji yra vizualizuojama kaip stačiakampis.
- ✚ Kompleksinė koncepcija: koncepcija, susidedanti iš (sub) koncepcijų kolekcijos. Kompleksinės koncepcijos yra skiriamos į:
 - Atvira koncepcija: kompleksinė koncepcija, kurios (sub) koncepcijos yra praplečiamos. Atvira koncepcija yra vizualizuojama kaip du balti stačiakampiai esantys vienas virš kito.
 - Uždara koncepcija: kompleksinė koncepcija, kurios (sub) koncepcijos nėra praplečiamos, kadangi tai nėra susiję su specifiniu turiniu. Uždara koncepcija yra vizualizuojama baltu stačiakampiu virš juodo stačiakampio.

Paveikslėlyje 2.17 yra iliustruojami trys koncepcijų tipai, naudojami modeliavimo technologijoje. Koncepcijos yra visada rašomos didžiosiomis raidėmis, ne tik diagramose, bet ir siejantis su jomis už diagramos ribų.

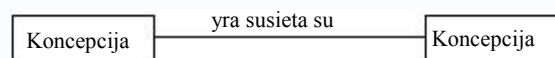


2.17 pav. Standartinė, atvira ir uždara koncepcijos 2.18 pav. Pavyzdys standartinės, atviros ir uždaros koncepcijų

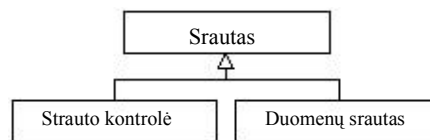
Paveikslėlyje 2.18 visos trys koncepcijos yra pateiktos pavyzdžiais. Iliustruojama dalis proceso duomenų diagramos reikalavimų darbo eigai unifikuotame procese (angl. Unified Process). Vartojimo atvejo modelis (angl. Use Case Model) yra atvira koncepcija ir susideda iš vieno ar daugiau aktorių (angl. Actors) ir vieno ar daugiau vartojimo atvejų (angl. Use Cases). Aktorius yra standartinė koncepcija, neturinti jokių tolimesnių sub koncepcijų. Vartojimo atvejis, vis tik yra uždara koncepcija. Vartojimo atvejis susideda iš apibūdinimo, įvykių sekos, sąlygų, specialių reikalavimų ir t.t. Kadangi šiuo atveju yra nebūtina atskleisti šią informaciją, vartojimo atvejis yra iliustruojamas uždara koncepcija.

2.5.1. Apibendrinimas

Apibendrinimas yra būdas išreikšti santykį tarp bendrinės koncepcijos ir specifinės koncepcijos. Taipogi, reikalui esant, galima nustatyti, ar koncepcijų grupė, kuri yra identifikuojama, yra iš dalies sutampanti ar atskira, užbaigta ar neužbaigta. Apibendrinimas yra vizualizuojamas stora rodykle su atvira rodyklės galvute, rodančia į šaltinį/pagrindą, kaip pavaizduota paveikslėlyje 2.19.



2.19 pav. Apibendrinimas

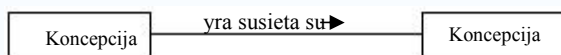


2.20 pav. Apibendrinimo pavyzdys

Paveikslėlyje 2.20 apibendrinimas yra vaizduojamas, parodant ryšius tarp skirtingų koncepcijų, apibūdintų šiame skyrelyje. Standartinė koncepcija ir Kompleksinė koncepcija yra abi koncepcijos specifinės rūšys. Lygiai taip pat Kompleksinė koncepcija gali būti apibūdinama kaip Atvira koncepcija ir Uždara koncepcija.

2.5.2. Asociacijos

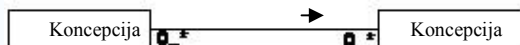
Asociacija yra struktūrinis ryšis, kuris apibūdina kaip koncepcijos yra susijusios tarpusavyje. Ji gali susieti dvi koncepcijas (dvejetainė asociacija) ir daugiau nei dvi koncepcijas (n asociacija). Asociacija yra pristatoma nenukreipta stora linija. Suteikti asociacijai prasmę, gali būti suteikiamas vardas ir vardo kryptis. Vardas būna aktyvioji veiksmažodžio forma ir vardo kryptis yra pristatoma trikampiū, kuris nurodo kryptį, kuria reikia skaityti. Asociacijos su vardu ir vardo kryptimi yra vaizduojamos paveikslėlyje 2.21.



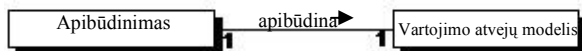
2.21 pav. Asociacija

2.5.3. Daugialypumas

Išskyrus vardą ir vardo kryptį, asociacija gali turėti daugiau charakteristikų. Daugialypiškumu galima nustatyti, kiek objektų tam tikros koncepcijos gali būti susiję atskirais asociacijos atvejais. Daugialypumas yra vaizduojamas naudojant sekančią išraišką: (1) tiksliai vienetui, (0..1) vienetui arba nuliui, (0..*) nuliui ir daugiau, (1..*) vienetui ir daugiau, arba pavyzdžiui (5) tiksliam skaičiui. Paveikslėlyje 2.22 vizualizuojama asociacija su daugialypumu.



2.22 pav. Daugialypumas

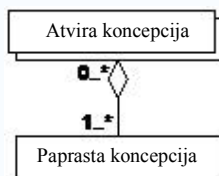


2.23 pav. Daugialypumo pavyzdys

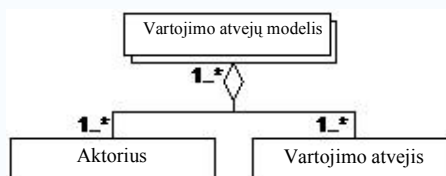
Daugialypumo pavyzdys yra pristatomas paveikslėlyje 2.23. Figūra skaitoma kaip „tiksliai vienas tyrimo apibūdinimas apibūdina tiksliai vieną vartojimo atvejo modelį“. Tai parodo, kad tyrimo apibūdinimas negali apibūdinti nulį ar daugiau nei vieną vartojimo atvejo modelį, ir vartojimo atvejo modelis negali būti apibūdintas nuliū ar daugiau nei vienu tyrimo apibūdinimu.

2.5.4. Agregacija

Specialus asociacijos tipas yra konglomeratas. Konglomeratas reprezentuoja ryšį tarp koncepcijos (kaip visumos), turinčios savyje kitas koncepcijas (kaip dalis). Tai gali būti taip pat apibūdinama kaip turinti ryšį koncepcija. Paveikslėlyje 2.24 iliustruojamas susikaupimo ryšys tarp atviros koncepcijos ir standartinės koncepcijos. Atvira koncepcija susideda iš vienos ar daugiau standartinių koncepcijų ir standartinė koncepcija yra dalis atviros koncepcijos.



2.24 pav. Susikaupimas

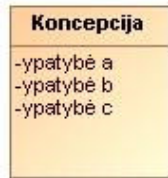


2.25 pav. Susikaupimo pavyzdys

Paveikslėlyje 2.25 susikaupimas yra vaizduojamas kaip dalis proceso duomenų diagramos reikalavimų darbo eigai UML paremtoje tinklo inžinerijoje. Vartojimo atvejo modelis susideda iš vieno ar daugiau aktorių ir vartojimo atvejų.

2.5.5. Savybės

Kartais yra poreikis koncepcijoms priskirti savybes. Savybės yra užrašytos žemiau, po koncepcijos vardu, kaip iliustruojama paveikslėlyje 2.26.



2.26 pav. Susikaupimas



2.27 pav. Susikaupimo pavyzdys

Paveikslėlyje 2.27 vaizduojamas pavyzdys koncepcijos su savybėmis. Koncepcija bruožas (angl. Feature) turi keturias savybes: prioritetus, tipą, riziką ir statusą.

Lentelė 2.2 sąrašas pristatomas kiekvienos koncepcijos reikalauja tinkamo apibūdinimo, kuris yra pageidautinai nukopijuotas iš standartinio specialių terminų žodyno. Visų koncepcijų vardai tekste yra rašomi didžiosiomis raidėmis.

2.2 lentelė. Koncepcijos apibūdinimo sąrašas

Koncepcija	Apibūdinimas
KONCEPCIJA A	Tai yra apibūdinimas KONCEPCIJOS A
KONCEPCIJA B	Tai yra apibūdinimas KONCEPCIJOS B

2.6. Ontologijos

Ir kompiuterių moksle ir informatikos moksle, **ontologija** yra duomenų modelis, kuris reprezentuoja koncepcijų rinkinį domene ir ryšius tarp tų koncepcijų. Jis yra naudojamas pagrįsti objektus domene.

Ontologijos yra naudojamos dirbtiniame prote, semantiniame tinkle, programinės įrangos inžinerijoje, biomedicinėje informatikoje ir informacijos architektūroje kaip žinių pristatymo forma apie pasaulį arba jo dalį. Ontologijos dažniausiai apibūdina:

- ✚ Individualius objektus: pagrindinius arba “pagrindinio lygio” objektus
- ✚ Klases: rinkinius, kolekcijas arba objektų tipus^[17]
- ✚ Atributus: savybes, bruožus, charakteristikas arba parametrus, kuriuos gali turėti objektai arba jie gali jais dalintis.
- ✚ Santykius: būdus, kaip objektai gali būti susiję vieni su kitais
- ✚ Įvykius: Atributų arba ryšių pokyčius

2.6.1. Skirtumai ir panašumai santykiyje su filosofine ontologija

Terminas ontologija originaliai siejamas su filosofija, kur tai yra šaka fundamentalios metafizikos, liečianti skirtingus egzistencijos tipus ir modelius, dažnai dėmesį skiriant santykiams tarp smulkmeniškumo ir universalumo, tarp vidinių ir išorinių savybių. Pagal Tom'ą Gruber'į iš Stanfordo Universiteto, ontologijos reikšmė kompiuterių moksle yra „apibūdinimas koncepcijų ir ryšių, kurie gali egzistuoti veiksniai ar veiksmų bendruomenė“. Jis patikslina, kad ontologija yra iš esmės parašyta kaip „formalaus žodyno apibūdinimų rinkinys“.

Tai, ką ontologija turi bendro su kompiuterių mokslu, yra pristatymas esmių, idėjų ir įvykių, kartu su jų savybėmis ir ryšiais, pagal kategorijų sistemą. Abiejose srityse galima rasti žymių darbų apie ontologijos reliatyvumo problemas (pvz. Quine and Kripke filosofijoje, Sowa ir Guarino kompiuterių moksle), ir debatus, liečiančius klausimą, ar normatyvi ontologija yra perspektyvi (pvz debatai dėl fundamentalizmo filosofijoje, debatai dėl Cyc projekto AI).

Skirtumai dėl abiejų yra iš esmės dėl akcentų. Filosofai yra mažiau susirūpinę dėl sukūrimo fiksuoto ir kontroliuojamo žodyno, nei tyrinėtojai kompiuterių moksle, kuomet kompiuterių specialistai yra mažiau įsitraukę į diskusijas apie dėl pirminių principų (tokių, kaip debatai, ar yra tokie dalykai, kaip fiksuota esmė, ar esmė turi būti ontologiškai pirmesnė nei procesas). 20 a. Antroje pusėje filosofai plačiai svarstė įmanomus metodus arba būdus kurti ontologijas, nekuriant jokių itin įmantrių ontologijų. Tuo metu kompiuterių mokslininkai kūrė dideles ir reikalaujančias pastangų ontologijas (tokias kaip WordNet irCyc) pakankamai mažai diskutuodami apie tai, kaip jos buvo sukurtos.

Pirmaisiais 21a. metais tarpdisciplininis projektas kognityvaus mokslo priartino abiejų tipų mokslininkus. Pavyzdžiui, egzistuoja kalba apie „kompiuterinį poslinkį filosofijoje“, tai apima filosofus, analizuojančius kompiuterių mokslo formalias ontologijas (kartais net tiesiogiai dirbant su programine įranga), kuomet tyrinėtojai kompiuterių moksle daug dažniau siejosi su mokslininkais, dirbančiais su ontologija (kartais su tiesioginėmis pasekmėmis jų metodams). Vis tik dauguma mokslininkų abiejose srityse nėra įsitraukę į šį kognityvų mokslą ir tęsia darbus nepriklausomai vieni nuo kitų, analizuodami paskirai savo problematiką.

2.6.2. Ontologijos elementai

Šiuolaikinės ontologijos turi daug struktūrinių panašumų, nepaisant kalbos, kuria jos yra išreikštos. Kaip paminėta anksčiau, dauguma ontologijų apibūdina individualius objektus (pavyzdžius), klases (konceptijas), atributus ir ryšius.

Individualūs objektai (pavyzdžiai)

Individualūs objektai yra pagrindiniai, "pagrindinio lygio" ontologijos komponentai. Individualūs objektai ontologijoje gali apimti konkrečius objektus, tokius kaip žmonės, gyvūnai, stalai, automobiliai, molekulės, planetos, taipogi kaip ir abstrakčius individualius objektus kaip

skaičiai ir žodžiai. Tiksliau tariant, ontologijai nereikia įtraukti individualaus objekto, bet vienas iš jos esminių tikslų yra suteikti priemones klasifikuoti individualius objektus, nors tie individualūs objektai nėra ontologijos dalis..

Klasės (koncepcijos)

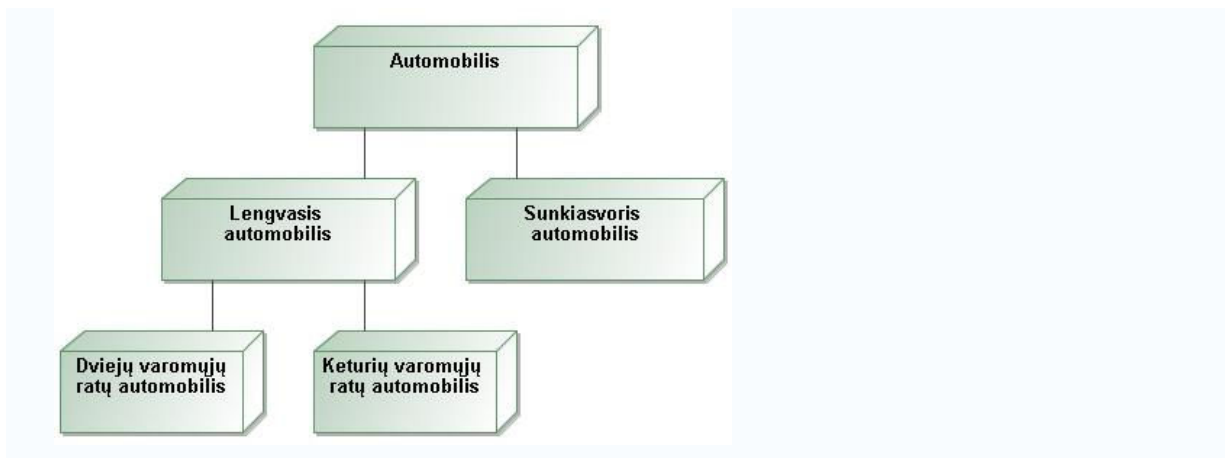
Klasės (koncepcijos) yra abstrakčios grupės, rinkiniai arba objektų kolekcijos. Jie gali turėti individualius objektus, kitas klase arba kombinaciją abiejų. Klasių pavyzdžiai:

- ✚ **Asmuo**, visų žmonių klasė
- ✚ **Molekulė**, visų molekulių klasė
- ✚ **Skaičius**, visų skaičių klasė
- ✚ **Transporto priemonė**, visų transporto priemonių klasė
- ✚ **Mašina**, visų mašinų klasė
- ✚ **Individas**, reprezentuojantis visų individų klasę
- ✚ **Klasė**, reprezentuojanti visų klasių klasę
- ✚ **Daiktas**, reprezentuojantis visų daiktų klasę
- ✚ **Buvusi studentė**, reprezentuojanti visų buvusių studentėlių klasę

Ontologijos varijuoja priklausomai nuo to, ar klasė gali turėti savyje kitų klasių, ar klasė gali priklausyti pati sau, ar yra universalios klasė (tai yra, klasė, turinti viską savyje) ir t.t.

Ontologijų klasės gali būti iš prigimties galinčios plėstis arba nesiplečiančios. Klasė yra galinti plėstis, jeigu ir tik jeigu ji yra charakterizuojama išimtinai savo narystės. Tiksliau klasė C yra galinti plėstis tik ir tik kaip klasė C, jeigu C' turi tiksliai tokius pačius narius kaip klasė C, tuomet klasė C ir C' yra identiškios. Jeigu klasė neatitinka šios sąlygos, tuomet ji yra nesiplečianti. Kuomet galinčios plėstis klasės matematiškai labiau paslankios ir geriau suvokiamos, taip būdamos mažiau problematinės filosofiskai, jos nedraudžia tiksliai apibrėžtų atotrūkių reikalingų dažnai ontologijoje. Pavyzdžiui ontologijai gali reikti atskirti klases visų gyvūnų, turinčių inkstus; ir klases visų gyvūnų, turinčių širdis, net tuo atveju, jeigu šios klasės turi tuos pačius narius.

Svarbu, kad klasė gali priskirti ir būti priskirta kitų klasių. Pavyzdžiui, transporto priemonės apima automobilius, kadangi tai, kas yra narys tolimesnės klasės, yra narys prieš tai esančios klasės. Priskyrimo ryšys yra naudojamas sukurti klasių hierarchiją, tipiskai su maksimaliai bendrine klase kaip **Daiktas** viršuje, ir labai specifine klase kaip **2002 Ford Explorer** pav 2.24.



2.24 pav. Specifinė klasė

Padalijimas yra rinkinys susijusių klasių ir su tuo susijusių taisyklių, kurios leidžia priskirti objektus į tinkamą klasę. Pavyzdžiui, dešinėje yra dalinė ontologijos diagrama, kuri parodo padalijimą automobilių klasės į 2-ratais varomas ir 4-ratais varomas.

Padalijimo taisyklė nustato, ar tam tikra mašina turi būti patalpinta į 2-ratais ar 4-ratais varomų mašinų klases.

Jeigu padalijimo taisyklė garantuoja, kad vienietinis objektas automobilis negali būti dviejose klasėse, tuomet padalinimas yra vadinamas **Išskirstymo Padalijimu**. Jeigu padalijimo taisyklės užtikrina, kad kiekvienas konkretus objektas super-klasėje yra pavienis objektas bent vienos padalijimo klasės, tuomet padalijimas yra vadinamas **Išsamiu Padalijimu**.

Atributai

Objektai ontologijoje gali būti apibūdinami pagal jiems priskiriamus atributus. Kiekvienas atributas mažiausiai turi vardą ir vertę, ir yra naudojamas saugoti informaciją, kuri yra būdinga objektui, prie kurio yra prisegtas. Pavyzdžiui objektas Ford Explorer turi tokius objektus:

- 🚗 *Vardas*: Ford Explorer
- 🚗 *Durų skaičius*:- 4
- 🚗 *Variklis*: {4.0L, 4.6L}
- 🚗 *Transmisija*: 6

Atributo vertė gali būti kompleksinio duomenų tipo; šiame pavyzdyje atributo, pavadinto *Variklis*, vertė yra verčių sąrašas, ne tik pavienė vertė.

Jeigu jūs neapibrėžėte koncepcijoms atributų, jūs turėsite arba taksonomiją (jeigu hiponiminės jungtys egzistuoja tarp koncepcijų) arba **Kontroliuojamą Žodyną**. Tai yra naudinga, bet nelaikomos tikromis ontologijomis. Jos gali būti grynos ontologijos.

Ontologija apibrėžia domeną.

2.6.3. Ontologijos kalbos

Ontologijos kalba yra formali kalba, kuri yra naudojama užšifruoti ontologiją. Esti keletas tokių kalbų ontologijoms, ir patentuotos, ir paremtos standartais:

- ✚ OWL (angl. Web Ontology Language) yra kalba sukurti ontologinius teiginius, sukurtus RDF ir RDFS, kaip ir ankstesniuose ontologijos projektuose, įskaitant OIL (angl. Ontology Inference Layer), DAML (angl. Darpa agent markup language) ir DAML+OIL. OWL yra skirta naudoti pasauliniame tinkle (angl. World Wide Web), ir visi jos elementai (klasės, savybės ir individualūs objektai) yra apibrėžiami kaip RDF šaltiniai, ir identifikuojami URIs.
- ✚ KIF yra sintaksė first-order logic paremta S-expressions.
- ✚ Cyc projektas turi savo ontologinę kalbą, vadinamą CycL, paremtą first-order predicate calculus su tam tikrais praplėtimais.

2.7. Modeliu paremta inžinerija

Modeliu paremta inžinerija (arba **MDE**) siejasi su sistematiniu naudojimu modelių, kaip pirminių inžinerijos artefaktų, inžinerijos cikle. MDE gali būti taikoma programinei įrangai, sistemos ir duomenų inžinerijai. Geriausiai žinoma MDE yra Objektų valdymo grupė (angl. Object Management Group (OMG)) vadinama modeliu paremta architektūra (sutr. MDA), kas yra registruotas prekinis ženklas, priklausantis OMG.^[18] Kitas susijęs akronimas yra modeliu paremtas vystymas (angl. Model-Driven Development (MDD)), kas irgi yra OMG prekinis ženklas.

Kas liečia programinės įrangos vystymą, modeliu paremtas vystymas siejasi su grupe vystymo metodų, kurie yra paremti programinės įrangos modeliavimo naudojimu kaip pirminės ekspresijos forma. Kartais modeliai yra konstruojami iki tam tikro detalių lygio, o kodas yra užrašomas ranka atskirame etape. Kartais užbaigti modeliai sukuriami įskaitant vykdomuosius veiksmus. Kodas gali būti sugeneruotas iš modelių, varijuojant nuo sistemos griaučių iki užbaigto produkto. Pristačius Unifikuotą modeliavimo kalbą (sutr. UML), MDD tapo šiandien labai populiariu tarp daugybės praktinių ir palaikančių įrankių. Labiau pažengę MDD tipai prasaplėtė, kad įgalintų industrinius standartus pastoviam taikymui ir rezultatams. Besitęsianti MDD evoliucija pridėjo ir padidino dėmesį architektūrai ir automatizacijai.

MDD technologijos su dideliu dėmesiu architektūrai ir automatizacijai atskleidžia aukštesnius abstrakcijos lygius programinės įrangos vystyme. Ši abstrakcija skatina paprastesnius modelius su didesniu dėmesiu probleminei vietai. Kartu su vykdomąja semantika tai tobulina įmanomos automatizacijos aukštesnes galimybes. Objekto valdymo grupė (OMG) sukūrė rinkinį standartų, pavadintų modeliu paremta architektūra (MDA), Sudarant pamatus šiam patobulintam architektūra paremtam metodui.

Modelį integruojanti kompiuterija yra dar viena MDE šaka. Pagal Douglas Schmidt, modeliu paremtos inžinerijos technologijos siūlo daug žadantį metodą adresuoti trečios kalbos nesugebėjimą palengvinti platformų kompleksiją ir išreikšti efektyviai domenų koncepcijas. [19].

2.8. Darbo tikslas ir siekiami privalumai

Tikslas:

- ✚ Modifikuoti esamą portalą, remiantis naujausiomis esamomis technologijomis ir jų tendencijomis rinkoje.
- ✚ Susisteminti informaciją apie statybos kaštus, naujas portalų kūrimo technologijas.
- ✚ Išanalizuoti naujas portalų kūrimo technologijas kokybės atžvilgiui.
- ✚ Išanalizuoti statybos bendrovių teikiamas paslaugas ir esamus projektus.

Privalumai:

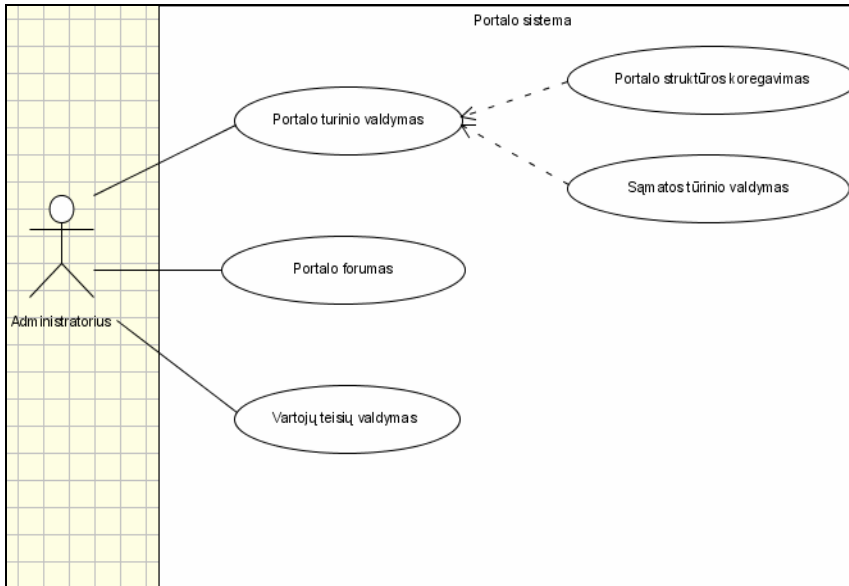
- ✚ Portale įdėtas metaduomenų lygis, kuriame yra pagrindiniai portalo elementai.
- ✚ Atskirti duomenys ir metaduomenys leidžia lengviau modifikuoti portalo struktūrą.
- ✚ Padidėja portalo veikimo greitis.

2.9. Kompiuterizuojamos sistemos funkcijos

- ✚ Vartotojo registracija
- ✚ Šamatos skaičiavimas
- ✚ Šamatos saugojimas duomenų bazėje
- ✚ Firmos puslapio valdymas ir koregavimas
- ✚ Vartotojų koregavimas
- ✚ Vartotojai išskiriami pagal teises
- ✚ Portalo turinio valdymas

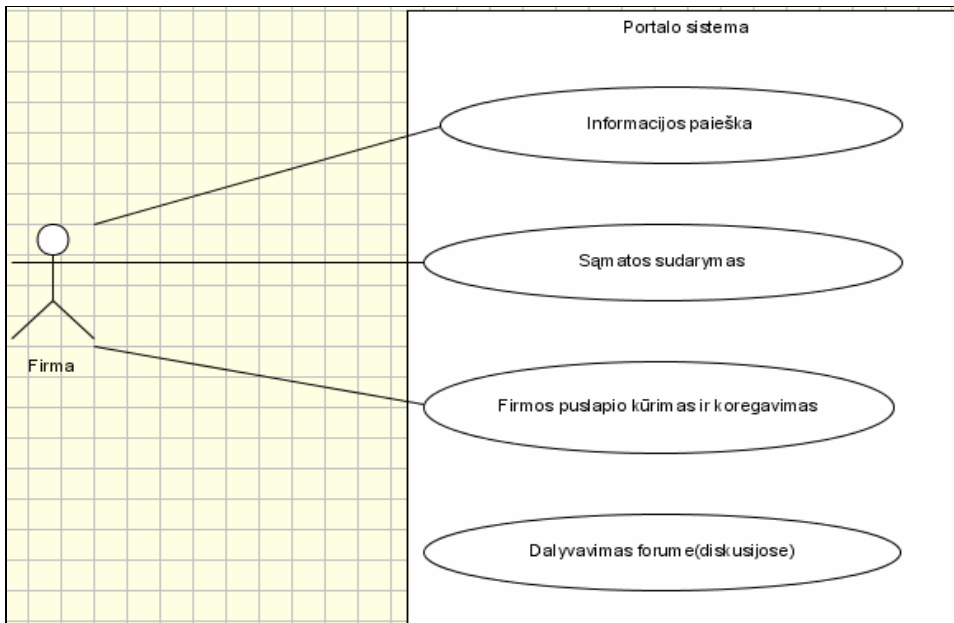
Žemiau pateikta panaudojimo atvejų diagrama kaip tik ir iliustruoja projektuojamos sistemos funkcijas bei kokios funkcijos prieinamos vartotojams su skirtingomis teisėmis.

Administratoriaus Use Case diagrama, parodanti jo galimybes portale.



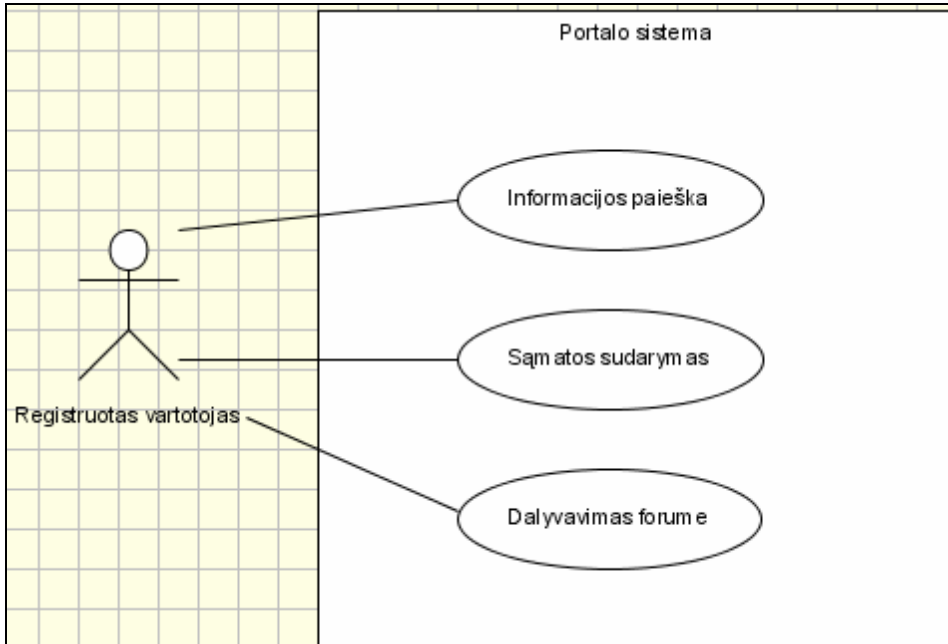
Pav 2.25 Administratoriaus use case diagrama

Organizacijos Use Case diagrama, parodanti jos galimybes portale.



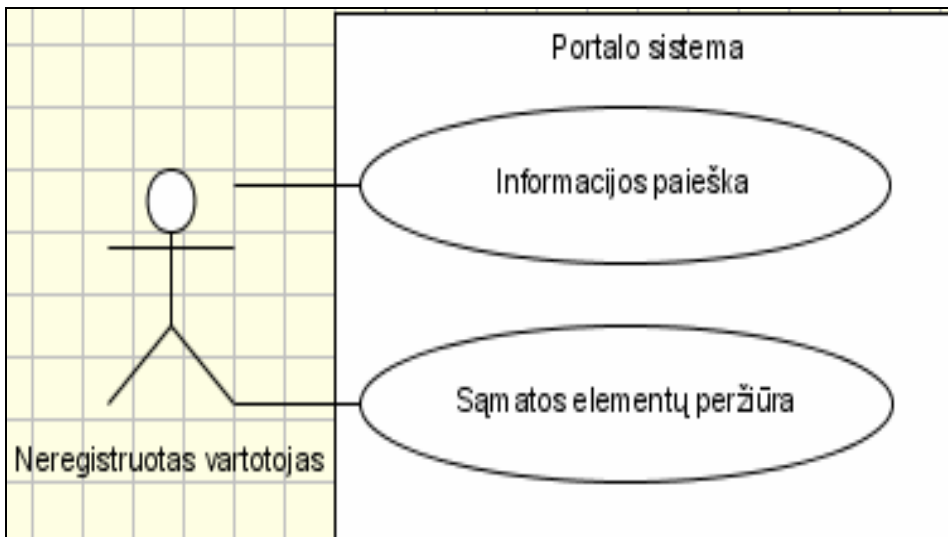
Pav 2.26 Firmos use case diagrama

Registruoto vartotojo Use Case diagrama, parodanti vartotojo galimybes portale.



Pav 2.27 Registruoto vartotojo use case diagrama

Neregistruoto vartotojo Use Case diagrama, parodanti jo galimybes portale.



Pav 2.28 Neregistruoto vartotojo use case diagrama

Administratorius – sistemos administratorius, jam suteikiamos didžiausios teisės, jis gali atlikti visus veiksmus t.y portalo struktūros koregavimą, šam atos turinio valdymas, forumo valdymas, vartotojų kūrimas ir jų teisių valdymas.

Firma- tai vartotojas, kuris yra užsiregistravęs prie sistemos ir administratorius jam suteikė „firmos“ statusą. Jis gali atlikti informacijos paiešką, sudaryti sąmatą, koreguoti ir pildyti firmos puslapį, dalyvauti diskusijose.

Registruotas vartotojas – tai vartotojas kuris, yra užsiregistravęs prie sistemos. Jam leidžiama atlikti informacijos paiešką, sudaryti sąmatą, dalyvauti diskusijose.

Neregistruotas vartotojas – tai vartotojas kuris, yra neužsiregistravęs prie sistemos. Jam leidžiama atlikti informacijos paiešką, sąmatos elementų peržiūra.

Mūsų sistemoje vartotojai yra dalinami į keturis grupes, nes norima atskirti priėjimą prie konkrečios paslaugos. Tokiu atveju bus išplečiamos ar blokuojamos konkrečių vartotojų grupių paslaugos.

2.10. Reikalavimai duomenims

Projektuojamas portalas turi tiktai skirtingoms kompiuterinėms sistemoms. Kadangi mūsų projektas yra realizuojamas internete, mes pasirinkome PHP ir SQL programavimo kalbas, nes suprojektuota sistema yra lengvai kuriama ir integruojama į Linux ar Windows sistemas. Pagrindiniai reikalavimai prieš projekto įdiegimą, bus privaloma įdiegti MySQL duomenų bazės, Apache serverius ir PHP. Kadangi portalas bus internetinis, todėl jis bus lengvai pasiekiamas, bet kokiam vartotojui, kuris turės prieigą prie interneto.

2.11. Nefunkciniai reikalavimai ir apribojimai

Projektuojama sistema turi būti lengvai plečiama. Ateityje reikalingi sistemos plečiamumo darbai neturėtų reikalauti daug pastangų aiškinantis esamos vartotojo sąsajos veikimo principus. Esant reikalui galima būtų koreguoti ir pildyti portalo turinį, informaciją be didesnių pastangų, netgi nežinant HTML / HTTP / FTP / JAVA / Frontpage / Dreamweaver / kalbų ar puslapio redagavimo įrankių. Portalas turi būti sukurtas taip, kad būtų lengvai suderinamas su kitomis sistemomis.

Pritaikomumas

- 🚦 Vartotojo sistema turi būti lengvai pritaikoma pagal kliento poreikius;
- 🚦 Paprastai koreguojamos vartotojų grupės;
- 🚦 Vartotojų grupės pritaikomos pagal poreikius;
- 🚦 Papildomas ar susiaurinamas paslaugų sąrašas;

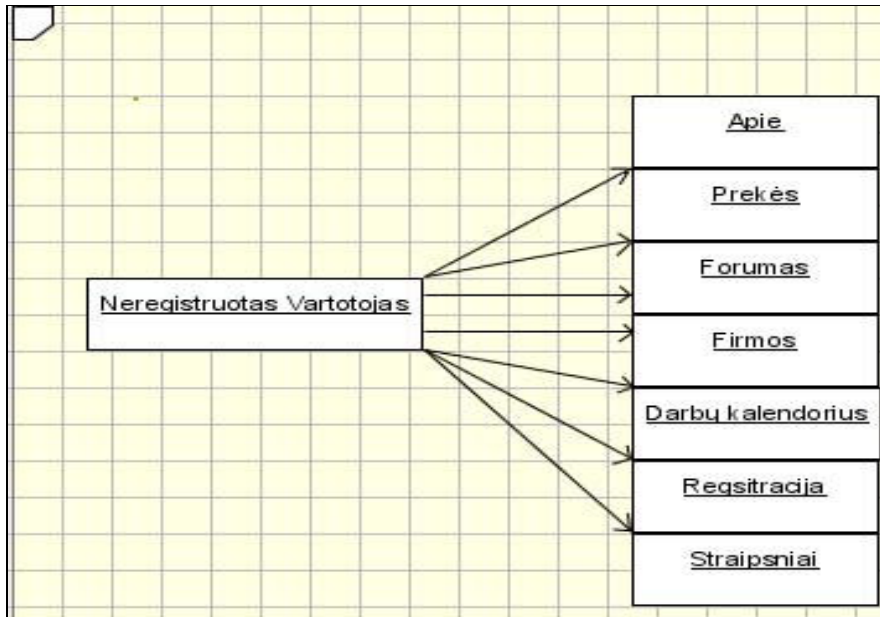
Saugumas

- 🚦 Sistema turi būti saugi;
- 🚦 Prisijungimas reikalaujantis prisijungimo vardo ir slaptažodžio
- 🚦 Portalo valdymo funkcijos vartotojams priskiriamas pagal jiems suteiktas teises

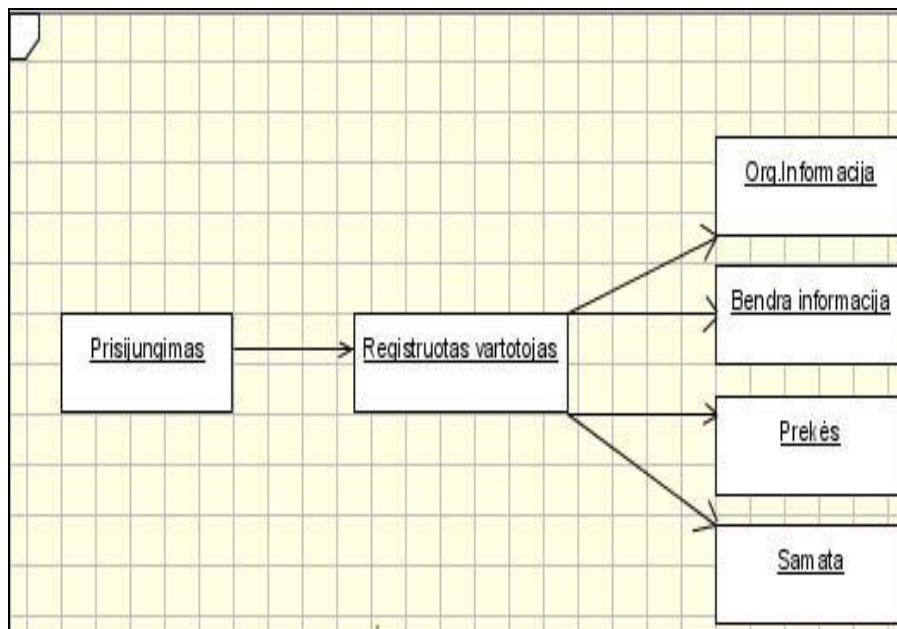
Paprastumas

Vartotojo sąsaja turi būti suprantama visiems vartotojams ir net neturintiems didelės patirties dirbant su kompiuteriu. Taip pat galima būtų greitai ir lengvai rasti ir pasinaudoti informacija, bei atlikti reikiamas užduotis, kad nereikėtų gaišti brangaus laiko ir blaškytis po visą svetainę. Neturėtų varginti didelis spalvų spektras, vartotojo sąsajos struktūra turėtų būti neperkrauta. Naudojimasis turi pareikalauti minimalių vartotojo pastangų.

Žemiau pateikiamos projektuojamos grafinės sąsajos struktūrinės schemas.



Pav 2.29 Neregistruoto vartotojo struktūrinė sąsajos schema



Pav 2.30 Registruoto vartotojo struktūrinė sąsajos schema

2.11.1. Rizikos faktorių analizė

- ✚ Tinklapis neveiks dėl serverio kaltės.
- ✚ Portalas neveiks dėl techninių kliūčių.

2.11.2. Rezultato kokybės kriterijai

- ✚ Greita ir kokybiška informacija.
- ✚ Naujausios technologijos.
- ✚ Patikima informacija.

2.12. Analizės išvados

1. Išanalizavus portalų kūrimo technologijas nustatyta, kad anksčiau sukurti portalai neturi metaduomenų lygio.
2. Atlikus nagrinėjamų portalų lyginamąją analizę, apibrėžtos kuriamo portalo išskirtinės savybės.
3. Portalo plėtimo galimybės labai didelės. Kadangi turinio valdymo sistema, leidžia greitai ir patogiai įkelti duomenis, juos koreguoti, papildyti ir panaikinti.
4. Apsibrėžus sistemos funkcijas ir ribas, buvo nustatyta projektuojamos ir realizuojamos darbo dalies apimtis.
5. Atlikus rizikos faktorių analizę, nustatėme, kad nėra daug sistemos savybių, dėl kurių atsiranda papildoma rizika kuriant IS, todėl apsibrėžėme rizikos faktorių eliminavimo būdus, kad to išvengti.
6. Išanalizavus esamus portalus, nustatėme rezultato kokybės kriterijus, kurie bus įvertinti darbo pabaigoje.
7. Išanalizavus sukurtų portalų esamą padėtį Lietuvoje, buvo apibrėžta kuriamo portalo nauda.
 - ✚ Supaprastintas portalo valdymas, nes portalo struktūra yra apibrėžta metaduomenų lygyje.
 - ✚ Greita informacija.
 - ✚ Taupomos eksploatacinės išlaidos

3. Projektinė dalis

3.1. Įvadas

Tobulėjant interneto sistemoms, tapo įprasta naudoti dinامينius tinklo puslapius, kurie reaguoja į vartotojo įvestį. Serveryje veikiančių puslapių kūrimas - Active Server Pages (ASP) ir Java Server Pages (JSP) leidžia kūrėjams tiesiogiai įdėti tekstus į HTML puslapius, tokiu būdu suprastinant programavimo modelį. Sudėtingesniems įmontuotų tekstų taikymams atsirado poreikis atskirti veiklos logiką nuo pateikties logikos puslapių lygyje. Deja, nėra vienos kūrimo strategijos, kuri tiktų visoms situacijoms. Dėl programinės įrangos kūrimo priemonių konkurencijos atsirado būtinybė pašalinti perdėtą įvairovę ir sudėtingumą.

3.2. Portalo dalykinės srities modelis

3.2.1. Architektūros pateikimas

Dokumente sistemos architektūra pateikiama keliais vaizdais:

Panaudojimo atveju (PA)

- a) statinis,
- b) dinaminis,
- c) išdėstymo.

Šie vaizdai yra pateikiami naudojant unifikuotą modeliavimo kalbą (UML).

Sistemos specifikacija pateikta šiais vaizdais,

- ✚ panaudojimo atvejų vaizdas (panaudojimo atvejų diagrama);
- ✚ sistemos statinis vaizdas (paketai ir klasių diagramos);
- ✚ sistemos dinaminis vaizdas (būsenu, veiklos, sekų);
- ✚ išdėstymo vaizdas (išdėstymo diagrama).

Jiems įgyvendinti reikia UML diagramų.

3.2.2. Architektūros tikslai ir apribojimai

Architektūrinius sprendimus įtakoiantys reikalavimai:

- ✚ sistema turi būti suprojektuota taip, kad ją galima būtų lengvai pakeisti.
- ✚ sistema neturi leisti neautorizuotiems vartotojams prie jos prisijungti;
- ✚ turi būti atsižvelgta į būtinas programos vykdymo charakteristikas,

apibrėžtas reikalavimų specifikacijoje.

3.2.3. Kompiuterizuojamos sistemos funkcijos

- ✚ Vartotojo registracija
- ✚ Šamatos skaičiavimas
- ✚ Šamatos saugojimas duomenų bazėje

- ✚ Firmos puslapio valdymas ir koregavimas
- ✚ Vartotojų koregavimas
- ✚ Vartotojai išskiriami pagal teises
- ✚ Portalo turinio valdymas

3.2.4. Panaudojimo atvejai

Joks objektinės orientacijos projektavimas neišsivaizduojamas be klasių diagramų. Tačiau programinės įrangos kūrimo procesas dažniausiai pradedamas nuo panaudos modelio (Use Case model) sudarymo. Panaudos modelis apibrėžia pagrindines sistemos funkcijas (kaip jas suvokia vartotojas). Šis modelis sudaromas analizuojant vartotojų reikalavimus sistemai. Apibrėžiant sistemos funkcijas ir sąveikos su ja taškus, kartu nurodomos sistemos ribos, t.y. ką nagrinėjama sistema atlieka, ir ką jai atlikti priklauso. Be to, panaudos modelis padeda planuoti vystymo procesą ir atlikti sistemingą produkto testavimą. Panaudos modelį sudaro panaudos diagramos. Nėra visuotinai priimto vertimo terminui “use case diagram”. Vartojami terminai:

- ✚ “vartojimo atvejų diagrama”,
- ✚ “taikomojo uždavinio diagrama”,
- ✚ “naudojimo variantų diagrama”,
- ✚ “panaudos atvejų diagrama”,
- ✚ “scenarijų diagrama”,
- ✚ “precedentų diagrama” ir pan.

Panaudos diagramos yra paprastos ir intuityviai suvokiamos: jas supranta ir sistemos užsakovai. Panaudos diagramą sudaro: panaudos atvejai (use cases), juose dalyvaujantys aktoriai (actors) ir ryšiai tarp jų.

Panaudos atvejis yra tipinė vartotojo ir sistemos sąveika. Tai tam tikra sistemos funkcija, vartotojo suvokiama kaip vientisa baigtinė operacija, kuri duoda matomą (“užčiuopiamą”) rezultatą. Pavyzdžiui, kalbant apie interneto naršyklę, panaudos atvejais galima laikyti operacijas “pereiti į kitą tinklalapį”, “grįžti į ankstesnį tinklalapį” ir pan. Panaudos atvejis vaizduojamas kaip ovalas, su viduje įrašytu pavadinimu. Pavadinimas dažniausiai būna trumpas veiksmo aprašymas iš keleto žodžių.

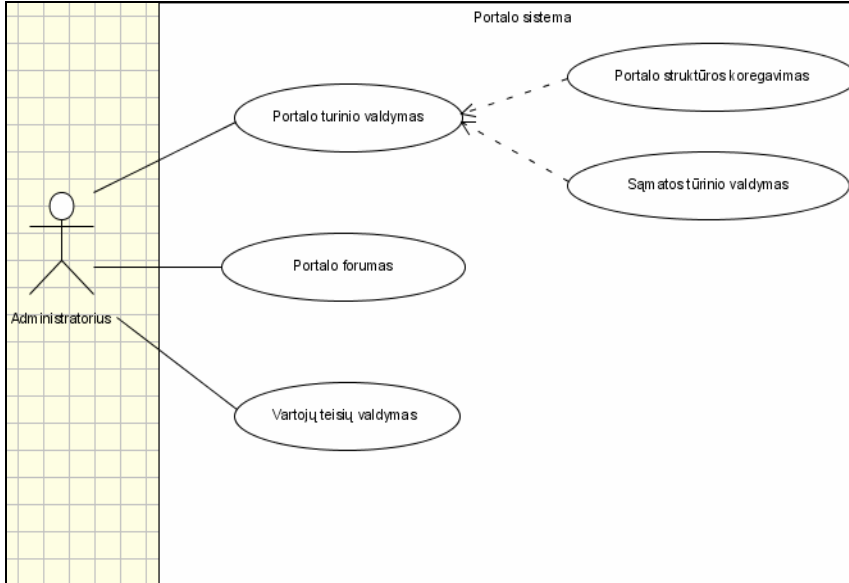
Aktorius yra ne atskiras vartotojas (pvz. Jonas Jonaitis), o tam tikras vartotojo tipas, vaidmuo (direktorius, mokytojas, klientas). Vienas vartotojas gali “vaidinti” keletą vaidmenų, pvz.: vienas realus asmuo gali kreiptis į sistemą ir kaip mokyklos direktorius, ir kaip mokytojas. Aktoriumi gali būti ir ne žmogus, o kita kompiuterizuota sistema (pvz. asmens tapatybės nustatymo sistema, kreditinės kortelės tikrinimo sistema). Aktorius vaizduojamas kaip stilizuotas žmogeliukas. Asociacijos ryšys tarp aktoriaus ir panaudos atvejo dažnai vadinamas bendravimo ryšiu (angl. communication), kuris rodo, kad aktorius dalyvauja panaudos atvejuje, t.y. keičiasi pranešimais su

sistema. Kas siunčia pranešimą ir kam, parodo ryšio kryptis. Komunikacijos ryšio galuose galima nurodyti ryšio kardinalumą. Komunikacija yra vienintelis galimas ryšys tarp aktorius ir panaudos atvejo. Aktoriai ir panaudos atvejai gali būti susiję generalizacijos ryšiu ir atitinkamai panaudos ryšiais su kitais aktoriais. Generalizacijos ryšys aktoriams ir panaudos atvejams turi tokią pačią prasmę kaip ir klasėms. Panaudos atvejai gali būti susiję tarpusavyje priklausomybės (dependability) ryšiais. Priklausomybės ryšys parodomas kaip punktyrinė linija su paprasta rodykle, nukreipta į naudojamąjį elementą. Panaudos atveju tarpusavio priklausomybė būna dviejų tipų:

Kai vienas panaudos atvejis be kitų savo funkcijų atlieka ir kito panaudos atvejo aprašomą funkciją, sakoma, kad pirmasis “apima” (include) antrąjį. “Apėmimo” ryšys taikomas siekiant išvengti pasikartojimų dviejuose arba daugelyje panaudos atveju. Tai labai primena klasių agregaciją. Kai vienas panaudos atvejis yra labiau specializuotas kito panaudos atvejo variantas, sakoma, kad pirmasis “praplečia” (extend) antrąjį. “Praplėtimo” ryšys naudojamas aprašyti normalaus sistemos darbo pakitimus. Šie ryšiai leidžia panaudos atvejus naudoti pakartotinai, padeda geriau suvokti ir analizuoti sistemos funkcijas. Panaudos atvejai gali būti skirtingo detalumo laipsnio. Pavyzdžiui, vystant teksto redaktorių galima specifikuoti panaudos atveją “užrašyti tekstą pariebintu šriftu”, o galima iki tokio lygio nenusileisti ir apsiriboti “stambesniu” panaudos atveju “suformatuoti tekstą”. Metodologijų kurėjai nesutaria kiek panaudos atveju kokiai sistemai reikia. Pavyzdžiui, Ivaras Jacobsonas teigia, kad aprašyti projektą, kuriam vystyti vienas žmogus užtruktų 10 metų, reikia iki 20 panaudos atveju (neskaitant “praplėtimo” ir “apėmimo” būdu išvestų panaudos atveju). Kiti specialistai teigia, kad tokios apimties projektui gali reikėti šimto ar daugiau panaudos atveju.

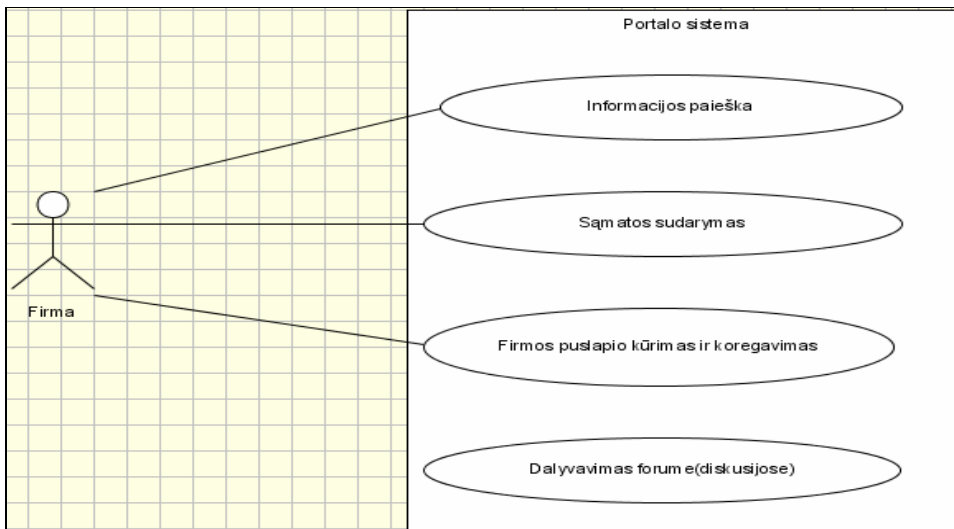
Žemiau pateikta panaudojimo atveju diagrama kaip tik ir iliustruoja projektuojamos sistemos funkcijas bei kokios funkcijos prieinamos vartotojams su skirtingomis teisėmis.

Administratoriaus Use Case diagrama, parodanti jo galimybes portale



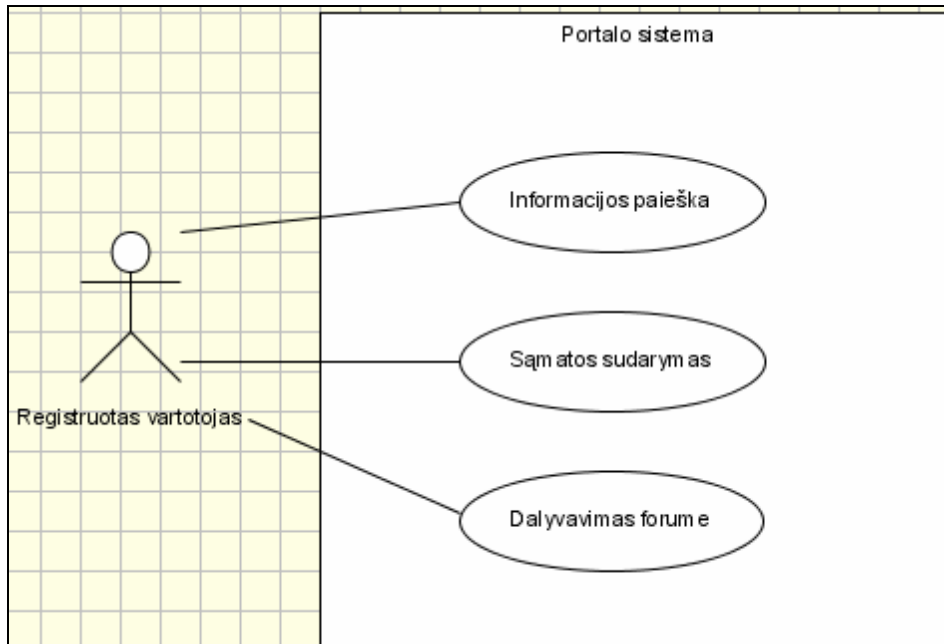
Pav 3.1 Administratoriaus use case diagrama

Organizacijos Use Case diagrama, parodanti jos galimybes portale



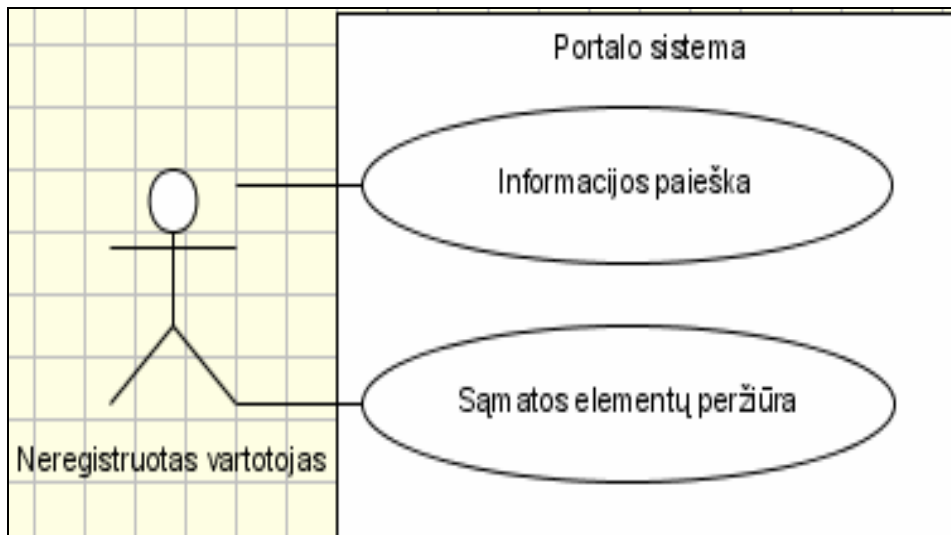
Pav 3.2 Firmos use case diagrama

Registruoto vartotojo Use Case diagrama, parodanti vartotojo galimybes portale



Pav 3.3 Registruoto vartotojo use case diagrama

Neregistruoto vartotojo Use Case diagrama, parodanti jo galimybes portale



Pav32.4 Neregistruoto vartotojo use case diagrama

Administratorius – sistemos administratorius, jam suteikiamos didžiausios teisės, jis gali atlikti visus veiksmus t.y portalo struktūros koregavimą, šam atos turinio valdymas, forumo valdymas, vartotojų kūrimas ir jų teisių valdymas.

Firma- tai vartotojas, kuris yra užsiregistravęs prie sistemos ir administratorius jam suteikė „firmos“ statusą. Jis gali atlikti informacijos paiešką, sudaryti sąmatą, koreguoti ir pildyti firmos puslapį, dalyvauti diskusijose.

Registruotas vartotojas – tai vartotojas kuris, yra užsiregistravęs prie sistemos. Jam leidžiama atlikti informacijos paiešką, sudaryti sąmatą, dalyvauti diskusijose.

Neregistruotas vartotojas – tai vartotojas kuris, yra neužsiregistravęs prie sistemos. Jam leidžiama atlikti informacijos paiešką, sąmatos elementų peržiūra.

Mūsų sistemoje vartotojai yra dalinami į keturis grupes, nes norima atskirti priėjimą prie konkrečios paslaugos. Tokiu atveju bus išplečiamos ar blokuojamos konkrečių vartotojų grupių paslaugos.





3.2.5. Reikalavimai duomenims

Projektuojamas portalas turi tiktai skirtingoms kompiuterinėms sistemoms. Kadangi mūsų projektas yra realizuojamas internete, mes pasirinkome PHP ir SQL programavimo kalbas, nes suprojektuota sistema yra lengvai kuriama ir integruojama į Linux ar Windows sistemas. Pagrindiniai reikalavimai prieš projekto įdiegimą, bus privaloma įdiegti MySQL duomenų bazės, Apache serverius ir PHP. Kadangi portalas bus internetinis, todėl jis bus lengvai pasiekiamas, bet kokiam vartotojui, kuris turės prieigą prie interneto.



3.2.6. Nefunkciniai reikalavimai ir apribojimai

Projektuojama sistema turi būti lengvai plečiama. Ateityje reikalingi sistemos plečiamumo darbai neturėtų reikalauti daug pastangų aiškinantis esamos vartotojo sąsajos veikimo principus. Esant reikalui galima būtų koreguoti ir pildyti portalo turinį, informaciją be didesnių pastangų, netgi nežinant HTML / HTTP / FTP / JAVA / Frontpage / Dreamweaver / kalbų ar puslapio redagavimo įrankių. Portalas turi būti sukurtas taip, kad būtų lengvai suderinamas su kitomis sistemomis.

Pritaikomumas

-  Vartotojo sistema turi būti lengvai pritaikoma pagal kliento poreikius;
-  Paprastai koreguojamos vartotojų grupės;
-  Vartotojų grupės pritaikomos pagal poreikius;
-  Papildomas ar susiaurinamas paslaugų sąrašas;

Saugumas

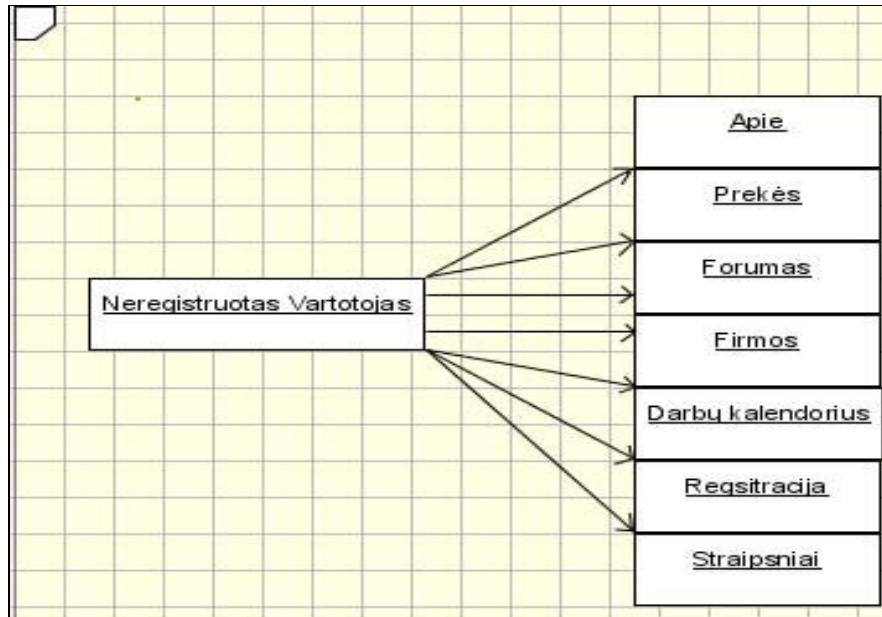
-  Sistema turi būti saugi;
-  Prisijungimas reikalaujantis prisijungimo vardo ir slaptažodžio

🚦 Portalo valdymo funkcijos vartotojams priskiriamas pagal jiems suteiktas teises

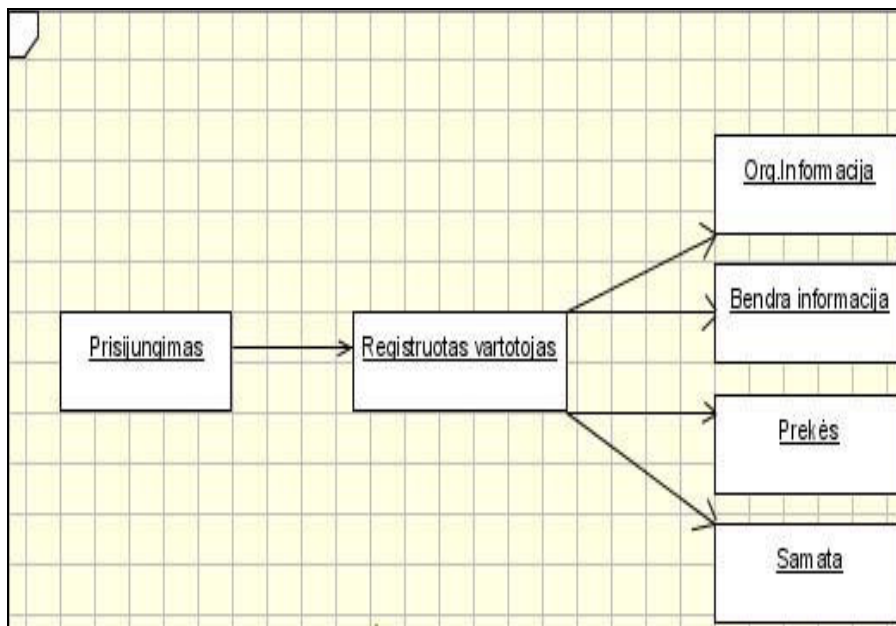
Paprastumas

Vartotojo sąsaja turi būti suprantama visiems vartotojams ir net neturintiems didelės patirties dirbant su kompiuteriu. Taip pat galima būtų greitai ir lengvai rasti ir pasinaudoti informacija, bei atlikti reikiamas užduotis, kad nereikėtų gaišti brangaus laiko ir blaškytis po visą svetainę. Neturėtų varginti didelis spalvų spektras, vartotojo sąsajos struktūra turėtų būti neperkrauta. Naudojimasis turi pareikalauti minimalių vartotojo pastangų.

Žemiau pateikiamos projektuojamos grafinės sąsajos struktūrinės schemos.



Pav 3.5 Neregistruoto vartotojo struktūrinė sąsajos schema



Pav 3.6 Rregistruoto vartotojo struktūrinė sąsajos schema

3.2.7. Rizikos faktorių analizė

- Tinklapis neveiks dėl serverio kaltės.
- Portalas neveiks dėl techninių kliūčių.

3.2.8. Rezultato kokybės kriterijai

- Greita ir kokybiška informacija.
- Naujausios technologijos.
- Patikima informacija.

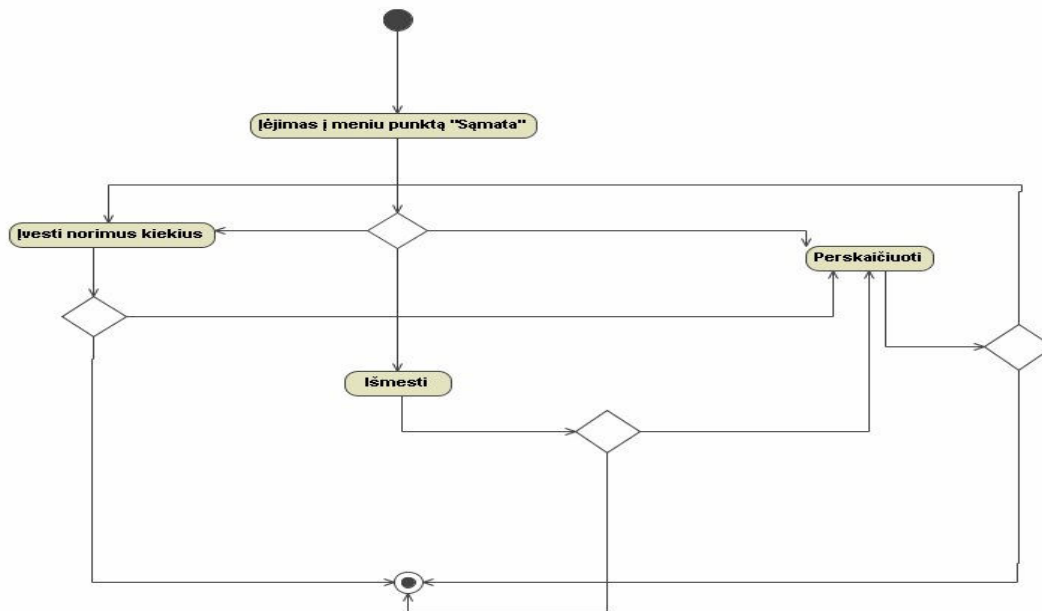
3.3. Projektuojamo portalo projektas

3.3.1. Veiklos diagramos

Šios UML diagramos naudojamos modeliuoti sistemos elgesį arba veiklą. Veikla – tai tam tikras darbas, kurį reikia atlikti. Programinės realizacijos požiūriu, veikla yra kažkokios klasės metodas. Veiklos diagramoje darbas vaizduojamas kaip užapvalintas keturkampis. Kai veikla yra tik kažkokio sprendimo priėmimas (sąlygos tikrinimas), jai atvaizduoti naudojamas tuščiaviduris rombas.

Iš pažiūros veiklos diagramos atrodo panašios i blokines algoritmų schemas (flowchart), tačiau yra ypatingos tuo, kad gali turėti sinchronizacijos linijas. Sinchronizacijos linija rodo, kad vienu metu pradedami daryti du ar daugiau darbų. Dažnai tai reiškia tik tai, kad darbų vykdymo eilės tvarka nesvarbi. Dėl galimybės atvaizduoti sinchronizavimą, veiklos diagramos gerai tinka projektuoti lygiagrečiai veikiančias sistemas.

Sąmatos peržiūros veiklos diagrama pateikta 3.7 paveiksle.:



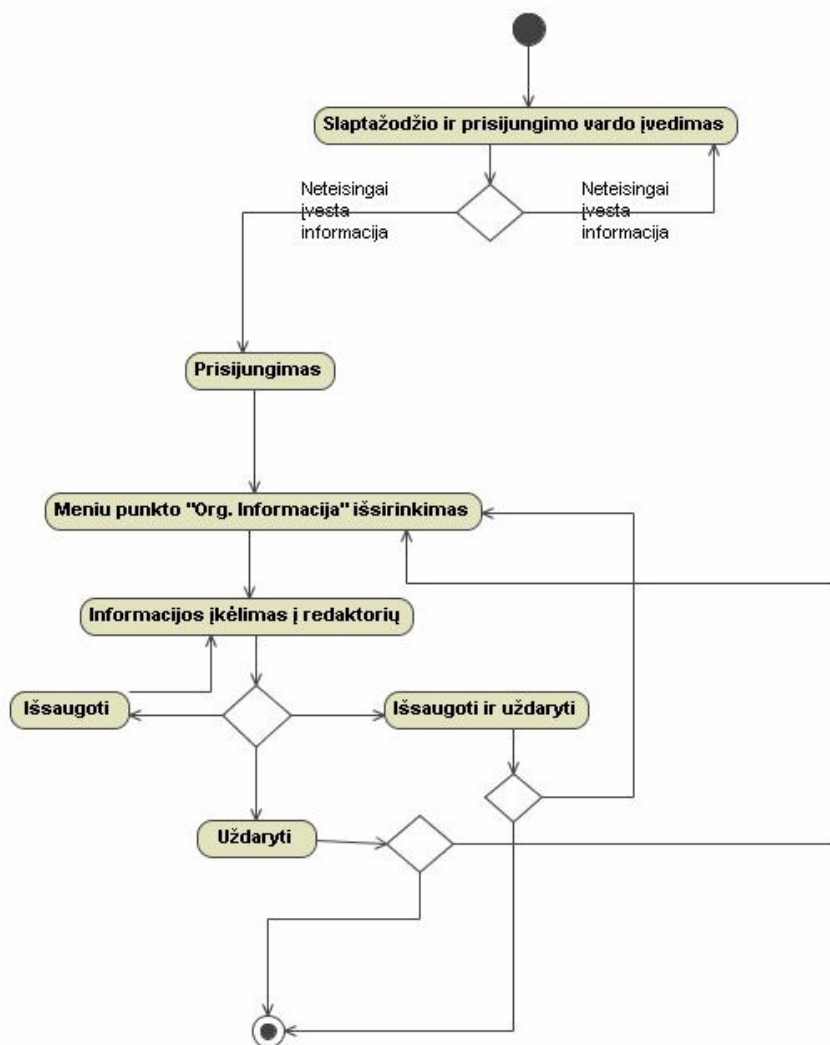
Pav 3.7 Sąmatos peržiūros „activity“ diagrama

Organizacijos informacijos įkėlimas į tinklapį.

Šiais informacinių technologijų laikais, visos organizacijos plečia veiklą internete. Jos nori turėti savo reklaminius tinklapius. Tai mes leidžiame visoms įmonėms padaryti savo portale.

Kiekviena registruota firma gali įkelti savo kontaktus ir reklaminę informaciją.

Organizacijos informacijos įkėlimo diagrama pateikta 3.8 paveiksle.

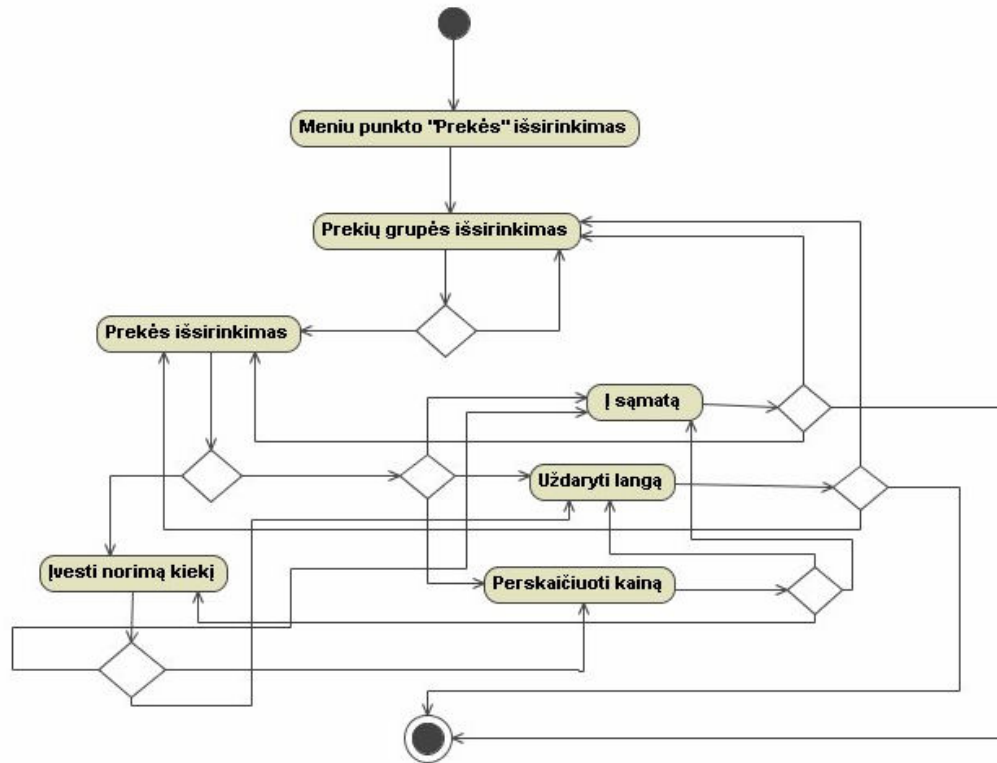


Pav 3.8 Organizacijos informacijos įkėlimas į tinklapį

Pirmiausia organizacija turi prisijungti prie savo vartotojo srities. Jeigu slaptažodis ir vartotojo vardas teisingai įvesti, prisijungiama. Tada išsirenkama meniu sritis “Org. Informacija”. Šioje srityje organizacija gali patalpinti visą informaciją apie savo veiklą bei kontaktus. Kai

informacija įkelta, galima ją išsaugoti, taip pat uždaryti langą ir išsaugoti, taip pat uždaryti langą neišsaugant įkeltos informacijos.

Prekės įkėlimo į sąmatą veiklos diagrama pateikta 3.9 paveiksle:



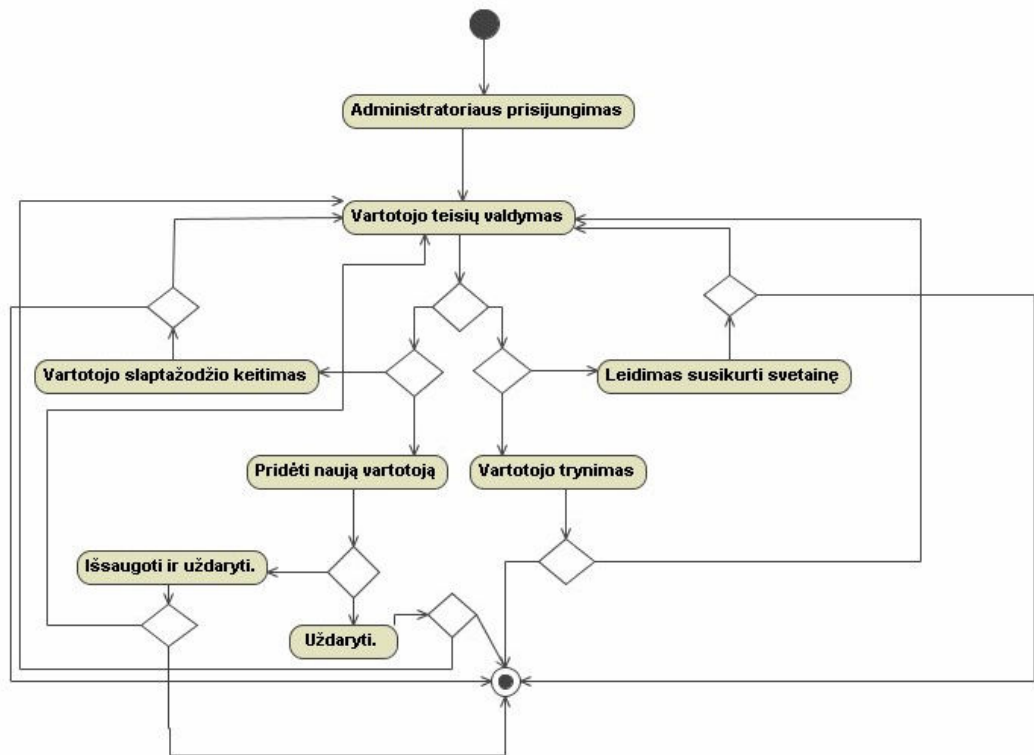
Pav 3.9 Prekės įkėlimas į sąmatą „activity“ diagrama

Jeigu mes norime įkelti naują prekę į savo sąmatą, mes turime: išsirinkti prekių katalogą, tada išsirinkti pačią prekę. Ją pasirinkus, galime įkelti prekę į sąmatą, uždaryti langą, neįkėlus prekės į sąmatą, taip pat perskaičiuoti sąmatą su nauja preke.

Administratoriui suteiktos teisės valdyti vartotojų informaciją. Taigi jis vartotojams gali suteikti arba atimti tam tikras teises:

1. Leisti susikurti svetainę
2. Pakeisti vartotojo slaptažodį
3. Sukurti naują vartotoją
4. Ištrinti esamą vartotoją

Administratoriui ležiamų valdyti teisių veiklos diagrama pateikta 3.10 paveiksle.



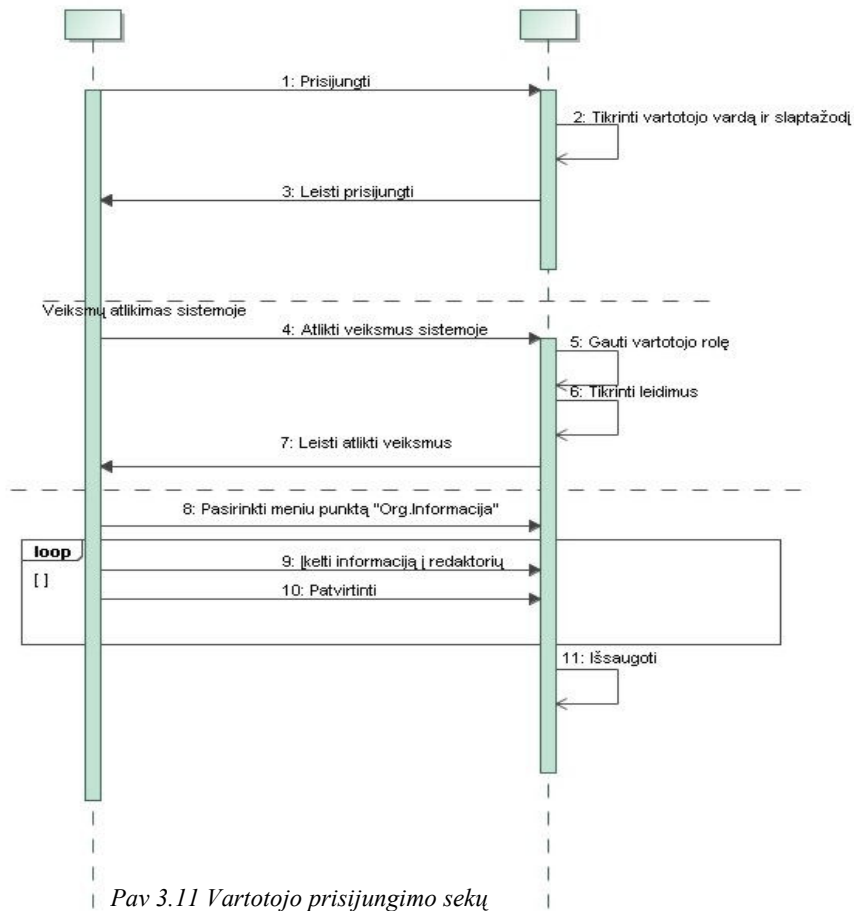
Pav 3.10 Administratoriaus vartotojui suteikiamų teisių „activity“ diagrama

3.3.2. Sekų diagramos

Sekų diagramos atvaizduoja tokią pačią informaciją kaip ir bendradarbiavimo diagramos. Jos sudarytos iš objektų, pranešimų, vertikalios laiko ašies. Sekų diagrama atvaizduoja pasiskirsčiusius laike įvykius.

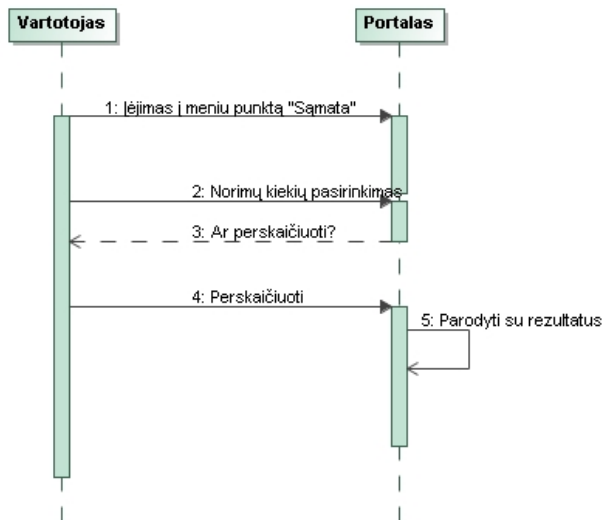
Šiuo atveju sistemos naudotojas užkrauna prisijungimo formą ir prisijungę prie SQL kliento bando autorizuotis. Jei autorizacija pavyksta, užsikrauna pagrindinė Forma, iš kurios yra pasiekiamas meniu.

Vartotojo prisijungimo sekų diagrama pateikta 3.11 paveiksle.



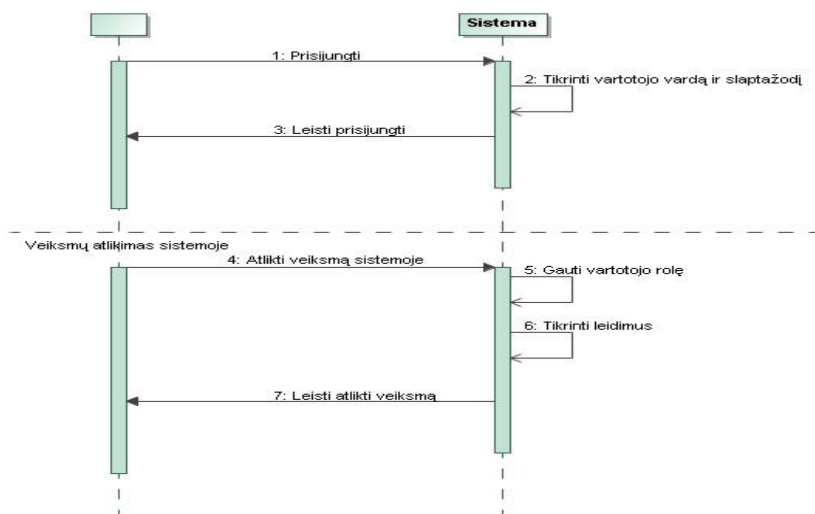
Pav 3.11 Vartotojo prisijungimo sekų

Organizacijos informacijos įkėlimo sekų diagrama pateikta 3.12 paveiksle.



Pav 3.12 Organizacijos informacijos įkėlimo sekų diagrama

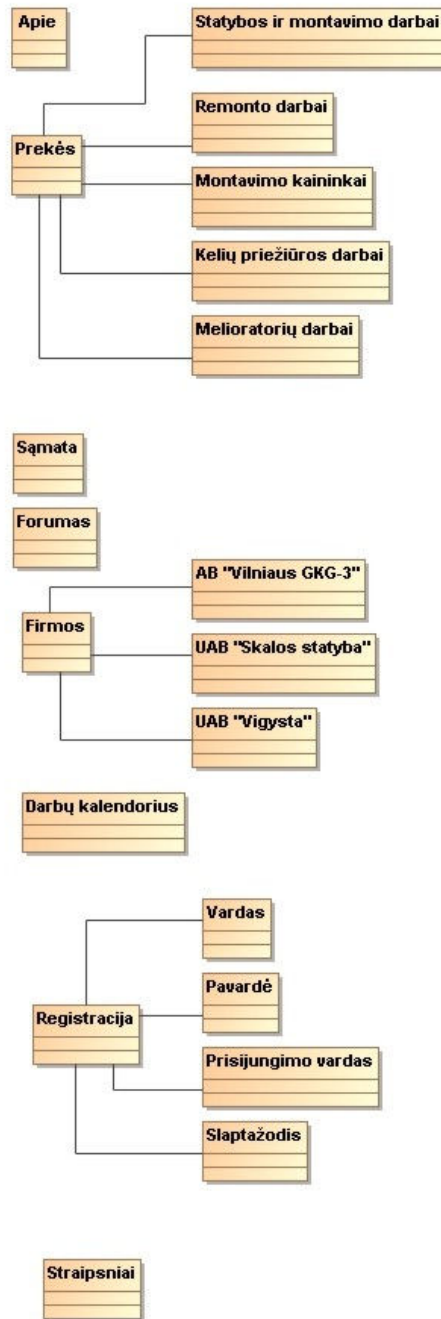
Sąmatos peržiūros sekų diagrama pateikta 3.13 paveiksle.



Pav 3.13 Sąmatos peržiūros sekų diagrama

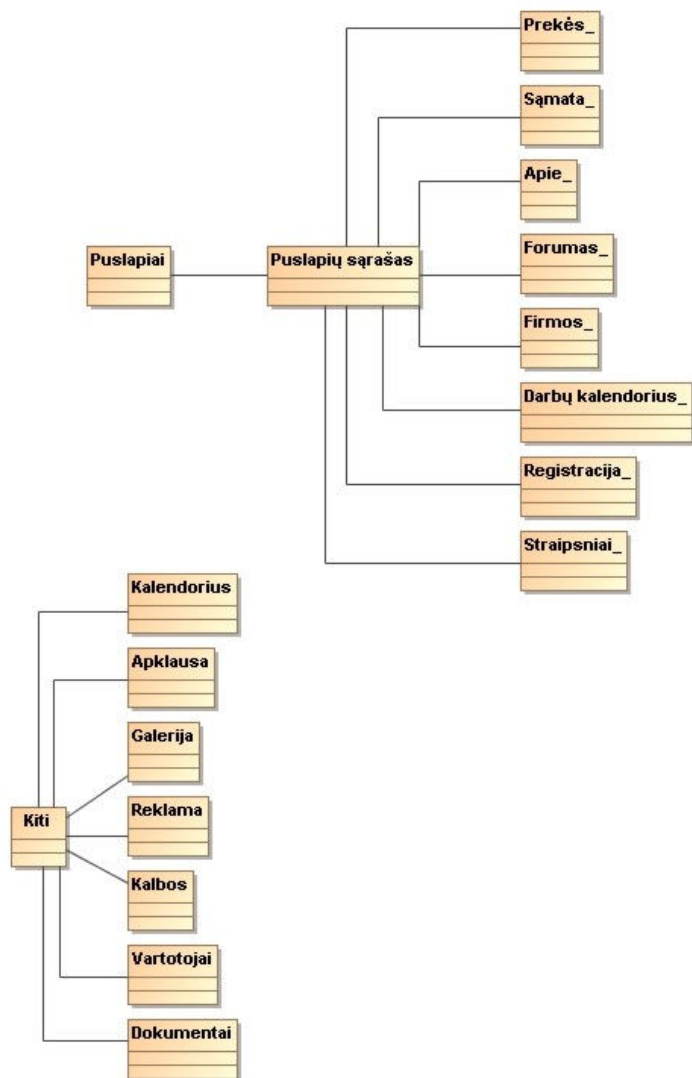
3.3.3. Portalo meniu klasių diagramos

Portalo meniu struktūra pateikta 3.14 paveiksle.



Pav 3.14 portalo meniu struktūra

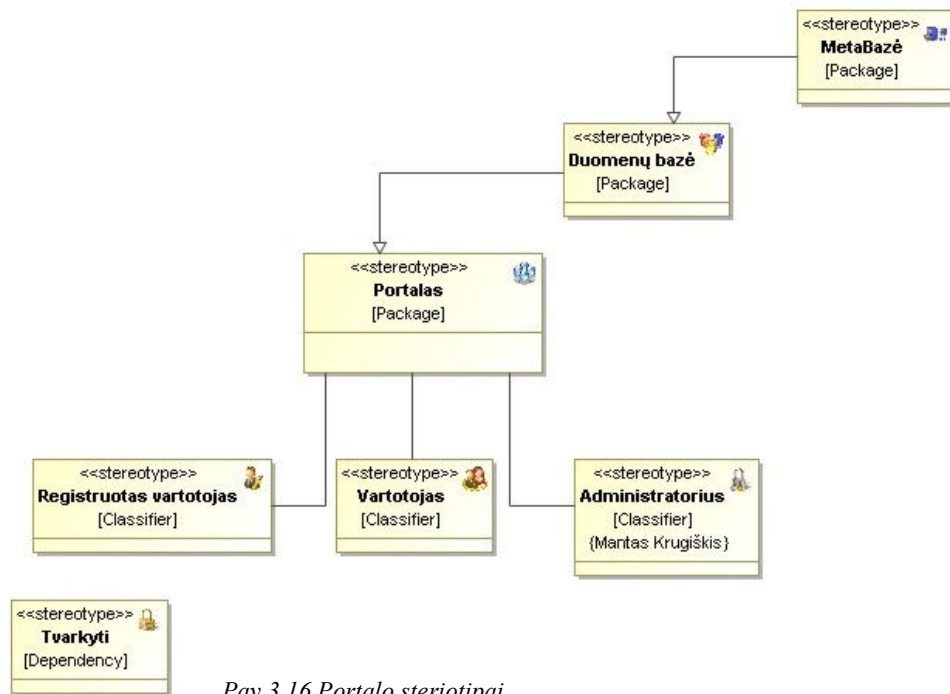
Portalo administravimo meniu struktūra yra pateikta 3.15 paveiksle.



Pav 3.15 administravimo meniu struktūra

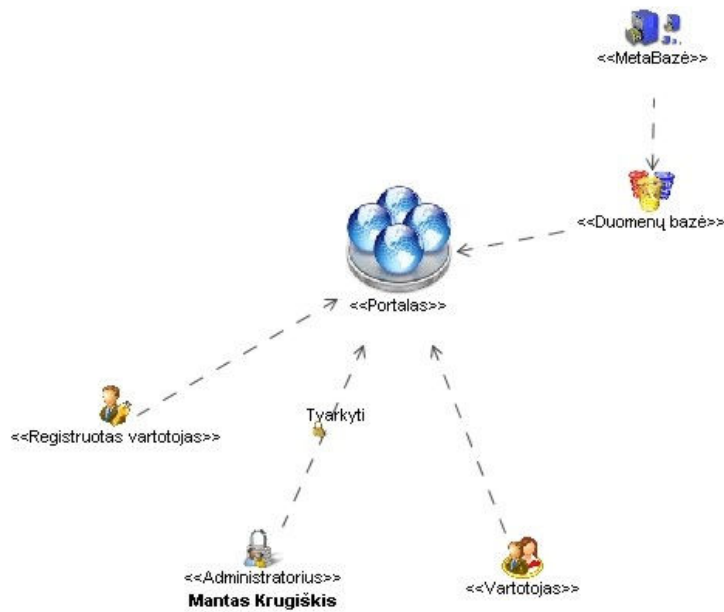
3.3.4. Portalo steriotipų diagramos

Portalo steriotipai pateikti 3.16 paveiksle.



Pav 3.16 Portalo steriotipai

Portalo vaizdiniai steriotipai pateikti 3.17 paveiksle.



Pav 3.17 Portalo vaizdiniai steriotipai

3.3.5. Sistemos architektūra - statinės struktūros modelis Vartotojui teikiamos paslaugos

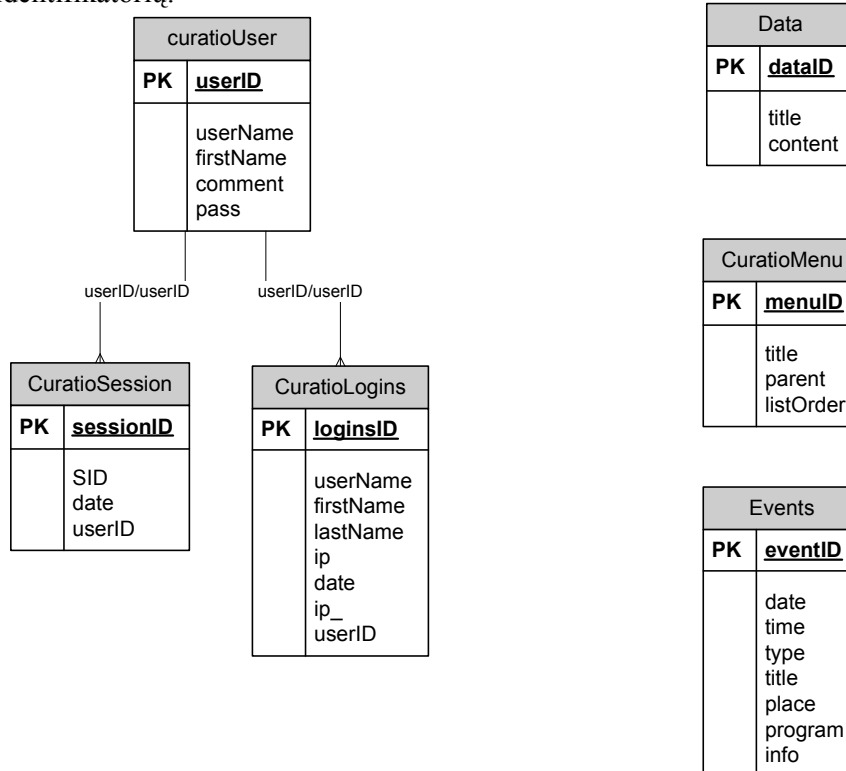
- ✚ Vartotojo registracija
- ✚ Šamatos skaičiavimas
- ✚ Šamatos saugojimas duomenų bazėje
- ✚ Portalo valdymas
- ✚ Portalo koregavimas
- ✚ Vartotojui paslaugas teikiančių įmonių sąrašas.

3.3.6. Duomenų bazės schema

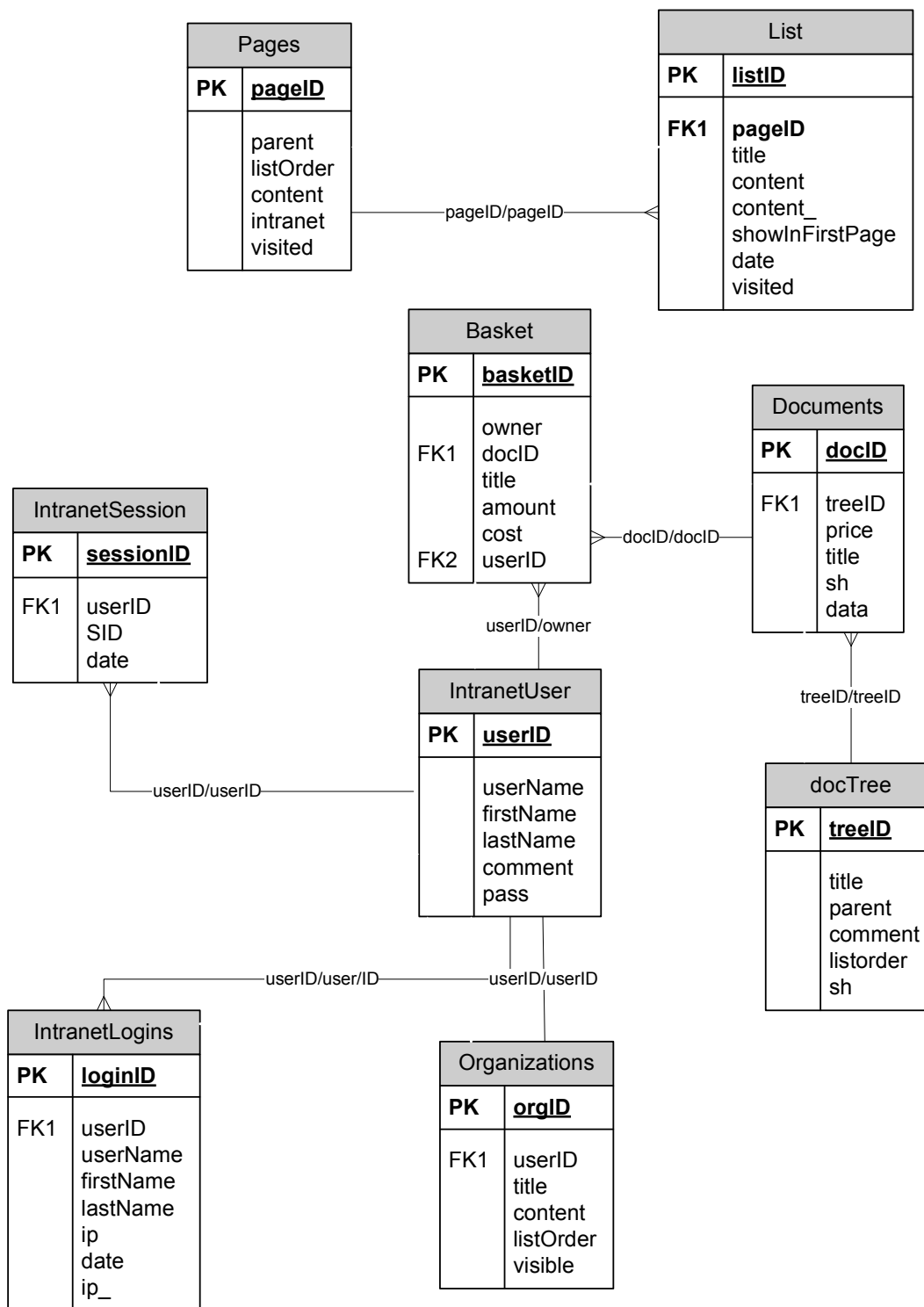
Pagrindiniai informacinėje sistemoje veikiantys objektai, jų savybės, atributai, priklausomybės tarp sistemos objektų.

ER diagrama

Susietos tarpusavyje 12 iš 15 lentelių. Likusios nesusietos ryšiais, nes jos tarpusavyje nepriklausomos. „Pages“ ir „List“ tarpusavyje susietos vienas su daug ryšiu, pagal puslapio „pageID“ identifikatorių.



Pav 3.18. DB schema 1



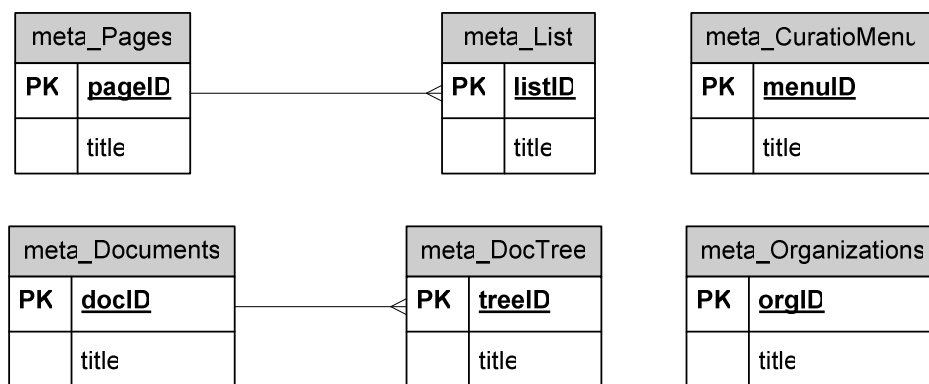
Pav 3.19. DB schema 2

„IntranetUsers“ ir „CuratioUsers“ lentelių ryšiai su lentelėmis, kuriose saugoma informacija apie vartotojų (IntranetSessions, CuratioSessions) sesijas ir jų prisijungimus (IntranetLogins, CuratioLogins) yra tokios pačios. Jos susietos su atitinkamomis lentelėmis pagal tuos pačius

identifikatorius „userID“ (lentelėse „IntranetUser“ ir „CuratioUser“ vartotojai skirtingi, todėl „userID“ – skirtingi) . „IntranetUsers“ su „Basket“- vartotojo krepšeliu yra susieta ryšiu vienas su daug, nes vienas vartotojas gali užregistruoti daug paslaugų. Pagal tą patį identifikatorių „IntranetUsers“ lentelė susieta su „Organizatos“ lentele ryšiu vienas su vienu, nes vienas vartotojas gali turėti vieną vienintelį organizacijos puslapį. Taip pat pagal tą patį identifikatorių ryšiu vienas su daug susietos „DocTree“ lentelė su lentele „Documents“ – kiekviename kataloge gali būti saugoma vienas ir daugiau dokumentų. O pagal identifikatorių „docID“ „Documents“ lentelė yra susiejama su lentele „Basket“ vienas su daug, nes daug vartotojų gali turėti tą pačią paslaugą savo krepšelyje.

ER metabazės diagrama

Metabazės esybių ryšių diagrama pateikta 3.20 paveiksle.



Pav 3.20. meta bazės esybių ryšių diagrama

Priklausomybės tarp sistemos objektų

Kiekvienas sistemos vartotojas turi unikalų savo identifikacinį numerį, pagal kurį jis susiejamas su jo paties duomenimis. Vienas vartotojas negali būti susietas su daugiau nei vienu įrašu vartotojų duomenų lentelėje. Kuomet vartotojas išsirenka naują paslaugą, tuomet „Basket“ lentelėje pažymima, kuris vartotojas įvedė duomenis. Vartotojas gali užregistruoti daug paslaugų. Sukūrus organizaciją (pvz. įmonę, firmą), jai priskiriamas unikalus identifikatorius, kuriuo pažymima, kad jis turi jam priklausomą puslapį ir yra užregistruojama jo vardu. Kiekviena organizacija gali keisti ir talpinti informaciją apie savo veiklą. Portalo sistemoje paslaugų katalogus („DocTree“) ir paslaugų aprašymus kuria tik administratorius. Jis pirmiausia sukuria paslaugų katalogą ir tame kataloge patalpina paslaugas, kurios yra susiejamos su juo. Registruotam vartotojui išsirinkus paslaugą iš paslaugų katalogo, į lentelę „Basket“ įrašomas įrašas „owner“, kuriuo pažymima, kokiam vartotojui priklauso ši išsirinkta paslauga.

„Data“, „Events“ lentelės nesusietos jokiais ryšiais, nes jos yra nepriklausomos nuo kitų duomenų.

3.4. Išvados

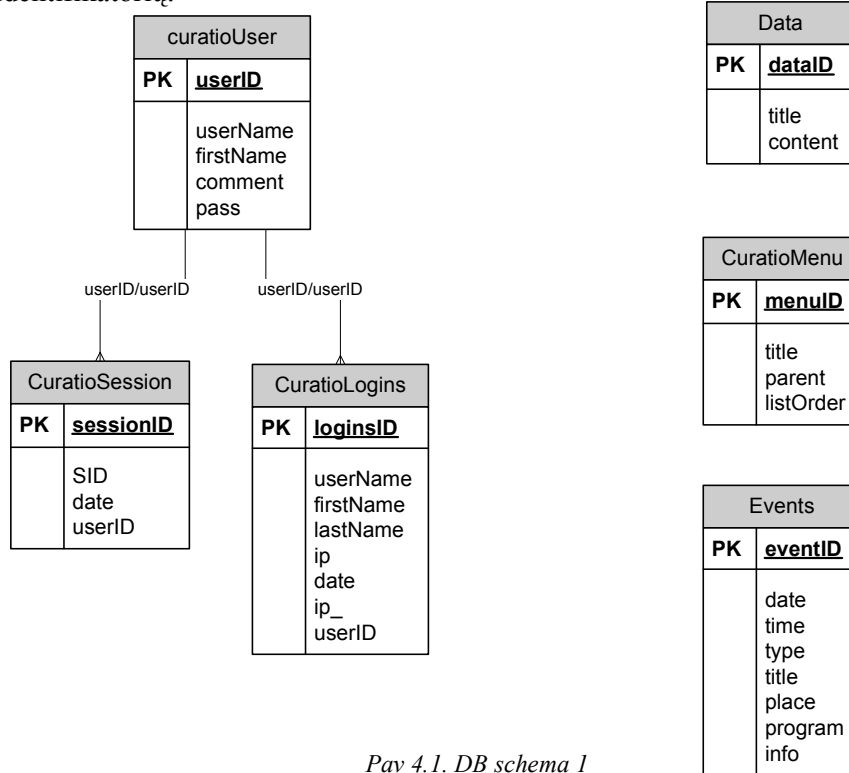
- Remiantis panašių sistemų analize ir vartotojų problemomis, iškelti pagrindiniai reikalavimai. Siekiant nustatyti portalo funkcionalumą nubraižytos ir specifikuotos panaudojimo atvejo veiklos diagramos.
- Remiantis apsibrėžtais reikalavimais, suprojektuota:
 - loginė architektūra;
 - sistemos elgsenos modelis, pateiktas sekų diagramomis;
 - reliacinės duomenų bazės schema;
 - sistemos testavimo modelis.
- Suprojektuotas metabazės lygis, kuriam esant daug lengviau perprojektuoti portalo struktūrą, pritaikant kitam tikslui bei tvarkant esamą portalo struktūrą.
- Atlikus palyginimą su panašios paskirties sistemomis pagal iškeltus tokių sistemų kriterijus galima daryti išvadą, kad projektuojama sistema bus pranašesnė tokių savybių atžvilgiu: greitesnis veikimas, stabilumas, suderinamumas, patogumas naudotis.

4. Realizacija

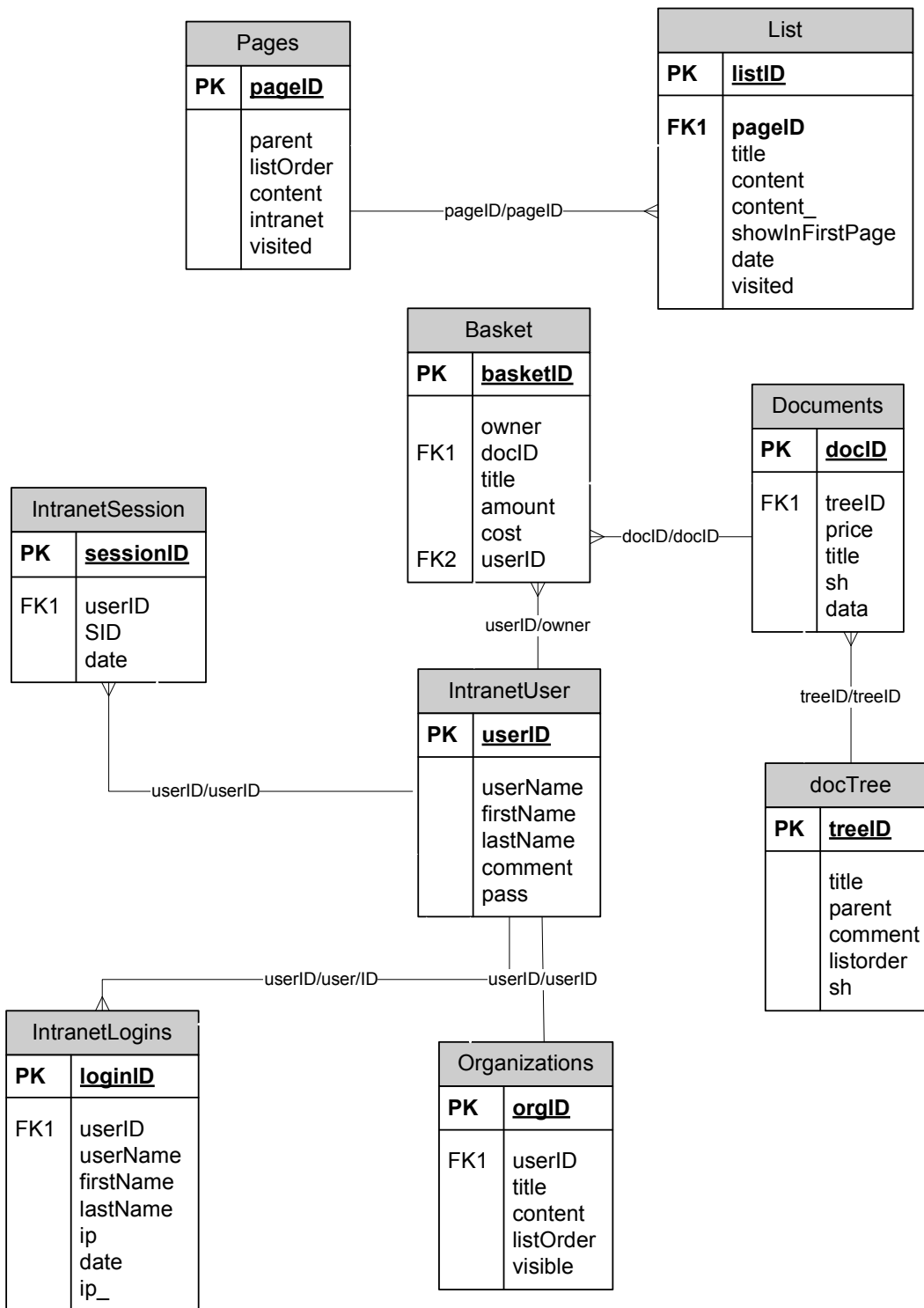
4.1. DB ryšių schema

4.1.1. Pagrindinės DB ryšių schema

Susietos tarpusavyje 12 iš 15 lentelių. Likusios nesusietos ryšiais, nes jos tarpusavyje nepriklausomos. „Pages“ ir „List“ tarpusavyje susietos vienas su daug ryšiu, pagal puslapio „pageID“ identifikatorių.



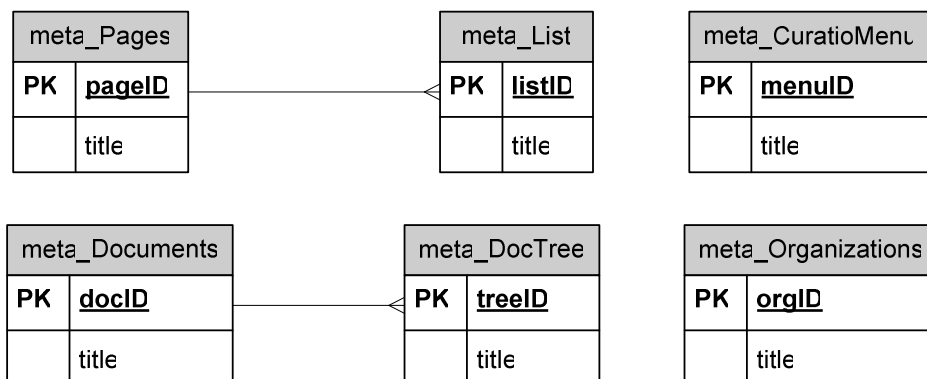
Pav 4.1. DB schema 1



Pav 4.2. DB schema 2

4.1.2. ER metabazės diagrama

Metabazės esybių ryšių diagrama pateikta 4.3 paveiksle.



Pav 4.3 meta bazės esybių ryšių diagrama

4.1.3. Pagrindiniai informacinės sistemos objektai, jų savybės, atributai Metabazėje

Metaduomenų bazėje saugoma informacija apie pagrindinėje bazėje esančius elementus. Metabazės lentelių turinys aprašytas žemiau.

„*meta_Pages*“ - Lentelėje skirta saugoti portalo puslapių duomenims, tokiems kaip, pavadinimas, vietos identifikatorius. (4.5 lentelė).

„*meta_List*“ - Lentelė skirta saugoti informaciją straipsnių, tokiems kaip pavadinimas, eilės numeris. (4.6 lentelė).

„*meta_CuratioMenu*“ - Lentelė skirta turinio valdymo sistemos struktūros duomenims saugoti, tokiems kaip, pavadinimas, vietos identifikatorius (4.4 lentelė).

„*meta_Documents*“ – Lentelė, kurioje saugoma informacija apie paslaugas : pavadinimas, vietos identifikatorius (4.3 lentelė).

„*meta_DocTree*“ - Lentelė skirta saugoti informacijai apie paslaugų katalogus, joje saugoma informacija apie katalogo pavadinimą, identifikatorių, kuris nurodo tėvo katalogo identifikacijos numerį. (4.2 lentelė).

„*meta_Organizations*“ - Lentelė skirta saugoti organizacijų duomenims, tokiems kaip pavadinimas, vartotojo identifikatorius. (4.1 lentelė).

4.1.4. Pagrindiniai informacinės sistemos objektai, jų savybės, atributai meta bazėje

4.1 lentelė. „meta_Organizations“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumai įvesti</i>	<i>tipas</i>	<i>Null</i>
orgID	Įrašo į lentelę bei registruoto vartotojo, kuriam suteikta organizacijos teisės, identifikatorius.	Ne	int(11) unsigned	Ne
Title	Organizacijos vardas, arba įmonės tipas (pvz.: UAB).	Ne	varchar(250)	Taip

4.2 lentelė. „meta_DocTree“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumai įvesti</i>	<i>tipas</i>	<i>Null</i>
TreeID	Įrašo į lentelę bei katalogo identifikatorius.	Ne	int(11) unsigned	Ne
title	Katalogo pavadinimas	Taip	varchar(255)	Taip

4.3 lentelė. „meta_Documents“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumai įvesti</i>	<i>tipas</i>	<i>Null</i>
docID	Įrašo į lentelę bei prekės identifikatorius.	Ne	int(10) unsigned	Ne
title	Prekės pavadinimas	Taip	varchar(255)	Taip

4.4 lentelė. „meta_CuratioMenu“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
menuID	Įrašo į lentelę bei Menu identifikatorius.	Ne	int(50) unsigned	Ne
title	Menu punkto pavadinimas.	Taip	char(200)	Taip

4.5 lentelė. „meta_Pages“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
pageID	Įrašo į lentelę bei portalo identifikatorių.	Ne	int(11) unsigned	Ne
title	Puslapio meniu punkto pavadinimas.	Taip	varchar(255)	Taip

4.6 lentelė. „meta_List“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
listID	Įrašo identifikatorius.	Ne	int(11) unsigned	Ne
title	Saugomi straipsniai skyrelio antraštės.	Taip	text	Taip

4.1.5. Pagrindiniai informacinės sistemos objektai, jų savybės, atributai paprastoje DB

Duomenų bazėje saugoma pagrindinė sistemos informacija. Duomenų bazės lentelių turinys aprašytas žemiau.

„Basket“ – Lentelė skirta saugoti kiekvieno vartotojo išsirinktų paslaugų krepšelio saugojimui (9.1 lentelė).

„IntranetUsers“ - Lentelė skirta saugoti registruoju vartotojų prisijungimo vardams, koduotiems slaptažodžiams ir kitiems tik prisijungimo metu naudojamiems darbiniam laukams (9.12 lentelė).

„IntranetLogin“ - Lentelė skirta saugoti prie sistemos prisijungusių vartotojų duomenims, tokiems kaip – vartotojo prisijungimo identifikatoriui, vartotojo id, vardui, pavardei ir ip adresui (9.11 lentelė).

„IntranetSessions“ - Lentelė skirta saugoti prie sistemos prisijungusių vartotojų sesijų duomenims, tokiems kaip – sesijos id, sesijos įrašui, datai, vartotojo id (9.10 lentelė).

„CuratioUsers“ - Lentelė skirta saugoti turinio valdymo sistemos administratorių prisijungimo vardams, koduotiems slaptažodžiams, ir kitiems tik prisijungimo metu naudojamiems darbiniam laukams (9.5 lentelė).

„CuratioLogin“ - Lentelė skirta saugoti prie turinio valdymo sistemos administratorių duomenims, tokiems kaip – administratoriaus prisijungimo identifikatoriui, vartotojo id, vardui, pavardei, ip adresui ir prisijungimo datai (9.2 lentelė).

„CuratioSessions“ - Lentelė skirta saugoti prie sistemos prisijungusių administratorių sesijų duomenims, tokiems kaip – sesijos id, sesijos įrašui, prisijungimo datai, vartotojo id (9.4 lentelė).

„Events“ - Lentelė skirta saugoti kalendoriaus duomenims, tokiems kaip – pavadinimas, turinys, renginio ar kito įvykio data, laikas, vieta (9.9 lentelė).

„CuratioMenu“ - Lentelė skirta turinio valdymo sistemos struktūros duomenims saugoti, tokiems kaip, pavadinimas, tėvo puslapio identifikatorius, vietos identifikatorius, turinys (9.3 lentelė).

„Data“ – Lentelė skirta kontaktų duomenis saugoti, tokiems kaip, pavadinimui ir turiniui (9.6 lentelė).

„DocTree“ - Lentelė skirta saugoti informacijai apie paslaugų katalogus, joje saugoma informacija apie katalogo pavadinimą, tėvo identifikatorių, kuris nurodo tėvo katalogo identifikacijos numerį, katalogo buvimo vietą ir indeksą, kuris nurodo, kokie vartotojai mato katalogus. (9.7 lentelė).

„Documents“ – Lentelė, kurioje saugoma visa informacija apie paslaugas : kainas, pavadinimas, komentaras, patalpinimo data (9.8 lentelė).

„List“ - Lentelė skirtas saugoti informaciją straipsnių, tokiems kaip pavadinimas, turinys, eilės numeris, data, apsilnakymo skaičius. (9.13 lentelė).

„Organizations“ - Lentelė skirta saugoti organizacijų duomenims, tokiems kaip pavadinimas, turinys, vartotojo identifikatorius. (9.14 lentelė).

„Pages“ - Lentelėje skirta saugoti portalo puslapių duomenims, tokiems kaip, tėvo puslapio identifikatorius, vietos identifikatorius, turinys. (9.15 lentelė).

4.2. Testavimo modelis bei duomenys, kontrolinis pavyzdys

4.7 lentelė testavimo modelis.

Kas testuojama?	Kaip testuojama?	Laukiami rezultatai	Realūs rezultatai
Prisijungimas prie sistemos.	1. Jungiamasi prie sistemos su klaidingais prisijungimo duomenimis. 2. Jungiamasi prie sistemos su teisingais administratoriaus duomenimis. 3. Jungiamasi prie sistemos su teisingais kliento duomenimis.	1. vartotojas gražinamas atgal į prisijungimų langą. 2. Užkraunama s administratoriaus darbalaukis 3. Užkraunama s kliento ataskaitų langas	1.vartotojas gražinamas atgal į prisijungimų langą. 2.Užkraunamas administratoriaus darbalaukis
Vartotojų duomenų apsauga.	Bandoma užkrauti operatoriaus ar administratoriaus darbalaukį per adresinę eilutę.	Vartotojas gražinamas į prisijungimo langą.	Vartotojas gražinamas į prisijungimo langą.
Naujų vartotojų registracija	1. Įrašomi esamo vartotojo duomenys 2. Įrašomas naujas vartotojas,	1. sistema neregistruoja vartotojo ir parodo	1. sistema neregistruoja vartotojo ir

	<p>tačiau skirtingi slaptažodžiai</p> <p>3. Įrašomi nauji tvarkingi duomenys.</p>	<p>atitinkamą pranešimą.</p> <p>2. Sistema neregistruoja vartotojo ir parodo atitinkamą pranešimą.</p> <p>3. Sistema sėkmingai užregistruoja naują vartotoją.</p>	<p>parodo atitinkamą pranešimą.</p> <p>2. Sistema neregistruoja vartotojo ir parodo atitinkamą pranešimą.</p> <p>3. Sistema sėkmingai užregistruoja naują vartotoją.</p>
--	---	---	--

4.3. Vartotojo sąsaja

4.3.1. Pagrindinis langas

Buvo sukurta vartotojo sąsaja, leidžianti intuityviai pasirinkti meniu punktus. Sąsaja suskirstyta į 4 dalis.



4.4 pav. Pagrindinis portalo langas

1 dalis. Vaizduojama informacija apie portalą. Identifikuojama kokiame portale yra vartotojas.

2 dalis. Pagrindinė dalis. Joje vaizduojama trečioje dalyje pasirenkamų meniu punktų informacija.

3 dalis. Talpinamas meniu.

4 dalis. Šioje dalyje yra talpinamos apklausos ir vartotojo prisijungimo meniu.



Meniu punktas su informacija apie portalą.

Meniu punktas su informacija apie prekes.

Meniu punktas su informacija apie sąmatą.

Meniu punktas su nuoroda į forumą.

Meniu punktas su informacija apie įmones.

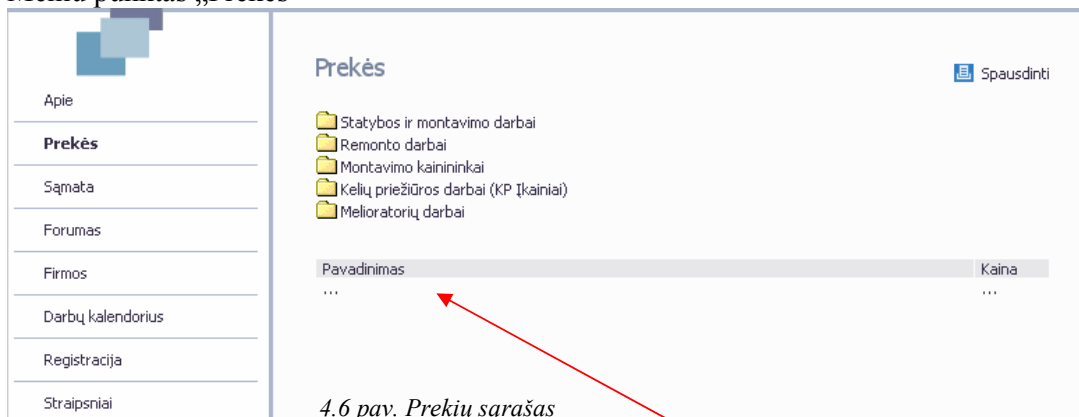
Meniu punktas su planuojamų darbų bei įvykių kalendoriumi.

Meniu punktas su registracijos struktūra.

Meniu punktas su esamų straipsnių sąrašu.

4.5 pav. Pagrindinis portalo meniu

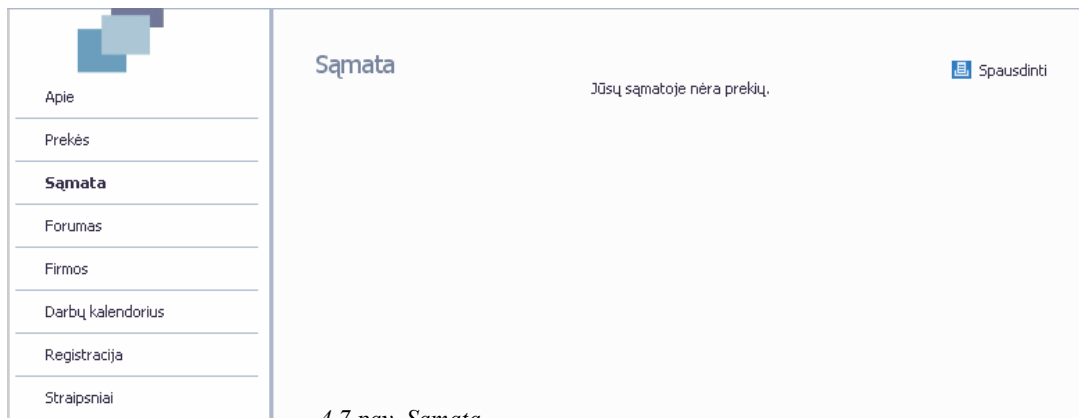
Meniu punktas „Prekės“



4.6 pav. Prekių sąrašas

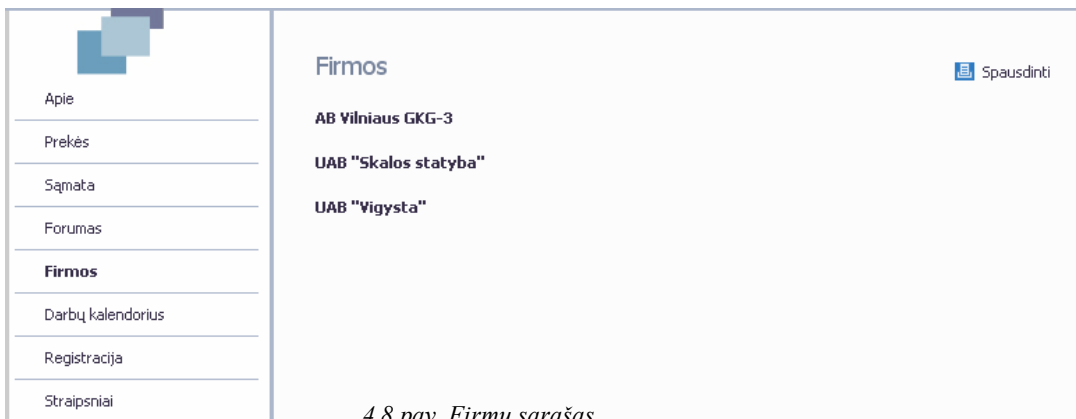
Skelbiama informacija apie prekių sąrašus. Šioje vietoje rodoma informacija apie prekes.

Meniu punktas „Sąmata“



4.7 pav. Sąmata

Meniu punktas „Firmos“



Firmos Spausdinti

AB Vilniaus GKG-3


UAB "Skalos statyba"

UAB "Vigysta"

4.8 pav. Firmų sąrašas

Skelbiama informacija apie susijusias įmones, taip pat skelbiama įmonių įkelta į portalą informacija.

Meniu punktas „Registracija“



Registracija Spausdinti

Prisijungimo vardas:

Slaptažodis:

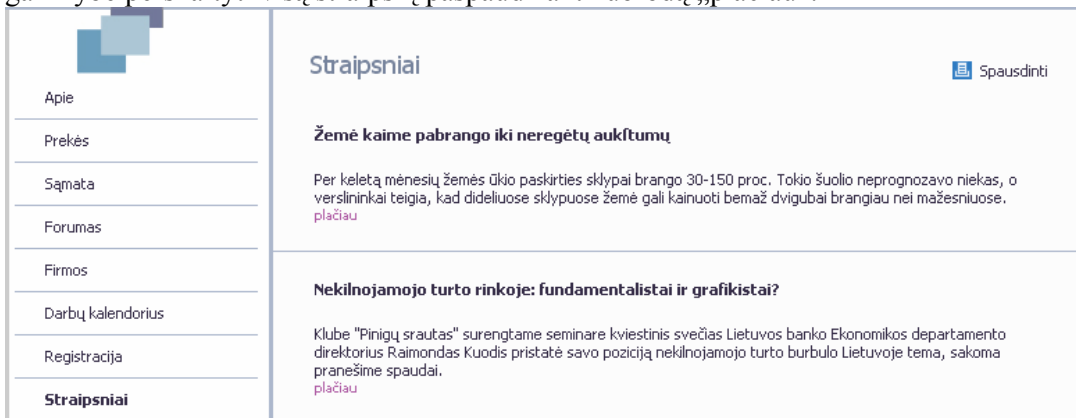
Pakartoti slaptažodį:

Vardas:

Pavardė:

4.9 pav. Registracijos forma

Meniu punktas „Straipsniai“. Pateikiamas sutrumpintas kiekvieno straipsnio variantas. Yra galimybė perskaityti visą straipsnį paspaudžiant nuorodą „plačiau“.



Straipsniai Spausdinti

Žemė kaime pabrango iki neregėtų aukštumų

Per keletą mėnesių žemės ūkio paskirties sklypai brango 30-150 proc. Tokio šuolio neprognozavo niekas, o verslininkai teigia, kad dideliuose sklypuose žemė gali kainuoti bemaž dvigubai brangiau nei mažesniuose. [plačiau](#)

Nekilnojamojo turto rinkoje: fundamentalistai ir grafikai?

Klube "Pinigų srautas" surengtame seminare kvietinis svečias Lietuvos banko Ekonomikos departamento direktorius Raimondas Kuodis pristatė savo poziciją nekilnojamojo turto burbulu Lietuvoje tema, sakoma pranešime spaudai. [plačiau](#)

4.10 pav. Straipsnių sąrašas

Prisijungusios organizacijos langas. Pateikiama bendra informacija apie galimybes.

Bendra informacija

Vartotojo gidas

Sąmatą susidėti yra paprasta. Tiesiog spaudžiate ant norimos paslaugos, tada atsirado langelis, kuriame yra mygtukas „Į sąmatą“, tada jį paspaudžiate ir paslauga bus įdedama į sąmatą? (1 pav.).

Į sąmatą - Microsoft Internet Explorer

Nelygumų ir iškyšų mūrinių sienų paviršiuje nukapojimas (kaina už 1 kvadratinį metrą)

Vieneto kaina: 49.93 Lt. Kiekis: 1 Viso: 49.93 Lt.

Į sąmatą Perskaičiuoti kainą Uždaryti langą

1 pav. Sąmatos formavimas

Prisijungęs AB "Vilniaus GKG-3" Asijungti

Apklausa

Ar jums patinka ši svetainė?

Jūs jau balsavote! Ačiū už nuomonę.

Rezultatai

4.11 pav. Prisijungusio vartotojo pradinis langas

Meniu punktas „Prekės“. Atvaizduojamas pilnas prekių sąrašas, grupuojant pagal tipus.

Prekės

- Statybos ir montavimo darbai
- Remonto darbai
- Montavimo kainininkai
- Kelių priežiūros darbai (KP įkainiai)
- Melioratorių darbai

Pavadinimas	Kaina
...	...

4.12 pav. Prisijungusio vartotojo prekių sąrašas

Prisijungęs AB "Vilniaus GKG-3" Asijungti

Apklausa

Ar jums patinka ši svetainė?

Jūs jau balsavote! Ačiū už nuomonę.

Rezultatai

Meniu punkte „Sąmata“ yra sudedama kiekviena meniu punkte „Prekės“ išsirinkta prekė. Taip pat yra galimybės įvesti kiekį, taip pat išmesti nereikalingą prekę ir perskaičiuoti sumą.

Sąmata

Prekė	Vieneto kaina	Kiekis	Bendra kaina	
Šlaitų apėjimas užpilant dirvožemiu (už 100 m ²)	10,00 Lt.	1	10,00 Lt.	Išmesti
Pakopų įrengimas 0,65 kubinio metro kaušo talpos ekskavatoriais ant skersinių nuolydžių, pakraunant gruntą į autosavivarčius, kaina už 1000 kubinių metrų	10,00 Lt.	1	10,00 Lt.	Išmesti
Plokščio tipo karkasai ir karkasinės konstrukcijos katilams, kurių garo našumas 1000 t/val	10,00 Lt.	1	10,00 Lt.	Išmesti
Sankasų viršaus ir ilaity lyginimas prikabinamais vidutiniais greideriais II grupės grunte (už 1000 m ²)	10,00 Lt.	1	10,00 Lt.	Išmesti
Spyrių (ramsčių) išsijų črengimas (kaina už 1m)	10,00 Lt.	1	10,00 Lt.	Išmesti
V grupės grunto kasimas 1 kubinio metro kaufo talpos ekskavatoriais, suverliant C sankasą, kaina už 1000 kubinių metrų	10,00 Lt.	8	80,00 Lt.	Išmesti

Iš viso: 130,00 Lt.

4.13 pav. Prisijungusio vartotojo saugoma sąmata

Prisijungęs AB "Vilniaus GKG-3" Asijungti

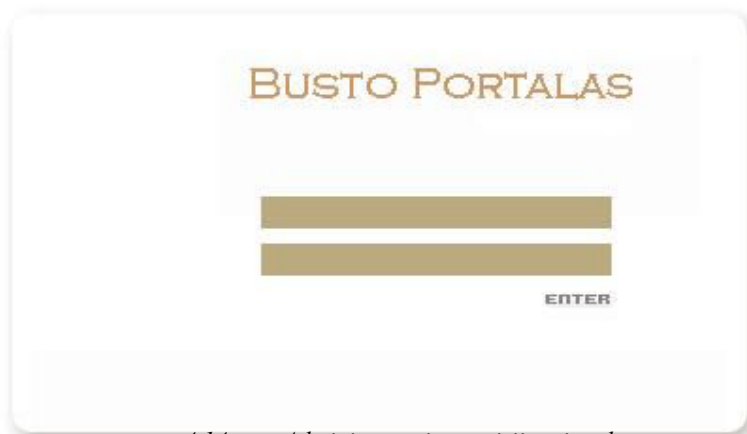
Apklausa

Ar jums patinka ši svetainė?

Jūs jau balsavote! Ačiū už nuomonę.

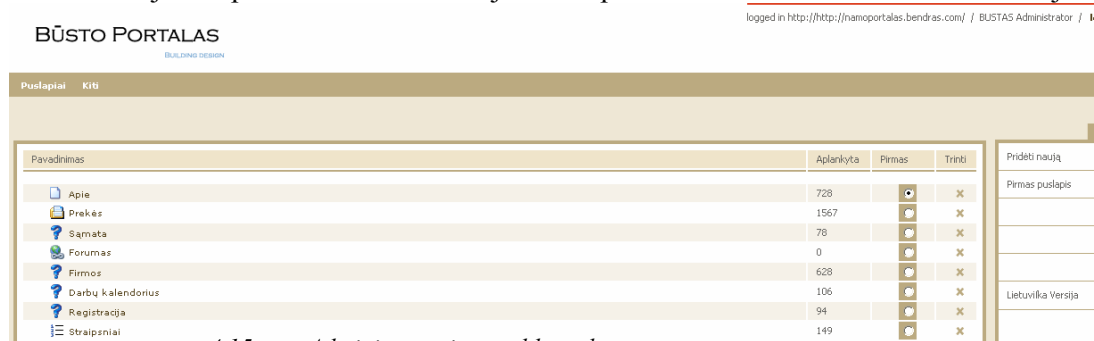
Rezultatai

Administratoriaus prisijungimo langas.



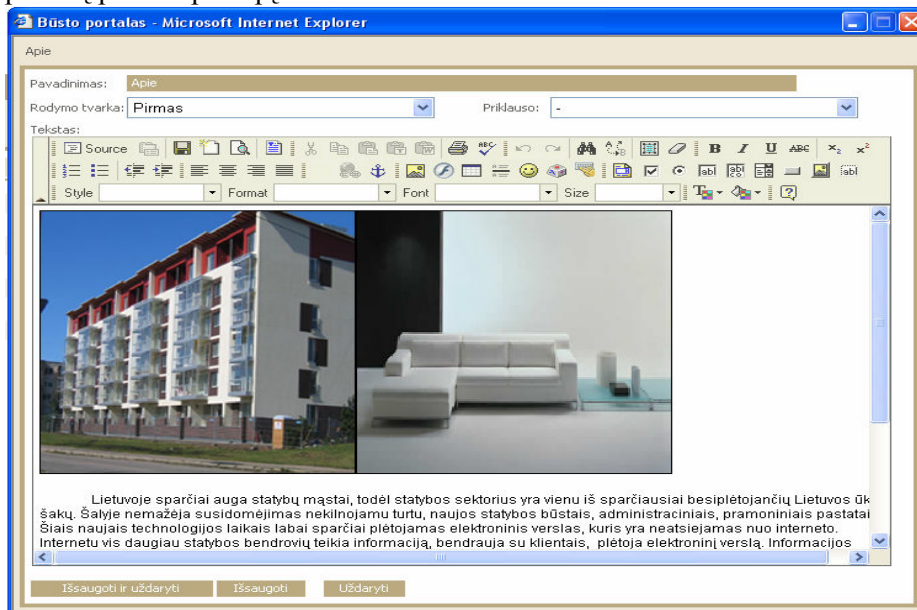
4.14 pav. Administratoriaus prisijungimo langas

Prisijungusio administratoriaus langas. Atvaizduojami du meniu, kurių pagalba administruojamas portalas. Kuriami nauji meniu punktai bei suteikiamos teisės vartotojams.



4.15 pav. Administratoriaus valdymo langas

Menu punkto „Apie“ redagavimas. Yra galimybė pasinaudojant redaktoriumi redaguoti pradinį portalo puslapį.



4.16 pav. Menu punkto „Apie“ informacijos koregavimas

Prekių sąrašo koregavimas, naujų prekių įterpimas bei koregavimas.

BŪSTO PORTALAS
BUILDING DESIGN

logged in http://http://namoportas.bendras.com/ / E

Puslapiai Kiti

Viešas Puslapis

- Statybos ir montavimo darbai
- Remonto darbai
- Montavimo kainininkai
- Kelių priežiūros darbai (KP Įkainiai)
- Melioratorių darbai

Naujas Katalogas Įdėti failą

Pavadinimas	Kaina
...	...

4.17 pav. Prekių sąrašo koregavimas

Straipsnių sąrašo koregavimas bei naujų įkėlimas. Senų straipsnių archyvavimas/

BŪSTO PORTALAS
BUILDING DESIGN

logged in http://http://namoportas.bendras.com/ / BU

Puslapiai Kiti

Straipsniai

Pavadinimas	Aplankyta		Trinti
Žemė kaime pabrango iki neregėtų aukštumų (2006-04-28)	11	Į archyvą	✕
Nekilnojamojo turto rinkoje: "fundamentalistai" ir "grafikistai" (2006-04-28)	10	Į archyvą	✕
Nekilnojamojo turto kainų ???burbulas??? Lietuvoje. Ar tikrai? (2006-04-24)	14	Į archyvą	✕
Ar pabrangs studentui buto nuoma? (2006-04-24)	10	Į archyvą	✕
Archiyvas			Trinti
Tuščia			

4.18 pav. Straipsnių sąrašo koregavimas

Administratoriaus meniu.

- Kalendorius
- Apklausa
- Kalbos
- Reklama
- Dokumentai
- Vartotojai
- Galerija

4.19 pav. Administratoriaus meniu

Menui punktą „Kalendorius“. Galimybė įterpti naują skelbimą, užduotį ar pranešimą tam tikrai dienai.

BŪSTO PORTALAS
BUILDING DESIGN

logged in http://http://namoportas.bendras.com/ / BU

Puslapiai Kiti

kalendorius:

- Kalendorius
- Apklausa
- Kalbos
- Reklama
- Dokumentai
- Vartotojai
- Galerija

Renginiai

2006 gruodis

R	Antradienis	Trečiadienis	Ketvirtadienis	Penktadienis	Šeštadienis	Sekmadienis
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

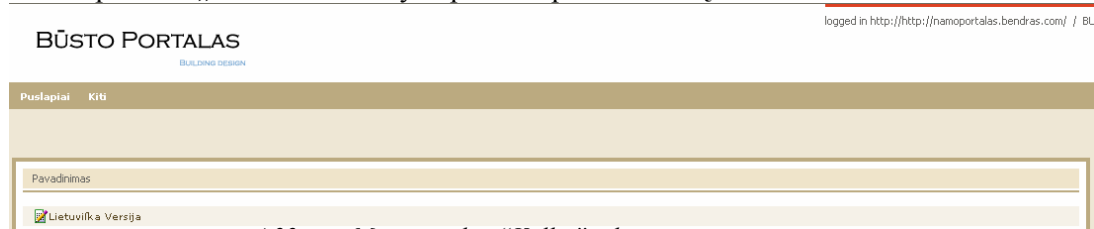
4.20 pav. Menui punkto „Kalendorius“ administravimas

Meniu punktas „Apklausa“. Galimybė aktyvuoti/deaktyvuoti apklausą/ Sukurti nauja, ištrinti seną.



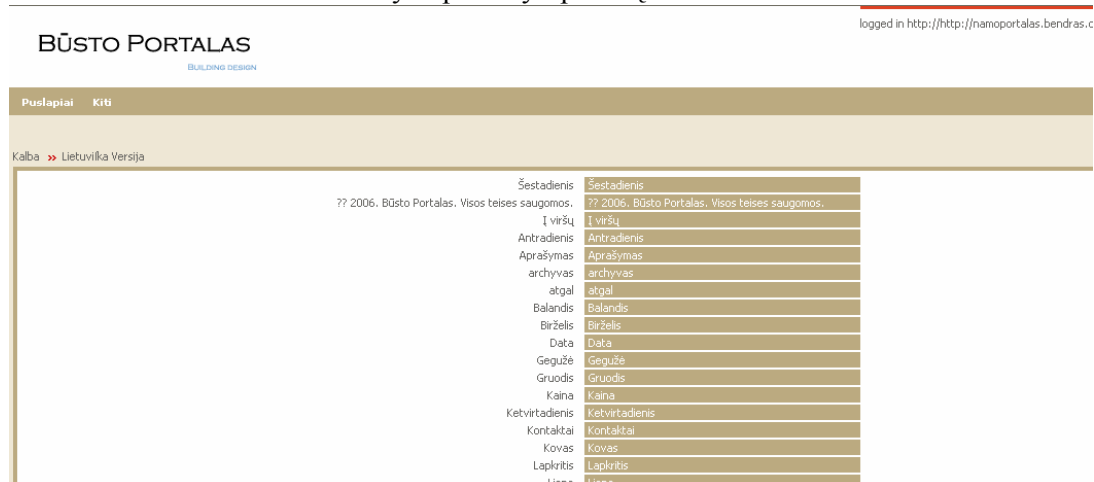
4.21 pav. Meniu punkto „Apklausa“ administravimas

Meniu punktas „Kalbos“. Galimybė parinkti portalo kalbą.



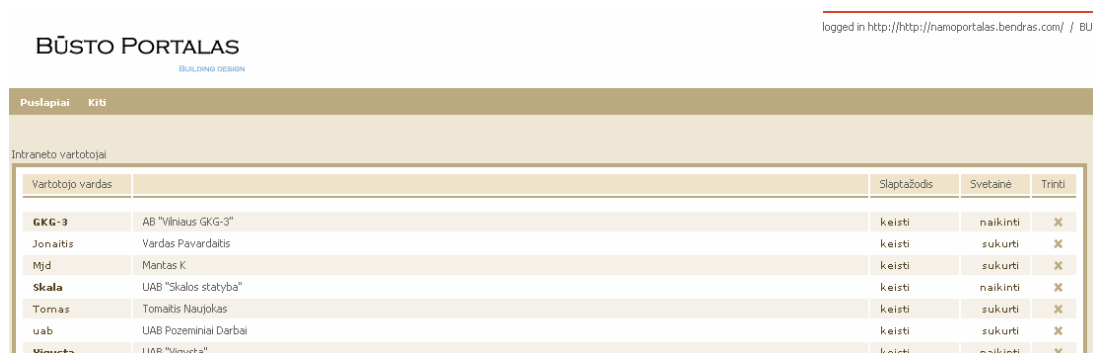
4.22 pav. Meniu punkto „Kalba“ administravimas

Kalbos administravimas. Galimybė pritaikyti portalą kitai kalbai.



4.23 pav. Portalo kalbos administravimas

Meniu punktas „Vartotojai“. Galimybė administruoti vartotojus, suteikti jiems teisę sukurti savo svetainę, ištrinti vartotoją.



4.24 pav. Portalo vartotojų administravimas

4.4. Bandomosios eksploatacijos rezultatų aprašymas

Kuriant programinę įrangą siekiama sukurti produktą turintį kuo mažiau klaidų ir defektų. Programinės įrangos testavimas gali parodyti klaidas, bet ne jų nebuvimą. Testavimo tikslas atskleisti kaip galima daugiau programinės įrangos klaidų, kad jas vėliau būtų galima ištaisyti. Tai leidžia užsakovui pateikti produktą su mažesniu klaidų kiekiu. Kitas svarbus testavimo tikslas yra patikrinti ar sukurta programinė įranga atitinka specifikaciją ir vartotojų reikalavimus. Skyriuje pateikiami testavimo atvejai leidžiantys užtikrinti minimalų informacinės sistemos klaidų kiekį ir aukštą kuriamos programinės įrangos kokybę.

4.4.1. Testavimo apimtis

Kad užtikrinti aukštą sistemos kokybę bus atliekami įvairaus lygio testavimai, pradedant atskirų vienetų testavimu, integravimo testavimu ir baigiant vartotojo sąsajos testavimu.

4.4.2. Pagrindiniai apribojimai

Pagrindinis apribojimas, kuris gali įtakoti testavimo plano nevykdyma tai patirties trūkumas.

4.4.3. Testavimo planas

Šiame skyriuje pateikiamos testavimo procedūros ir strategija, kurių pagalba testuojant programinę įrangą galima pasiekti užsibrėžtus testavimo tikslus.

4.4.4. Testavimo strategija

Skyriuje apžvelgiamos pasirinktos testavimo strategijos.

Vienetų testavimas: tikrinama ar pagal perduotus pradinis duomenis gaunami laukiami rezultatai. Duomenys parenkami neatsižvelgiant į struktūrą.

4.4.5. Testavimo resursai

Techninė ir programinė įranga: jokių specialių papildomų techninių ar programinių resursų nereikia. Sistemos testavimas atliekamas naudojantis vartotojo sąsaja, jai paduodant duomenis.

4.4.6. Testavimo procedūros

Skyriuje pateikiamos testavimo procedūros, kurios bus naudojamos atliekant testavimą.

-vienetų testavimas

4.8 lentelė .Prisijungimo testavimo atvejai

Testas	Laukiamas rezultatas
Įvedami teisingi prisijungimo duomenys	Parodomas prisijungusio vartotojo langas su vartotojo teises atitinkančiais menu punktais
Įvedami neteisingi prisijungimo duomenys	Vartotojui neleidžiama prisijungti
Atsijungiama nuo vartotojo srities, pasirenkant mygtuką "Atsijungti"	Vartotojas atjungiamas nuo jo srities ir atidaromas pagrindinis portalo langas.

4.9 lentelė. Pagrindinio lango testavimo atvejai

Testas	Laukiamas rezultatas
Pasirenkamas meniu punktas "Apie"	Atidaromas pagrindinis portalo langas su pradine informacija
Pasirenkamas meniu punktas "Prekės"	Atidaromas prekių sąrašas su aprašymu ir kainomis
Pasirenkamas meniu punktas "Šamata"	Atidaromas langas su šiuo metu šamatoje esančiomis prekėmis ir pilna jų kaina
Pasirenkamas meniu punktas "Forumas"	Atidaromas naujas langas su forumu ir jo meniu punktais.
Pasirenkamas meniu punktas "Firmos"	Atidaromas langas su firmų sąrašu.
Pasirenkamas meniu punktas "Darbų kalendorius"	Atidaromas langas su kalendoriumi ir pagrindinių įvykių žymomis.
Pasirenkamas meniu punktas "Registracija"	Atidaromas langas su registracijos forma.
Pasirenkamas meniu punktas "Straipsniai"	Atidaromas langas su straipsnių sąrašu.

4.10 lentelė. Administratoriaus posistemės prisijungimo testavimo atvejai

Testas	Laukiamas rezultatas
Įvedami teisingi prisijungimo duomenys	Parodomas prisijungusio administratoriaus langas
Įvedami neteisingi prisijungimo duomenys	Neleidžiama prisijungti
Atsijungiama nuo administratoriaus srities, pasirenkant mygtuką "logout"	Administratorius atjungiamas nuo jo srities ir atidaromas administratoriaus prisijungimo langas.

4.11 lentelė. Administratoriaus posistemės testavimo atvejai

Testas	Laukiamas rezultatas
Pasirenkamas meniu punktas "Puslapiai" → "Puslapių sąrašas"	Atidaromas administruojamų meniu sąrašas, kuris rodomas paprastam vartotojui.
Pasirenkamas meniu punktas "Kiti" → "Kalendorius"	Atidaromas kalendoriaus langas su papildomu meniu jo administravimui.
Pasirenkamas meniu punktas "Kiti" → "Apklausa"	Atidaromas apklausos langas su papildomu meniu jos administravimui.
Pasirenkamas meniu punktas "Kiti" → "Kalbos"	Atidaromas kalbos administravimo langas su papildomu meniu jos administravimui.
Pasirenkamas meniu punktas "Kiti" → "Reklama"	Atidaromas reklamos administravimo langas. Galimybė pateikti tam tikrą reklaminę medžiagą portale.
Pasirenkamas meniu punktas "Kiti" → "Dokumentai"	Atidaromas dokumentų redagavimo langas.
Pasirenkamas meniu punktas "Kiti" → "Vartotojai"	Atidaromas vartotojų administravimo langas.

4.12 lentelė. Užduočių įkėlimas į kalendorių.

Testas	Laukiamas rezultatas
Naujos užduoties įkėlimas į kalendorių. Spaudžiamas meniu punktas "Pridėti naują"	Atsidaro naujas langas su užduoties forma

4.13 lentelė. Naujos apklausos įkėlimas

Testas	Laukiamas rezultatas
Atsidarius meniu punktą "Kiti" → "Apklausa" pasirenkamas šoniniame meniu punktas „Pridėti naują“	Atsidaro naujas langas su apklausos forma.

4.4.7. Testavimo rezultatai

Žemiau lentelėse pateikiami atskirų vienetų atlikti testavimo atvejai ir rezultatai.

4.14 lentelė. Prisijungimo testavimo atvejai

Testas	Laukiamas rezultatas	Rezultatas
Įvedami teisingi prisijungimo duomenys	Parodomas prisijungusio vartotojo langas su vartotojo teises atitinkančiais meniu punktais	Pavyko
Įvedami neteisingi prisijungimo duomenys	Vartotojui neleidžiama prisijungti	Pavyko
Atsijungiama nuo vartotojo srities, pasirenkant mygtuką "Atsijungti"	Vartotojas atjungiamas nuo jo srities ir atidaromas pagrindinis portalo langas.	Pavyko

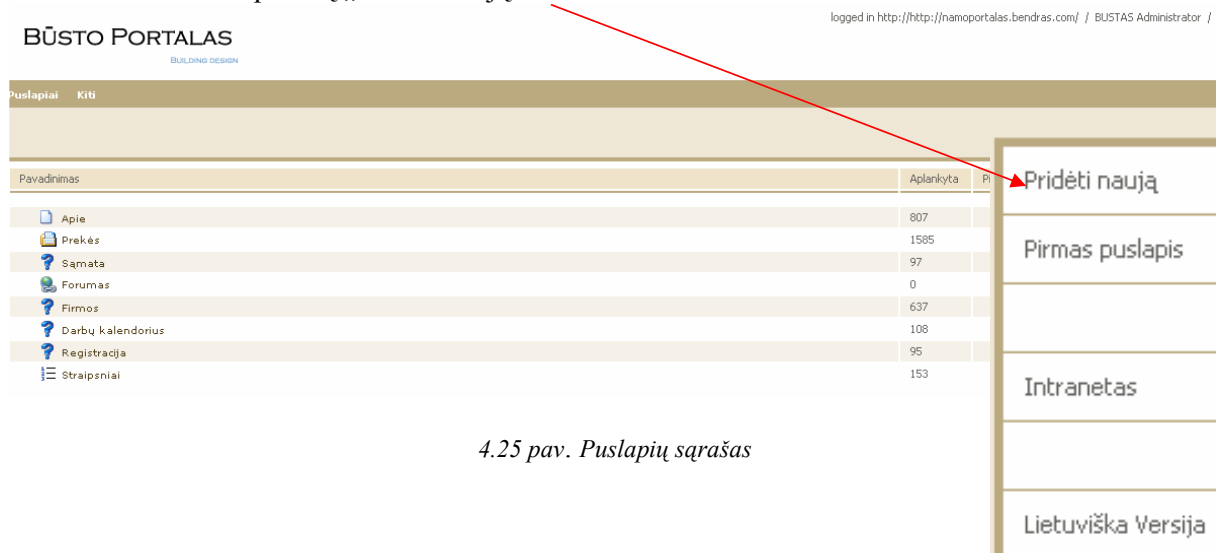
4.15 lentelė. Pagrindinio lango testavimo atvejai

Testas	Laukiamas rezultatas	Rezultatas
Pasirenkamas meniu punktas "Apie"	Atidaromas pagrindinis portalo langas su pradine informacija	Pavyko
Pasirenkamas meniu punktas "Prekės"	Atidaromas prekių sąrašas su aprašymu ir kainomis	Pavyko
Pasirenkamas meniu punktas "Šamata"	Atidaromas langas su šiuo metu šamatoje esančiomis prekėmis ir pilna jų kaina	Pavyko
Pasirenkamas meniu punktas "Forumai"	Atidaromas naujas langas su forumu ir jo meniu punktais.	Pavyko
Pasirenkamas meniu punktas "Firmos"	Atidaromas langas su firmų sąrašu.	Pavyko
Pasirenkamas meniu punktas "Darbų kalendorius"	Atidaromas langas su kalendoriumi ir pagrindinių įvykių žymomis.	Pavyko
Pasirenkamas meniu punktas "Registracija"	Atidaromas langas su registracijos forma.	Pavyko
Pasirenkamas meniu punktas "Straipsniai"	Atidaromas langas su straipsnių sąrašu.	Pavyko

Testas	Laukiamas rezultatas	Rezultatas
Įvedami teisingi prisijungimo duomenys	Parodomas prisijungusio administratoriaus langas	Pavyko
Įvedami neteisingi prisijungimo duomenys	Neleidžiama prisijungti	Pavyko
Atsijungiama nuo administratoriaus srities, pasirenkant mygtuką "logout"	Administratorius atjungiamas nuo jo srities ir atidaromas administratoriaus prisijungimo langas.	Pavyko

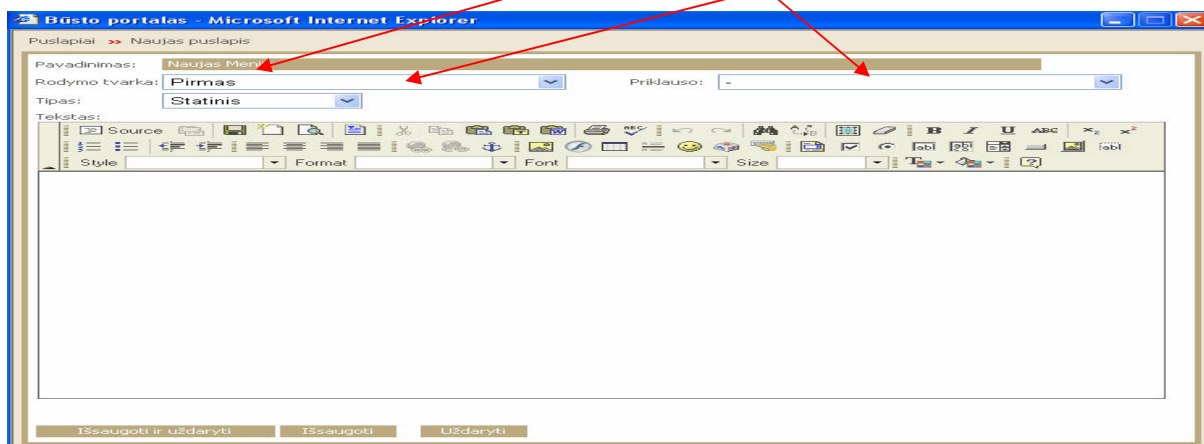
4.4.8. Naujo meniu punkto įterpimas

Atsidarę turinio valdymo sistemą, pasirenkame puslapių, esančių portale sąrašą. Pasirenkame meniu punktą „Pridėti naują“.





4.25 pav. Puslapių sąrašas

Atsidariusiame lange užrašome naujojo meniu punkto pavadinimą „Naujas Meniu“. Kelintas jis bus rodomas, ar priklausys kuriam nors jau egzistuojančiam meniu.



4.26 Pav Naujo meniu sudarymas

Paspaudus mygtuką „Įterpti“ į meniu sąrašą įsirašo naujas meniu elementas.

 Naujas Meniu	0		
 Apie	807		
 Prekės	1585		
 Šamata	97		
 Forumas	0		
 Firmos	637		
 Darbų kalendorius	108		
 Registracija	95		
 Straipsniai	153		

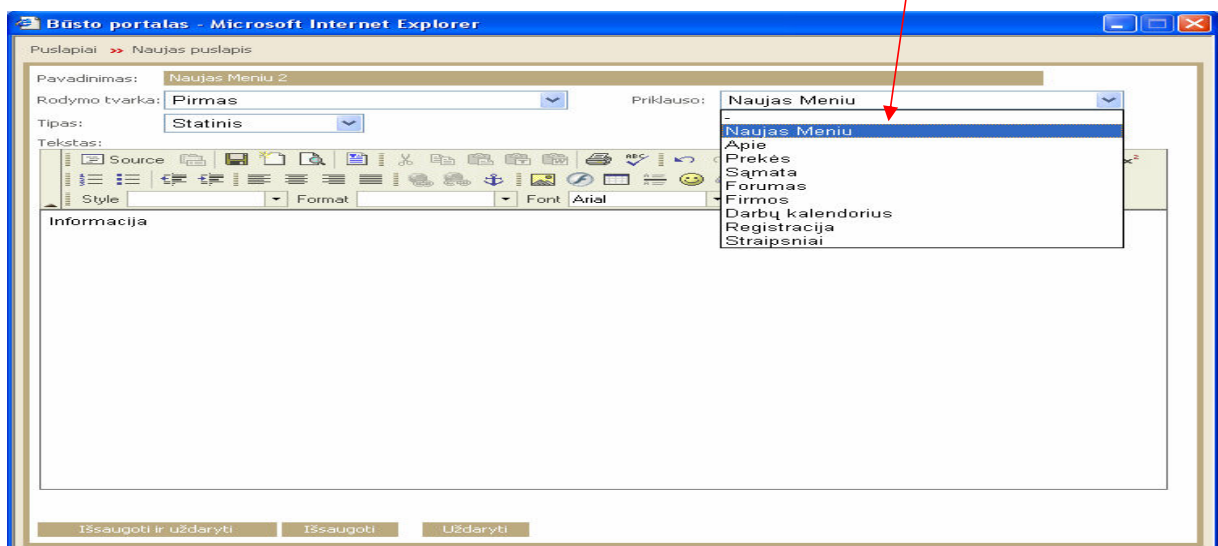
4.27 pav. Naujo meniu įterpimas

Taip pat naujas meniu elementas atsiranda ir portale.



4.28 pav Įterptas naujas meniu į portalą

Sukuriame antro lygio meniu punktą. Priskiriame jį jau sukurtam naujam meniu.



4.29 pav Naujo meniu poskyrio priskyrimas

Portale atsiranda dviejų lygių meniu.



4.30 pav. Meniu hierarchija

4.5. Išvados

1. Remiantis panašių sistemų analize ir vartotojų problemomis, pasinaudojant MySQL duomenų bazių valdymo sistema, realizuota sistemos duomenų bazė.
2. Realizuota metaduomenų bazė, kurios pagalba atsirado galimybė lengviau specifikuoti portalą:
 - Įterpti naują meniu punktą tiesiai per metabazę
 - Galimybė vartotojui lengviau dalyvauti portale
3. Suprojektuotas metaduomenų lygis leidžia patogesnę naudojimąsi portalui administruoti.
 - a. Lengviau pakeičiama duomenų struktūra
 - b. Lengvesnis vartotojų administravimas
 - c. Patogesnis informacijos įkėlimas
4. Metabazės lygis leidžia greičiau pasiekti duomenis
5. Metabazės lygis leidžia administratoriui lengviau ir greičiau perprojektuoti visą portalą kitam tikslui. (pasikeitus portalo informacijos sričiai)
6. Naudojantis metabazės lygiu tapo lengviau palaikyti ir plėtoti portalą.
7. Lengvesnis struktūros praplėtimas (pvz. Naujo meniu įkėlimas)
8. Lengviau naujam administratoriui prižiūrėti portalą, nes atskirta veikimo logika nuo duomenų.

5. Sistemos vartotojo dokumentas

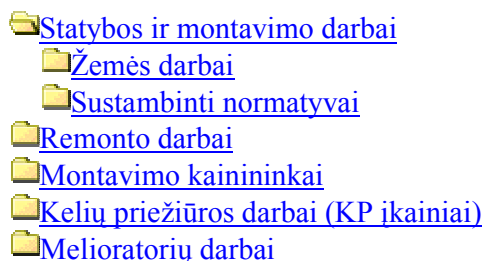
5.1. Neregistruotas vartotojas

Kiekvienas vartotojas atsidaręs Būsto Portalą iškart pamato pagrindinę struktūrą ir informaciją. Jis gali naršyti portale ir susidėti sąmatą naudodamasis meniu.

- Apie
- Prekės
- Forumas
- Firmos
- Kalendorius
- Registracija
- Straipsniai

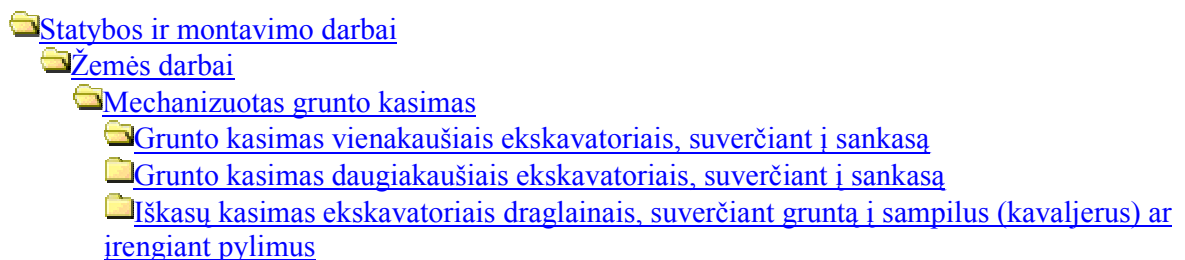
Punkte „Apie“ yra pateikiama pagrindinė informacija apie portalą.

„Prekės“ dalyje galima pamatyti visą paslaugų ir prekių sąrašą su kainomis bei susidėti sąmatą. (Paprastas, ne registruotas vartotojas, negali susidaryti savo sąmatos). Visą informaciją galima pamatyti išskleidžiant visą medžio tipo struktūrą: spaudžiant ant katalogų ir einant gilyn (5.1 pav.)



5.1. pav. Katalogų sandara

Išsirinkus paslaugų tipą ir einant gilyn katalogais, kol pasiekiami pagrindinė informacija(5.2 pav):



- [Vandens nutekėjimo prizmių ir griovių įrengimas iškasose](#)
- [Neiškastu grunto išėmimas iš mechanizuotai iškastų iškasų](#)
- [Išilginių vandens nuvedimo griovių kasimas](#)
- [Grunto kasimas ekskavatoriumi, pakraunant į autosavivarčius](#)
- [Pakopų įrengimas, pakraunant gruntą į autosavivarčius](#)
- [Laikinių grunto pervežimo kelių remontas ir priežiūra](#)
- [Skydų po ekskavatoriais ir pakloto po autotransporto priemonėmis įrengimas ir priežiūra](#)
- [Grunto kasimas skreperiais](#)
- [Grunto kasimas ir perstūmimas](#)
- [Tranšėjų ir duobių užpylimas iš sankasos](#)
- [Tranšėjų užpylimas iš sampilų \(kavaljerų\)](#)
- [Pirminis žemės paviršiaus planavimas buldozeriais](#)
- [Grunto purenimas](#)
- [Grunto kasimas rankiniu būdu](#)
- [Sustambinti normatyvai](#)
- [Remonto darbai](#)
- [Montavimo kainininkai](#)
- [Kelių priežiūros darbai \(KP įkainiai\)](#)
- [Melioratorių darbai](#)

Pavadinimas	Kaina
V grupės grunto kasimas 1 kubinio metro kaušo talpos ekskavatoriais, suverčiant į sankasą, kaina už 1000 kubinių metrų	3 256,00 Lt.
IV grupės grunto kasimas 1 kubinio metro kaušo talpos ekskavatoriais, suverčiant į sankasą, kaina už 1000 kubinių metrų	2 531,00 Lt.
III grupės grunto kasimas 1 kubinio metro kaušo talpos ekskavatoriais, suverčiant į sankasą, kaina už 1000 kubinių metrų	1 926,00 Lt.
II grupės grunto kasimas 1 kubinio metro kaušo talpos ekskavatoriais, suverčiant į sankasą, kaina už 1000 kubinių metrų	1 634,00 Lt.
I grupės grunto kasimas 1 kubinio metro kaušo talpos ekskavatoriais, suverčiant į sankasą, kaina už 1000 kubinių metrų	1 230,00 Lt.
V grupės grunto kasimas 1,25 kubinio metro kaušo talpos ekskavatoriais, suverčiant į sankasą, kaina už 1000 kubinių metrų	2 692,00 Lt.
IV grunto grupės kasimas 1,25 kubinio metro kaušo talpos ekskavatoriais, suverčiant į sankasą, kaina už 1000 kubinių metrų	2 059,00 Lt.
III grupės grunto kasimas 1,25 kubinio metro kaušo talpos ekskavatoriais, suverčiant į sankasą, kaina už 1000 kubinių metrų	1 578,00 Lt.
II grupės grunto kasimas 1,25 kubinio metro kaušo talpos ekskavatoriais, suverčiant į sankasą, kaina už 1000 kubinių metrų	1 297,00 Lt.
I grupės grunto kasimas 1,25 kubinio metro kaušo talpos ekskavatoriais, suverčiant į sankasą, kaina už 1000 kubinių metrų	1 028,00 Lt.

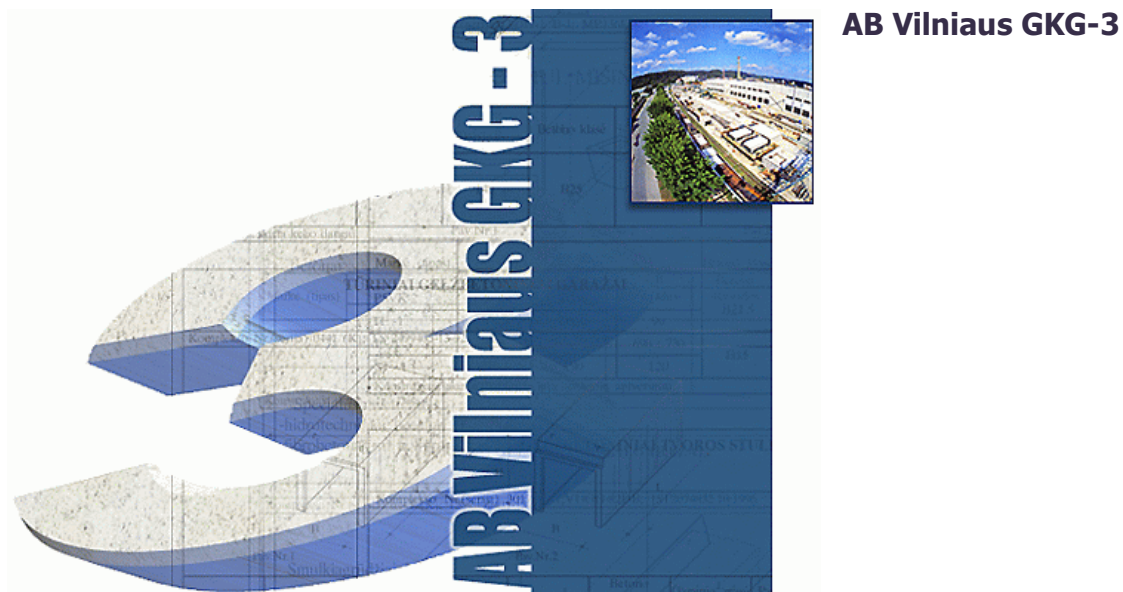
5.2 pav. Sąmatos informacija

Paprastas vartotojas gali tik matyti įvairių paslaugų kainas, tačiau negali susidaryti savo sąmatos.

„Forumas“ – čia pateikiamas forumas, kuriame galima diskutuoti iškilusiomis problemomis bei klausimais.

„Firmos“ – informacija, kurią pateikia pačios firmos, jų reklaminiai tinklapiai bei kontaktai.

Pavyzdžiui:



AB Vilniaus GKG-3

Ieškome verslo partnerių

AB "Vilniaus gelžbetoninių konstrukcijų gamykla Nr.3"

AB "Vilniaus GKG-3" ieško partnerių, galinčių tiekti metalą, cementą, kalkes, inertines medžiagas.

Komercinius pasiūlymus siūsti:

Adresas Šaltupio g. 3 (Žemieji Paneriai), 2053
Vilnius

El. paštas info@gkg3.lt

Tel. pasiteiravimui: (8-5) 264 76 66
(8-5) 264 76 64

Faks. (8-5) 264 13 26

Pav 5.3 Įmonės informacija

„Kalendorius“ – kalendorius. Galima pasirinkti norimų metų bei mėnesio kalendorių (5.4 pav.).

Darbu kalendorius Spausdinti

Renginiu kalendorius

Pirmadienis	Antradienis	Trečiadienis	Ketvirtadienis	Penktadienis	Šestadienis	Sekmadienis
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

5.4 Pav. Kalendorius.

„Registracija“ – paprastas vartotojas, gali prisiregistruoti ir prisijungęs naudotis visomis galimybėmis. T.y. turėti savo asmeninę sąmatą, kurią vėliau pats vartotojas galės redaguoti ir keisti (5.5 pav.).

Registracija Spausdinti

Prisijungimo vardas:

Slaptažodis:

Pakartoti slaptažodį:

Vardas:

Pavardė:

5.5 Pav. Registracija

„Straipsniai“ – pateikiami aktualūs straipsniai būsto tema (5.6 pav).

Straipsniai

Spausdinti

Žemė kaime pabrango iki neregėtų aukštumų

Per keletą mėnesių žemės ūkio paskirties sklypai brango 30-150 proc. Tokio šuolio neprognozavo niekas, o verslininkai teigia, kad dideliuose sklypuose žemė gali kainuoti bemaž dvigubai brangiau nei mažesniuose. [plačiau](#)

Nekilnojamojo turto rinkoje: „fundamentalistai“ ir „grafikistai“

Klube „Pinigų srautas“ surengtame seminare kvietinis svečias Lietuvos banko Ekonomikos departamento direktorius Raimondas Kuodis pristatė savo poziciją nekilnojamojo turto burbulo Lietuvoje tema, sakoma pranešime spaudai. [plačiau](#)

Nekilnojamojo turto kainų „burbulas“ Lietuvoje. Ar tikrai?

Nekilnojamojo turto rinkoje įtampa neslūgsta. Daugelis nekilnojamojo turto agentūrų vienu balsu tvirtina, kad nekilnojamojo turto kainos augs, nors jau ne tokiu greičiu kaip iki šiol ir kad reikia laukti kainų „burbulo“ sprogdimo... [plačiau](#)

Ar pabrangs studentui buto nuoma?

Nuo pirmosios šių metų dienos įsigaliojęs nekilnojamojo turto mokesčio įstatymas pastaruoju metu kelia daug prieštarų. Didžiausiu diskusijų objektu tapo tai, kad pirmą kartą nekilnojamojo turto mokesčio mokėtojais tapo fiziniai asmenys [plačiau](#)

© 2006. Būsto Portalas. Visos teisės saugomos.

[į viršų](#)

5.6 pav. Straipsnių sąrašas

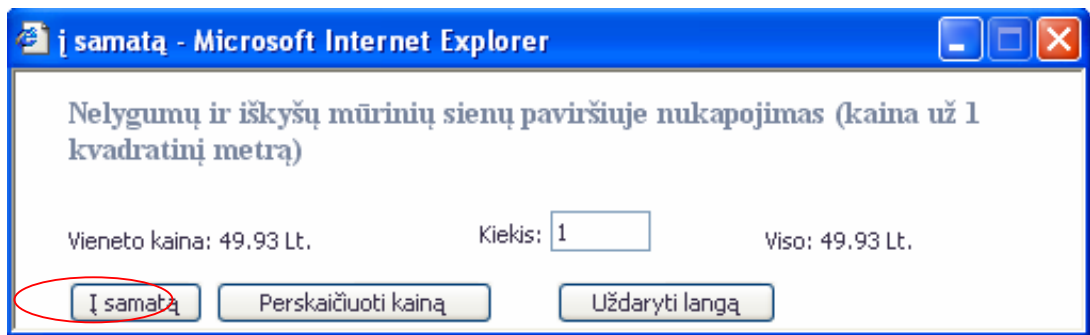
Norėdami paskaityti visą sąrašą, tiesiog reikia paspausti ant nuorodos „plačiau“ ir atsidarys visas straipsnis.

Norėdami atsispausdinti matomą vaizdą, reikia spausti ant mygtuko „Spausdinti“ ir bus atidarytas naujas langas su matomu tekstu ir paveikslėliais, kurių bus galima atsispausdinti.

5.2. Registruotas vartotojas

Registruotas vartotojas nuo paprasto vartotojo skiriasi tuo, kad gali prisijungti prie portalo ir susidėti savo sąmatą bei ją išsisaugoti.

Sąmatą susidėti yra paprasta. Tiesiog spaudžiate ant norimos paslaugos, tada atsidaro langelis, kuriame yra mygtukas „Į sąmatą“, tada jį paspaudžiate ir paslauga bus įdedama į sąmatą (5.7 pav.).



5.7 pav. Sąmatos formavimas

Taip pat galima toje pačioje formoje įvesti paslaugų kiekį, bei persiskaičiuoti sąmatą. Taip pat uždaryti langą, jeigu apsigalvojote ir nebenorite įkelti šios paslaugos į jūsų sąmatos sąrašą.

5.3. Registruotos įmonės dalis

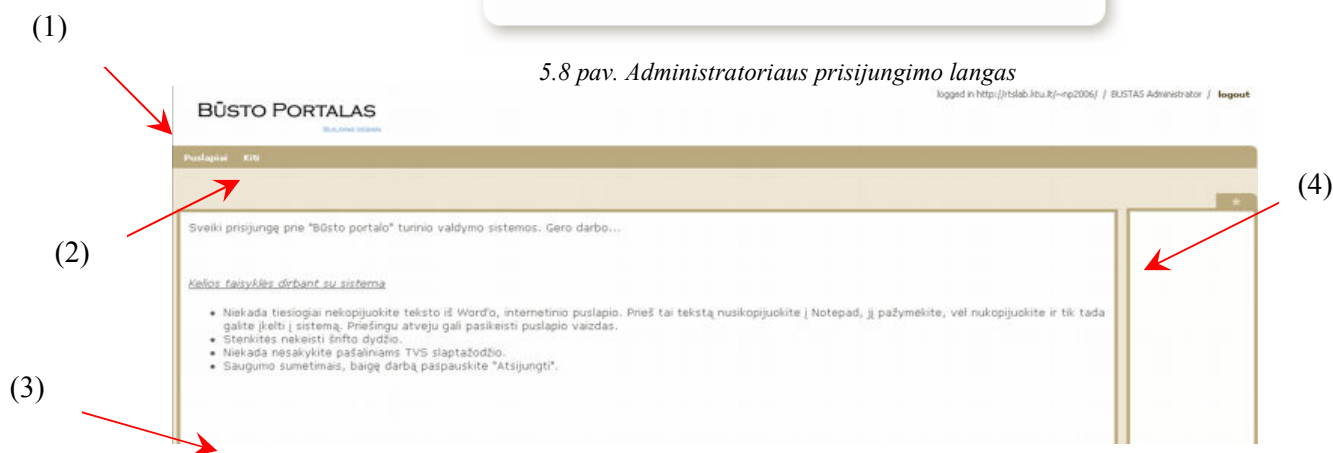
Įmonės dalis skiriasi nuo registruoto vartotojo tuo, kad ji gali susikurti savo reklaminių tinklapi, palikti ten savo kontaktus. Kaip tai padaryti yra apžvelgiama HTML redaktoriuje. Kitos funkcijos lieka tos pačios kaip ir registruoto vartotojo.

5.4. Portalo administratorius

Norint prisijungti prie administravimo skilties, reikia žinoti administracijos adresą, vartotojo vardą ir slaptažodį. Administracijos adresas dažniausiai yra *http://www.jususvetaine.lt/admin/*, tačiau gali ir skirtis. Šį adresą nustato ir suteikia sistemos platintojas instaliacijos metu, vėliau jis yra pastovus. Vartotojo vardą ir pradinį slaptažodį taip pat suteikia platintojas, slaptažodį vėliau galima keisti. Atsidarę administratoriaus prisijungimo formą, įvedate savo prisijungimo vardą ir slaptažodį ir spaudžiate *enter*.



5.8 pav. Administratoriaus prisijungimo langas



5.9 pav. Administratoriaus pradinis langas

Prisijungus prie sistemos matomas langas, suskirstytas į keturias dalis (5.9 pav.):

Administratoriaus dalies struktūra

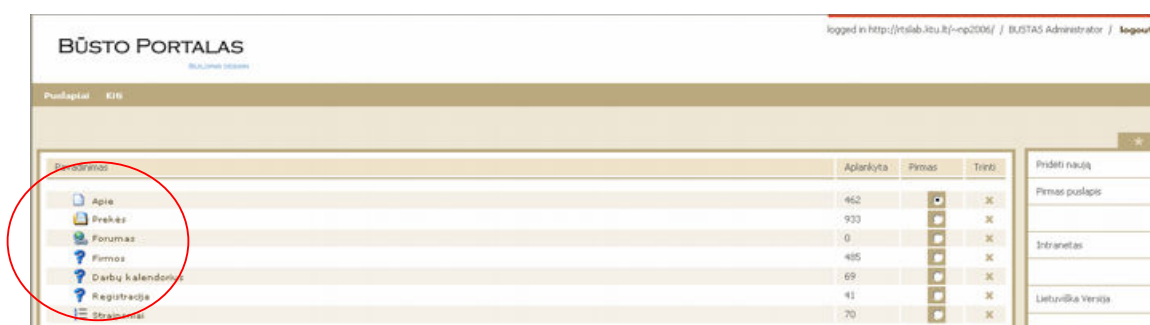
- (1) Antraštė
- (2) Administratoriaus meniu
- (3) Pateikiama informacija
- (4) Papildomas meniu, atsirandantis, paspaudus ant pagrindinio meniu

5.5. Sistemos administravimo galimybės

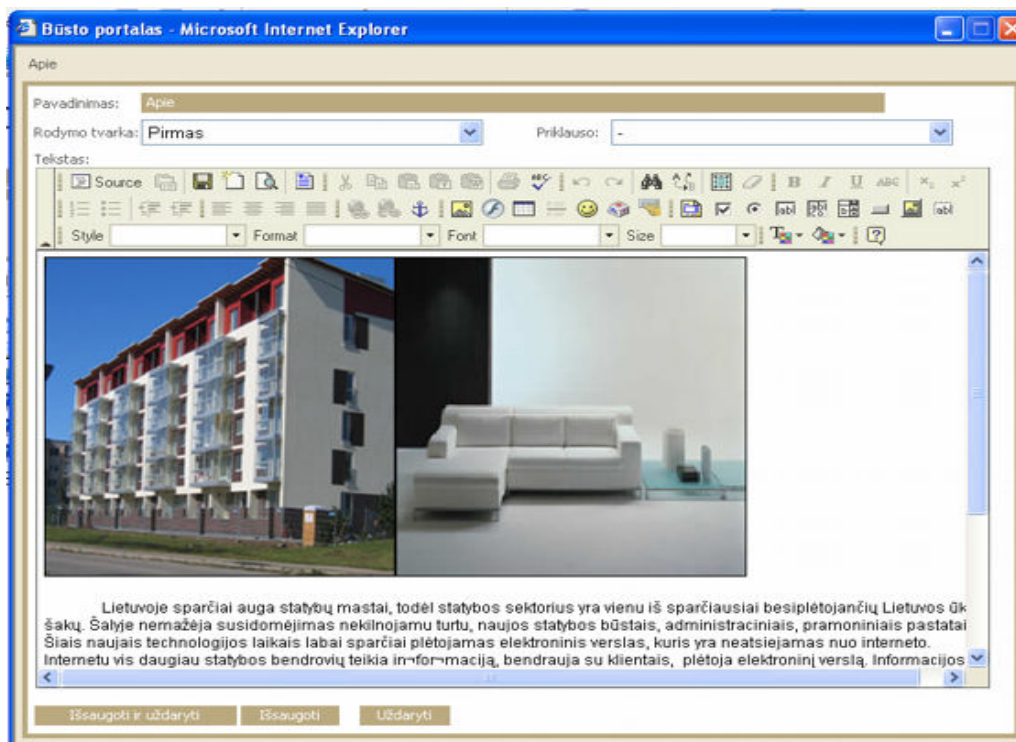
Administratorius, prisijungęs prie turinio valdymo sistemos gali, besinaudodamas savo meniu atlikti šiuos veiksmus:

- Paspaudus ant meniu punkto „Puslapiai” atsidarome esančių pagrindiniame puslapyje meniu sąrašą.

1. Pirmasis punktas „Apie”. Šioje srityje galime sukurti pradinį tinklapio puslapį. Paspaudus ant šio punkto atsidaro langas (5.10 pav.):



5.10 pav. Pirmasis punktas „Apie“

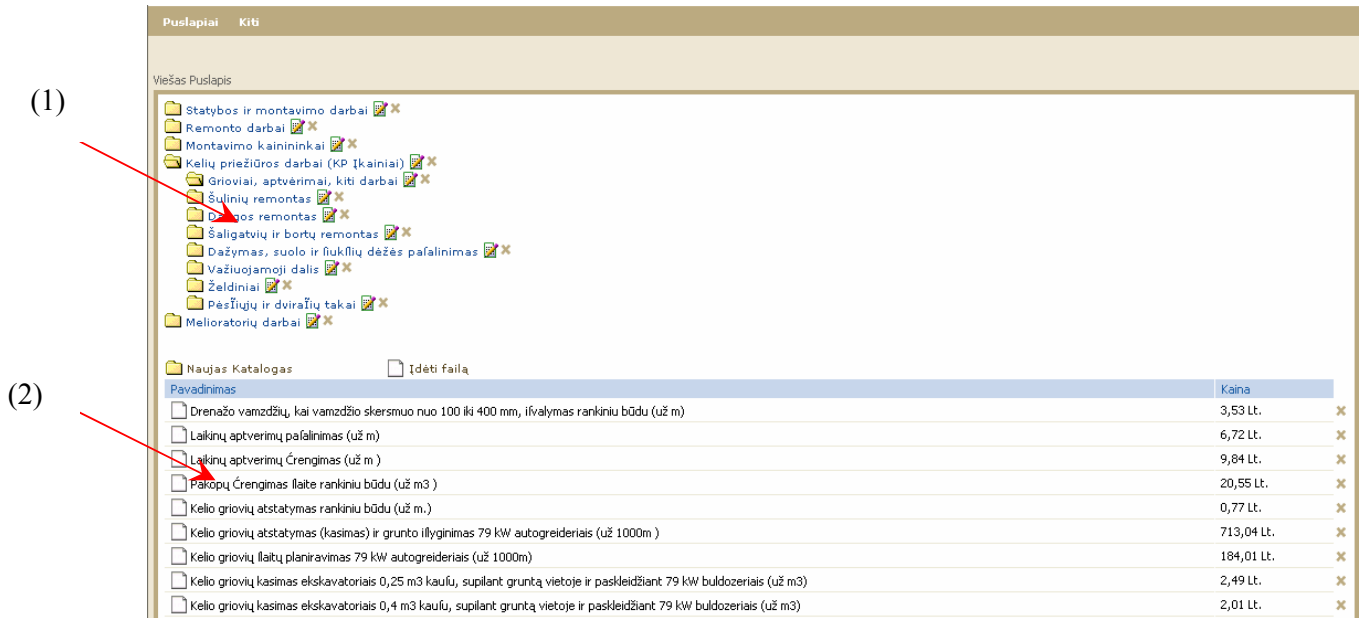


5.11 pav. Pradinio puslapio redagavimas

Šiame lange yra visos reikalingos funkcijos, norint sukurti pradinį puslapį, tam naudojamas įprastinis HTML redaktorius.

2. Antrasis punktas „Prekės”. Čia pateikiamos visos sąmatą sudarančios prekės bei paslaugos. Administratorius gali įkelti/ištrinti egzistuojančias paslaugas ir prekes. Tačiau jeigu norima ištrinti, tai reikia trinti po vieną elementą. Negalima ištrinti viso prekių medžio iškart.

Šitaip atrodo langas su paslaugomis ir katalogais, kuriuose jos yra (5.12 Pav.):



5.12 pav. Paslaugų sąrašas.

(1) – Katalogų medis.

(2) – Prekės, esančios kataloguose.

Norėdami įterpti naują paslaugą, ar redaguoti esamą, mes ją atsidarome paspausdami ant ženkliuko . Jeigu norime redaguoti, spaudžiame šį ženkliuką, esantį prie jau įdėtos paslaugos. Jeigu norime įterpti naują spaudžiame šį ženkliuką, esanti prie „Įdėti failą”. Atsidaro langas (5.13 pav.)

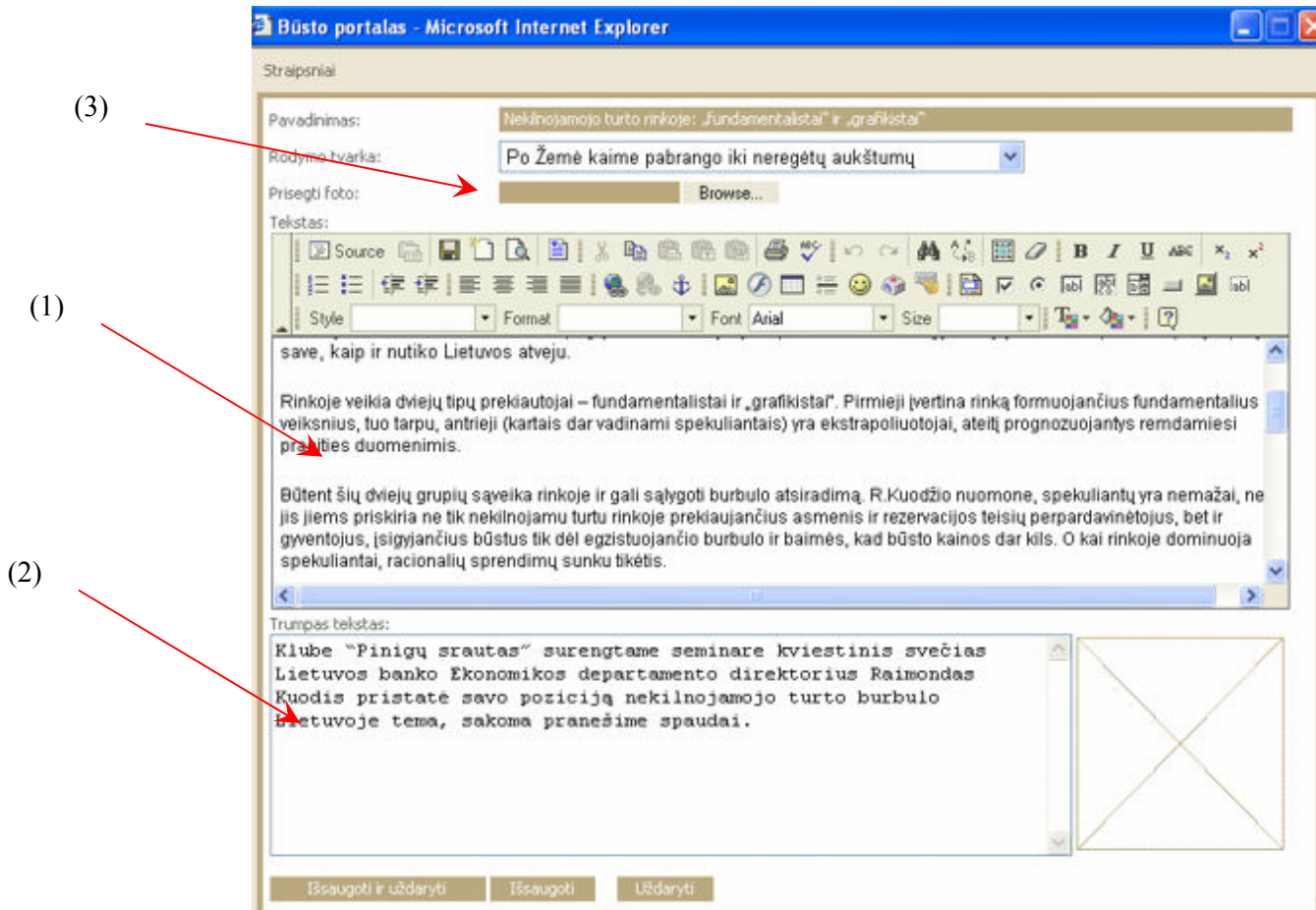
The screenshot shows a web form titled 'Būsto portalas - Microsoft Internet Explorer'. The form has the following fields and buttons:

- Pavadinimas:
- Komentaras:
- Data:
- Kaina: Lt.
- Išsaugoti ir uždaryti
- Išsaugoti
- Uždaryti

5.13 Pav. Naujos paslaugos įterpimo forma

Užpildę šią formą ir paspaudę mygtuką „Išsaugoti ir uždaryti”, mes gauname naują įterptą paslaugą.

3. Punte „Forumas” yra nustatomas forumo adresas ir pavadinimas.
4. „Straipsniai” – čia pateikiama informacija su straipsniais, susijusiais su būstu.
Atsidarę šį punktą mes matome sąrašą visų straipsnių. Pasirinkę kuri nors vieną, mes galime jį redaguoti. Redagavimo forma labia panaši į tinklapio redagavimo formą, tačiau yra keletas papildomu sričių (5.14 pav.):



5.14 pav. Straipsnių redagavimo forma

- (1) – Pagrindinis straipsnio turinys. (Rodomas išskleidus straipsnį)
- (2) – Sutrumpintas straipsnis. (Rodomas straipsnių sąrašė).
- (3) – Prisegama fotografija.

- Paspaudus ant meniu punkto „Kiti” atidarome modulių esančių puslapyje sąrašą.
 - Kalendorius – svetainėje matomas kalendorius.
 - Apklausa – galima daryti portale apklausą.
 - Kalbos – kalbos, kuriomis galima vaizduoti portalą.
 - Reklama – reklama.
 - Dokumentai
 - Vartotojai – vartotojų sąrašas, su jiems suteiktomis privilegijomis(5.15 pav.):



Vartotojo vardas		Slaptažodis	Svetainė	Trinti
GGG-3	AB "Vilniaus GGG-3"	keičti	naikinti	X
Skala	UAB "Skalos statyba"	keičti	naikinti	X
Agayta	UAB "Agayta"	keičti	naikinti	X

(1) (2) (3) (4) (5)

5.15 pav. Langas „Kiti“

- (1) – Vartotojo ar firmos prisijungimo vardas.
- (2) – Vartotojo ar firmos pavadinimas.
- (3) – Slaptažodžio keitimas.
- (4) – Galima įjungti/išjungti vartotojui jo svetainės turėjimą. Pavyzdžiui, jeigu firma pageidauja turėti savo reklaminį tinklą mūsų portale, jai tai leidžiama ir ji gali naudodamasi HTML redaktoriumi susikurti savo reklaminį tinklą.
- (5) – Administratorius turi galimybę panaikinti vartotoją.
 - Galerija

Baigus darbą su administratoriaus sistema rekomenduojama nuo jos atsijungti (nuoroda pagrindinio lango dešinėje), taip pat uždaryti naršyklės langą.

5.6. HTML redaktorius

Interneto svetainės į naršyklę patenka parašytos joms suprantama kalba - HTML. Tam, kad paprastam vartotojui nereikėtų jos mokytis, turinio valdymo sistema gali būti įdiegta su WYSIWYG (What You See Is What You Get - Ką matai, tą ir gauni) HTML redaktoriumi

Redaktoriaus langas ir mygtukai primena paplitusias teksto redagavimo programas, todėl susipažinus su bent viena iš jų neturėtų būti sunku perprasti šio redaktoriaus veikimą ir valdyti savo svetainės tekstus.

Pastaba! Redaktorius dirba paragrafų režimu - t.y. paspaudus [Enter] pradedamas naujas paragrafas (dvigubas tarpas prieš naują eilutę). Norint tiesiog perkelti kursorių į kitą eilutę reikėtų spausti [Shift]+[Enter]. (Pilnas HTML redaktoriaus aprašymas pateikiamas 9.2 priede).

6. Išvados

1. Išanalizavus portalų kūrimo technologijas nustatyta, kad anksčiau sukurti portalai neturi metaduomenų lygio, kas yra atlikta portale „Būsto portalas“.

2. Atlikus nagrinėjamų portalų lyginamąją analizę buvo apibrėžtos kuriamo portalo išskirtinės savybės, kurios parodė, kad šio portalo galimybės yra labai didelės:

- ✚ Tobulesnė turinio valdymo sistema leidžia greitai ir patogiai kelti duomenis, juos koreguoti, papildyti ir panaikinti įrašus.

3. Išanalizavus sukurtų portalų esamą padėtį Lietuvoje, buvo apibrėžta kuriamo portalo nauda.

- ✚ Supaprastintas portalo valdymas, nes portalo struktūra yra apibrėžta metaduomenų lygyje.

- ✚ Greita informacija.

- ✚ Taupomos eksploatacinės išlaidos.

4. Remiantis panašių sistemų analize ir vartotojų problemomis, apibrėžti pagrindiniai vartotojo reikalavimai. Portalo funkcionalumą apibrėžia panaudojimo atvejų ir veiklos diagramos.

5. Suprojektuota:

- ✚ Loginė architektūra.

- ✚ Sistemos elgsenos modelis, pateiktas sekų diagramomis.

- ✚ Reliacinės duomenų bazės schema.

- ✚ Sistemos testavimo modelis.

- ✚ Metabazės lygis, leidžiantis lengviau projektuoti portalo struktūrą, pritaikant ją kitam tikslui bei tvarkant esamą portalą.

6. Realizuotas metabazės lygis, kuriam esant daug lengviau perprojektuoti portalo struktūrą, pritaikant kitam tikslui bei tvarkant esamą portalo struktūrą.

7. Metaduomenų lygio panaudojimas suteikė tokią naudą.

7.1 Realizuota metaduomenų bazė, pasinaudojant MySQL duomenų bazių valdymo sistema, kurios pagalba atsirado galimybė lengviau specifiuoti portalą:

- ✚ Įterpti naują meniu punktą tiesiai per metabazę.

- ✚ Galimybė vartotojui lengviau dalyvauti portale.

7.2 Realizuotas metaduomenų lygis leidžia patogesnę naudojimąsi portalui administruoti.

- ✚ Lengviau pakeičiama duomenų struktūra.

- ✚ Lengvesnis vartotojų administravimas.

- ✚ Patogesnis informacijos įkėlimas.

- ✚ Metabazės lygis leidžia greičiau pasiekti duomenis.

- ✚ Metabazės lygis leidžia administratoriui lengviau ir greičiau perprojektuoti visą portalą kitam tikslui. (pasikeitus portalo informacijos sričiai).
- ✚ Naudojantis metabazės lygiu tapo lengviau palaikyti ir plėtoti portalą.
- ✚ Lengvesnis struktūros praplėtimas (pvz. Naujo meniu įkėlimas).
- ✚ Lengviau naujam administratoriui prižiūrėti portalą, nes atskirta veikimo logika nuo duomenų.

8. Atlikus palyginimą su panašios paskirties sistemomis pagal iškeltus tokių sistemų kriterijus galima daryti išvadą, kad realizuota sistema pranašesnė tokių savybių atžvilgiu:

- ✚ greitesnis veikimas,
- ✚ didesnis stabilumas,
- ✚ geresnis sistemos adaptyvumas,
- ✚ patogesnis naudojimas:
 - lengviau pakeičiama duomenų struktūra,
 - lengvesnis vartotojų administravimas,
 - patogesnis informacijos įkėlimas.

7. Literatūra

- [1] American Library Association, Task Force on Metadata Summary Report, June 1999. [žiūrėta 2007-09-24]. Prieiga per internetą: <http://www.libraries.psu.edu/tas/jca/ccda/tf-meta3.html>
- [2] D.C.A. Bultermann, Is It Time For a Moratorium on Metadata?, IEEE MultiMedia, Oct-Dec 2004
- [3] Lexbe. New Approaches Required for e-Discovery and Metadata. [žiūrėta 2007-09-24]. Prieiga per internetą: <http://www.lexbe.com/hp/indepth-e-discovery-rule-metadata.htm>
- [4] Bretherton, F.P. and Singley, P.T. 1994, Metadata: A User's View, Proceedings of the International Conference on Very Large Data Bases (VLDB), 1091-1094
- [5] Ralph Kimball, The Data Warehouse Lifecycle Toolkit, Wiley, 1998, ISBN 0-471-25547-5
- [6] Making a List of Data About Metadata and Exploring Information Cataloging Tools. 1998 [žiūrėta 2007-09-24]. Prieiga per internetą: <http://www.fortunecity.com/skyscraper/oracle/699/orahtml/dbmsmag/9803d05.html>
- [7] Guy V Tozer, Metadata Management for Information Control and Business Success, Artech House, 1999, ISBN 0-89006-280-3
- [8] Adrienne Tannenbaum, Metadata Solutions: Using Metamodels, Repositories, XML, and Enterprise Portals to Generate Information on Demand, Addison-Wesley, 2002, ISBN 0-201-71976-2
- [9] David Marco, Building and Managing the Meta Data Repository: A Full Lifecycle Guide, Wiley, 2000, ISBN 0-471-35523-2
- [10] David C. Hay, Data Model Patterns: A Metadata Map, Morgan Kaufman, 2006, ISBN 0-12-088798-3
- [11] R. Todd Stephens (2003). Utilizing Metadata as a Knowledge Communication Tool. Proceedings of the International Professional Communication Conference 2004. Minneapolis, MIN: Institute of Electrical and Electronics Engineers, Inc.
- [12] Metadata in the Oxford Digital Library. [žiūrėta 2007-09-24]. Prieiga per internetą: <http://www.odl.ox.ac.uk/metadata.htm>
- [13] Innovation and research. [žiūrėta 2007-09-24]. Prieiga per internetą: <http://www.cs.cornell.edu/wya/DigLib/MS1999/Chapter4.html>
- [14] Federal Geographic Data Committee (FGDC) Metadata. [žiūrėta 2007-09-24]. Prieiga per internetą: <http://libraries.mit.edu/guides/subjects/metadata/standards/fgdc.html>
- [15] Federal Geographic Data Committee. Geospatial Metadata. [žiūrėta 2007-09-24]. Prieiga per internetą: <http://www.fgdc.gov/metadata>

- [16] Metamodelių „zoologijos sodai“. [žiūrėta 2007 m. rugsėjis 24 d.]. Prieiga per internetą: <http://www.eclipse.org/gmt/am3/zoos/>
- [17] Knowledge Systems, AI laboratory. Stanford University. What is an Ontology? [žiūrėta 2007-09-24]. Prieiga per internetą: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>
- [18] The Object Management Group (OMG). Prieiga per internetą: <http://www.omg.com>,
- [19] Schmidt, D.C. (February 2006). "Model-Driven Engineering". *IEEE Computer* **39** (2). [žiūrėta 2007-09-24]. , "A promising approach to address platform complexity — and the inability of third-generation languages to alleviate this complexity and express domain concepts effectively — is to develop Model-Driven Engineering (MDE) technologies..."
- [20] Mantas Krugiškis. Informacinės sistemos, sukurtos META modelio pagrindu, ypatumai. Informacinės technologijos 2007: Teorija, Praktika, Inovacijos. 2007. Alytus. ISBN 978-9955-9779-4-0

8. Metadata based CRM system

8.1. Summary

The main goal of this project is to create an informative system based on metadata level due to the research done on existing sites in Lithuania, which has shown that those sites don't have some functional possibilities: they are difficult to manage, loading data is complicated, redesigning is expensive and hard, slow operation, high financial maintenance.

Logical site architecture has been designed as well as the model of system behaviour (using diagrams of sequences), scheme of relative data base, model of system testing and meta-base level. System model has been tested, maintenance and user's instruction documents have been prepared, any logical and other mistakes have been corrected.

This site, The dwelling site, lets easily change the structure of data and widen it, it has become more convenient to change user's possibilities and load new data and use the site itself. The data is quicker to reach because of the metadata level. It is also easier for site administrator to redesign the whole site for the different purpose when the range of information changes. When managing of content is easy to maintain it is also easier for a less experienced administrator to look after the site, because the logic of site operation is separated from data being used.

9. Priedai

9.1. Duomenų bazės lentelės

9.1 lentelė. „Basket“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumasis įvesti</i>	<i>tipas</i>	<i>Null</i>
basketID	Įrašo į lentelę bei krepšelio identifikatorius.	Ne	int(11) unsigned	Ne
owner	Krepšelio avininko identifikatorius.	Ne	varchar(200)	Taip
title	Krepšelio pavadinimas.	Taip	varchar(200)	Taip
amount	Paslaugos kiekis.	Taip	float	Taip
cost	Paslaugos kaina.	Taip	float	Taip
docID	Įrašytos į lentelę prekės dokumento identifikatorius.	Ne	int(11) unsigned	Taip

9.2 lentelė. „CuratioLogins“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumasis įvesti</i>	<i>tipas</i>	<i>Null</i>
loginID	Įrašo į lentelę bei prisijungimo identifikatorius.	Ne	int(11) unsigned	Ne
userID	Su šia lentele susieto vartotojo identifikatorius.	Ne	int(11) unsigned	Taip
userName	Vartotojo prisijungimo vardas.	Ne	varchar(50)	Taip
firstName	Vartotojo vardas.	Ne	varchar(50)	
lastName	Vartotojo pavardė.	Ne	varchar(50)	Taip

ip	IP adresas prisijungusio vartotojo.	Ne	varchar(50)	
date	Prisijungimo data ir laikas.	Ne	datetime	
ip_	Proxy serverio IP adresas.	Ne	varchar(50)	

9.3 lentelė. „CuratioMenu“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
menuID	Įrašo į lentelę bei Menu identifikatorius.	Ne	int(50) unsigned	Ne
title	Menu punkto pavadinimas.	Taip	char(200)	Taip
parent	Identifikatorius, kuris nurodo tėvo katalogo identifikatoriaus numerį.	Taip	char(250)	Taip
listOrder	Puslapio buvimo vieta.	Taip	int(3) unsigned	Taip

9.4 lentelė. „CuratioSessions“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
sessionID	Įrašo į lentelę bei sesijos identifikatorius.	Ne	int(11) unsigned	Ne

SID	Sesijos įrašas. Vartotojo prisijungimo metu sukuriant sesiją sugeneruojamas atsitiktinė skaičių seka, kuri saugoma šiame laukelyje. Ji naudojama pastoviai identifikuojant vartotoją darbo su sistema metu.	Ne	varchar(50)	Taip
date	Sukurtos sesijos data ir laikas.	NE	timestamp(14)	Ne
userID	Su šia lentele susieto vartotojo identifikatorius.	Taip	int(11) unsigned	Taip

9.5 lentelė. „curatioUserss“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
userID	Įrašo į lentelę bei vartotojo identifikatorius.	Ne	int(11) unsigned	Ne
usersName	Vartotojo prisijungimo vardas.	Taip	varchar(50)	Taip
firstName	Vartotojo vardas.		varchar(50)	
lastName	Vartotojo paradė.	Taip	varchar(50)	Taip
pass	Vartotojo prisijungimo koduotas slaptažodis.	Taip	varchar(50)	Taip

9.6 lentelė. „Data“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
-----------------------------	---------------------------	-------------------------	--------------	-------------

dataID	Įrašo į lentelę bei duomenų identifikatorius.	Ne	int(11) unsigned	Ne
title	Įrašo į lentelę pavadinimas.	Taip	varchar(255)	taip
content	Įrašo į lentelę turinys.	Taip	text	taip

9.7 lentelė. „DocTree“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
TreeID	Įrašo į lentelę bei katalogo identifikatorius.	Ne	int(11) unsigned	Ne
title	Katalogo pavadinimas	Taip	varchar(255)	Taip
parent	Identifikatorius, kuris nurodo tėvo katalogo identifikatoriaus numerį.	Taip	int(11) unsigned	Taip
listOrder	Katalogo buvimo vietą.	Taip	int(11) unsigned	Taip
sh	Indeksas, kuris nurodo, kokie vartojai mato katalogą.	Ne	tinyint(3)	Taip

9.8 lentelė. „Documents“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
docID	Įrašo į lentelę bei prekės identifikatorius.	Ne	int(10) unsigned	Ne
price	Prekės kaina.	Taip	double	Taip

title	Prekės pavadinimas	Taip	varchar(255)	Taip
treeID	Nurodo paslaugos susietos su konkrečiu katalogu identifikatorių.	Ne	int(11) unsigned	Taip
comment	Prekės komentaras	Taip	text	Taip
sh	Numeris, kuris parodo, kokia vartotojo grupė mato prekę.	Ne	tinyint(2)	Taip
data	Prekės patalpinimo data ir laikas.	Ne	varchar(50)	Taip

9.9 lentelė. „Events“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumai įvesti</i>	<i>Tipas</i>	<i>Null</i>
eventID	Įrašo į lentelę bei kalendoriaus identifikatorius.	Ne	int(11) unsigned	Ne
date	Renginio ar kito įvykio data.	Taip	date	Taip
time	Renginio ar kito įvykio laikas.	Taip	time	Taip
type	Renginio ar kito įvykio antraštė.	Taip	varchar(50)	Taip
title	Renginio ar kito įvykio antraštės pavadinimas.	Taip	varchar(255)	Taip
place	Renginio ar kito įvykio vieta.	Taip	varchar(255)	Taip
program	Renginio ar kito įvykio aprašymas.	Taip	text	Taip

9.10 lentelė. „Intranetlogins“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumai įvesti</i>	<i>tipas</i>	<i>Null</i>
-----------------------------	---------------------------	-------------------------	--------------	-------------

loginID	Įrašo į lentelę bei registruoto vartotojo prisijungimo identifikatorių.	Ne	int(11) unsigned	Ne
userID	Su šiuo įrašu susieto vartotojo identifikatorius.	Ne	int(11) unsigned	Taip
userName	Registruoto vartotojo prisijungimo vardas.	Ne	varchar(50)	Taip
firstName	Vartotojo vardas, arba įmonės tipas (pvz.: UAB).	Ne	varchar(50)	Taip
lastName	Vartotojo pavardė arba įmonės pavadinimas.	Ne	varchar(50)	Taip
ip	IP adresas prisijungusio vartotojo.	Ne	varchar(15)	Taip
date	IP adresas prisijungusio vartotojo data ir laikas	Ne	datetime	Taip
ip_	IP proxy adresas prisijungusio vartotojo.	Ne	varchar(15)	Taip

9.11 lentelė. „IntranetUsers“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
userID	Įrašo į lentelę bei registruoto vartotojo identifikatorių.	Ne	int(11) unsigned	Ne
userName	Registruoto vartotojo prisijungimo vardas.	Taip	varchar(50)	Taip
firstName	Vartotojo vardas, arba įmonės tipas (pvz.: UAB).	Taip	varchar(50)	Taip
lastName	Vartotojo pavardė arba įmonės	Taip	varchar(50)	Taip

	pavadinimas.			
pass	Vartotojo prisijungimo koduotas slaptažodis.	Taip	varchar(50)	Taip

9.12 lentelė. „IntranetSessions“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
sessionID	Įrašo į lentelę bei sesijos identifikatorius.	Ne	int(11) unsigned	Ne
SID	Sesijos įrašas. Vartotojo prisijungimo metu sukuriant sesiją sugeneruojamas atsitiktinė skaičių seka, kuri saugoma šiame laukelyje. Ji naudojama pastoviai identifikuojant vartotoją darbo su sistema metu.	Ne	varchar(50)	Taip
date	Sukurtos sesijos data ir laikas.	NE	timestamp(14)	Ne
userID	Su šia lentele susieto vartotojo identifikatorius.	Taip	int(11) unsigned	Taip

9.13 lentelė. „List“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
listID	Įrašo identifikatorius.	Ne	int(11) unsigned	Ne
title	Saugomi straipsniai skyrelio antraštės.	Taip	text	Taip

content	Saugoma straipsnių pradžios turinys.	Taip	text	Taip
content_	Saugomas straipsnių pilnas turinys.	Ne	text	Taip
pageID	Reikšmė kuri norodo puslapio indeksą.	Ne	varchar(200)	Taip
listOrder	Nurodomas straipsnio eilės numerį.		int(11) unsigned	Taip
showInFirstPage	Indeksas, kuris nurodo ar straipnis yra matomas pirmame puslapyje. Jei indeksas lygus 1, tai jis matomas pirmajame, kitu atveju reikšmė lygu 0.	Taip	tinyint(2)	Taip
date	Patalpinto straipsnio data.	Ne	date	Taip
visited	Kiekvieno straipsnio apsilankymų skaičius.	Ne	int(11)	Taip

9.14 lentelė. „Organizations“ lentelės atributų aprašymas:

<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
orgID	Įrašo į lentelę bei registruoto vartotojo, kuriam suteikta organizacijos teisės, identifikatorius.	Ne	int(11) unsigned	Ne
Title	Organizacijos vardas, arba įmonės tipas (pvz.: UAB).	Ne	varchar(250)	Taip
content	Čia saugoma organizacijos puslapio HTML tekstas, kurį	Ne	longtext	Taip


	sugenarvo TVS redaktorius.			
userID	Su šiuo įrašu susieto registruoto vartotojo identifikatorius.	Ne	int(11) unsigned	Taip


9.15 lentelė. „Pages“ lentelės atributų aprašymas:


<i>Laukelio pavadinimas</i>	<i>Funkcijos, reikšmė</i>	<i>Būtinumas įvesti</i>	<i>tipas</i>	<i>Null</i>
pageID	Įrašo į lentelę bei portalo identifikatorių.	Ne	int(11) unsigned	Ne
parent	Identifikatorius, kuris nurodo tėvo meniu punkto identifikacijos numerį. Šis atvejis, naudojamas naudojant „krintatį“ meniu.	Taip	int(11) unsigned	Taip
listOrder	Saugomas meniu punkto išdėstymo tvarkos indekas.	Taip	float	Taip
content	Saugoma meniu punkto puslapio HTML tekstas, kurį sugenarvo TVS redaktorius.	Taip	text	Taip
intranet	Nurodoma, ar puslapį matys tik registruoti vartotojai. Jei ši reikšmė lygi „1“ - puslapis bus matomas tik registruotiems vartotojams, jei 0 – matys visi vartotojai.	Taip	tinyint(3)	Taip
visited	Apsilankymų skaičius.	Ne	bigint(20)	Taip


9.2. HTML redaktorius


HTML redaktoriaus meniu juostos mygtukų funkcijos:

 Source – paspaudus šį mygtuką, jūs pamatysite savo kuriamo puslapio HTML kodą.

 - paspaudus šį mygtuką, jūs išvysite dokumento informaciją.

 *Išsaugoti* - paspaudus šį mygtuką bus išsaugoti visi atlikti pakeitimai.

 - sukuriamas naujas puslapis.

 - parodomas langas toks, koks jis bus spausdinamas.

 - galima pasirinkti puslapio struktūrą iš esamų šablonų:



Paveikslėlis ir pavadinimas

Vienas pagrindinis paveikslėlis ir pavadinimas bei tekstas supantis paveikslėlį.



Keistasis šablonas

Šablonas, kuris turi du stulpelius, kiekvienas stulpelis turi savo pavadinimą ir tam tikrą tekstą.



Tekstas ir lentelė


Pavadinimas su tekstu ir lentele.

 *Iškirpti (cut)* - pažymėtas blokas perkeliamas į laikinąją atmintį įterpimui.


 *Kopijuoti (copy)* - pažymėtas blokas kopijuojamas į laikinąją atmintį įterpimui.


 *Įdėti (paste)* - įterpiamas tekstas iš laikinosios atminties.


 *Įdėti kaip gryną tekstą* - įterpiamas tekstas be formatavimo iš laikinos atminties.


 *Idėti iš Word* - įterpiamas tekstas iš laikinosios atminties ir išvalomas nuo Microsoft Word "šiukšlių". Šį mygtuką reikėtų naudoti visada, perkeltant tekstą iš Microsoft Word programos.


 *Spausdinti(print)* – spausdinamas lango turinys.


 *Tikrinti kalbą(spellchek)* - patikrinamas parašyto teksto taisyklingumas.


 *Atšaukti* - panaikinamas paskutinis atliktas veiksmas.

 *Atstatyti* - atkuriamas paskutinis panaikintas veiksmas.

 *Rasti* - iššokusiam langelyje, galima įrašyti norimą surasti tekstą - jis bus pažymėtas dokumento tekste.

 *Pakeisti* - iššokusiam langelyje, galima įrašyti norimą surasti tekstą ir pakeisti jį automatiškai kitu.

 *Pažymėti viską (select all)* – pažymimas visas lango turinys.

 pažymėtame tekste išvalomos visos apdailos/formatavimo žymės - paliekamas grynas tekstas.

B *pastorinimas(Bold)* – pažymėtas tekstas yra pastorinamas.


I *pasvirimas(Italic)* – pažymėtas tekstas yra pakreipiamas į šoną.


U *pabraukimas(Underline)* – pažymėtas tekstas yra pabraukiamas.


ABC *perbraukimas* – pažymėtas tekstas yra perbraukiamas.


x_a *(subscript)* – pažymėtas tekstas ar skaičiai yra pažeminami, lyginant su kitu tekstu.


x² *(superscript)* – pažymėtas tekstas ar skaičiai yra paaukštinami, lyginant su kitu tekstu.


 *Įterpti/Panaikinti numeruotą sąrašą* - kiekviena nauja teksto eilutė prasidės skaičiumi ar raide, einančiais iš eilės. Nustatyti sąrašo numeracijos tipą galima paspaudus dešinią pelės klavišą sąrašo viduje ir pasirinkus meniu punktą "Numeruoto sąrašo savybės".


 *Įterpti/Panaikinti suženklintą sąrašą* - kiekviena nauja teksto eilutė prasidės nustatyta žyme. Žymę pakeisti galima paspaudus dešinią klavišą sąrašo viduje ir pasirinkus meniu punktą "Suženklinto sąrašo savybės".


 *Sumažinti įtrauką* - tekstas pritraukiamas arčiau kraštų.


 *Padininti įtrauką* - tekstas atitraukiamas toliau nuo kraštų.

 *Lygiuoti kairę* - tekstas bus išlygintas pagal kairę pusę.


 *Centruoti* - tekstas bus rodomas puslapio centre.


 *Lygiuoti dešinę* - tekstas bus išlygintas pagal dešinę pusę.

 *Lygiuoti abi puses* - tekstas bus išlygintas pagal abudu šonus.

 *Įterpti/taisyti nuorodą* - pažymėjus tekstą, ant kurio norite sukurti nuorodą, arba nustačius kursorių ties jau sukurta nuoroda ir paspaudus šį mygtuką, iššokusiame langelyje galima įvesti nuorodos adresą bei keisti jos konfigūraciją. Paspaudus mygtuką "Naršyti po serverį" bus atidarytas dar vienas langas, kuriame galima pasirinkti norimą medžio elementą - jo adresas bus automatiškai įrašytas nuorodoje.


 *Panaikinti nuorodą*

 *Įterpti/taisyti žymę* - iššokusiame langelyje galima įrašyti žymės vardą. Žymė yra nematoma naršyklėje, tačiau į žymę galima sukurti nuorodą (jeigu žymės pavadinimas *zyme* tai nuorodos adresas turėtų būti *#zyme*, taip pat būtina pakeisti protokolą į "kita". Paspaudus tokią nuorodą, naršyklės langas automatiškai "persisuks" į vietą ties žyme.)

 *Įterpti/taisyti vaizdą* - paveiksluko/nuotraukos įkėlimas į HTML tekstą. Iššokusiame lange, galima įrašyti paveiksluko adresą arba jį pasirinkti iš jau įkeltų paveikslukų bibliotekos ("Naršyti po serverį"). Pasirinkus įkėlimą iš serverio, atsidaro papildomas langas, kuriame reikia pasirinkti norimą paveikslukų direktoriją, po to žemiau esančiame sąraše pasirinkti paveiksluką. Sumažinta jo versija bus rodoma šone - paspaudus pelyte sumažintą versiją, grįšite į ankstesnį langą. Paveikslukų bibliotekos lange taip pat galima įkelti ir naują paveiksluką iš disko į serverį - jis atsiras pasirinktos direktorijos sąraše. Pasirinkus/įrašius norimo paveiksluko adresą, lange bus matomas paveikslukas ir jo pozicija su tekstu. Paveiksluką galima sulygiuoti pagal teksto šoną (pasirinkimas "Lygiuoti"), taip pat nustatyti tarpus aplink jį (Hor. erdve, vert. erdvė), bei paveiksluko rėmelį (rekomenduojama įrašyti 0). Paspaudus OK - paveikslukas atsiras tekste. Paspaudus "Thumbnail" - tekste atsiras sumažinta paveiksluko versija, ir bus sukurta automatinė nuoroda, atidaranti padidintą

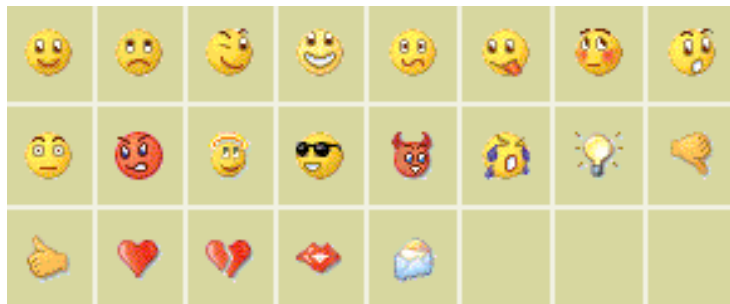
paveiksluko versiją. **NB!** Pastaroji ("Thumbnail") savybė tinkama tik serveryje esantiems per turinio valdymo sistemą įkeltiems paveikslukams!


 *Įterpti/taisyti Flash* - galimybė į tekstą įkelti Flash animaciją. **NB!** Ji nebus matoma HTML editoriuje taip kaip atrodytų naršyklėje!


 *Įterpti/taisyti lentelę* - atsidariusiame lange galima pasirinkti norimos įterpti lentelės parametrus (stulpelių/eilučių kiekį, tarpą tarp langelių ir pan.) Pažymėtina tai, kad standartiškai naršyklių kuriami rėmeliai yra neišvaizdūs, todėl rekomenduojama rėmelio plotį visada rašyti "0". Jeigu rėmeliai lentelėje reikalingi - sistemos platintojas gali sukurti keletą lentelės išvaizdos šablonų (stilių), kuriuos vėliau galima naudoti. Sukūrus lentelę, jos savybes galima keisti paspaudus dešinią pelės mygtuką bet kurioje celėje. Išsiskleidžiantis meniu leidžia įterpti/pašalinti stulpelius/eilutes/celes, juos sujungti ir skaidyti. Pasirinkus "Langelio savybės" ar "Lentelės savybės" bus atidaryti nauji langai, kuriuose galima keisti įvairius nustatymus (plotį, spalvas ir pan.)


 *Įterpti horizontalią liniją*

 - Įterpiama šypsienėlė iš rinkinio:



 *Įterpti specialų simbolį* - atsidariusiame lange, galima pelyte išsirinkti norimą įterpti simbolį.

 *Universali klaviatūra* - atsidariusiame lange, galima pelyte rinkti kelių kalbų ne lotyniškas raides.

 *Forma* – galima susikurti formą.

checkbox – pažymėti langeliai.

 Išsiskleidžiamas meniu

Mygtukas

Paskutinėje juostoje galime pasirinkti norimą šriftą, jo dydį bei pragrafų tipą:



Style – galima pasirinkti stilių:


- Custom Bold
- Custom Italic
- Title
- Code
- Title H3

Format – galima pasirinkti pragrafų tipą:

- Normal – paprastas tekstas
- Paragraph - paragrafas
- Formatted – suformuotas tekstas
- Address – adresas
- Heading 1 – pirmo lygio pavadinimas
- Heading 2 – antro lygio pavadinimas
- Heading 3 – trečio lygio pavadinimas
- Heading 4 – ketvirto lygio pavadinimas
- Heading 5 – penkto lygio pavadinimas
- Heading 6 – šešto lygio pavadinimas

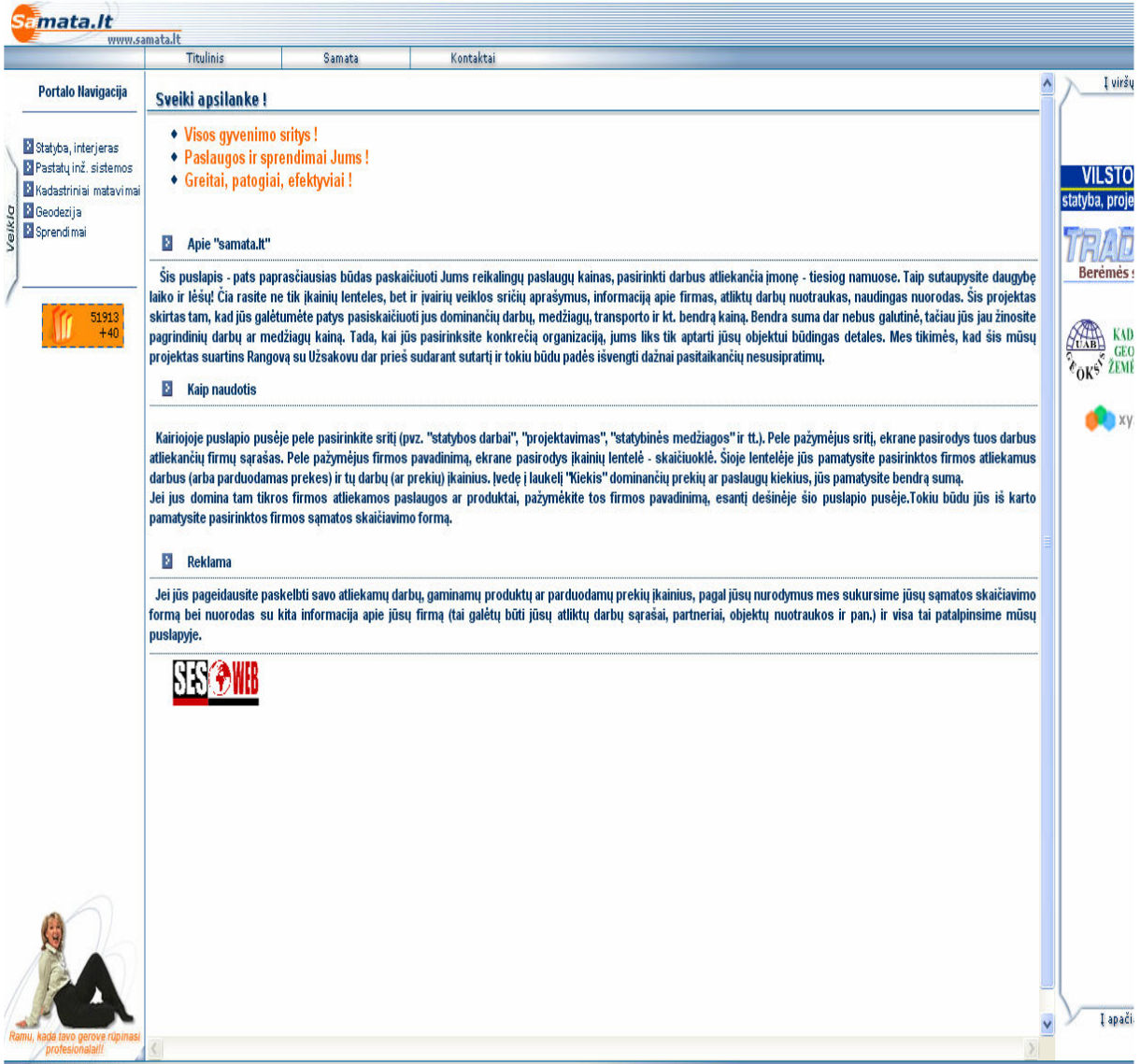
Font – galima pasirinkti norimą teksto šriftą.

Size – galima pasirinkti šrifto dydį.

 *Fono spalva* - analogiškas pasirinkimas po tekstu esančiai fono spalvai pakeisti. **NB!** Bus nudažytas tik plotas esantis po tekstu.

9.3. Tinklapių atvaizdai

9.3.1. Tinklapis www.samata.lt



The screenshot shows the homepage of the website www.samata.lt. The header features the logo and navigation links for 'Titulinis', 'Samata', and 'Kontaktai'. A left sidebar contains a 'Portalo Navigacija' menu with links to 'Statyba, interjeras', 'Pastatų inž. sistemos', 'Kadastriniai matavimai', 'Geodezija', and 'Sprendimai'. The main content area is titled 'Sveiki apsilanke!' and includes a list of services: 'Visos gyvenimo sritys!', 'Paslaugos ir sprendimai Jums!', and 'Greitai, patogiai, efektyviai!'. Below this is a section 'Apie "samata.lt"' with a paragraph describing the company's services and a 'Kaip naudotis' section. A 'Reklama' section follows, with a sub-header 'Reklama' and a paragraph about advertising services. The 'Reklama' section features a logo for 'SES WEB'. At the bottom left, there is a small image of a woman sitting on the floor with the text 'Ramų, kada tavo gerove rūpinasi profesionaliai!'. The right sidebar contains logos for 'VILSTO statyba, proje', 'TRAD Berėmės!', 'KAD GEO ŽEMĖ', and 'xy'. The page is framed by a blue border with 'Į viršų' and 'Į apačią' scroll indicators.

9.1 pav. Tinklapių www.samata.lt vaizdas

9.3.2. Tinklalapis www.bukra.lt

ru | en

Nuo projekto iki raktų

BukRA

Naujienos

INDIVIDUALŪS NAMAI

Apie mus | Technologija | Namas iki raktų | Mūsų darbai | Projektai | D.U.K

"Žmogus čia gali ateiti su savo dantų šepetuku ir gyventi, patirdamas visišką komfortą,"
"Lietuvos Rytas"
1994 m. spalio 21 d.

BukRA - namų statybos organizacija, geriausiai įgyvendinanti kiekvieno kliento norus.

Vardan ko organizacija egzistuoja?

Mes siekiame, kad Jūsų namai taptų Jūsų patogaus, saugaus ir ekologiško gyvenimo centru.

Kaip patikimas partneris mes:

- ☐ Konsultuojame išsirenkant namo idėją ir suprojektuojame pagal jūsų individualius poreikius.
 - ☐ Konsultuodami remiamės statybine ir gyvenimiška patirtimi, sveiku protu ir ateities tendencijomis.
- ☐ Organizuojame leidimų statybai gavimą.
 - ☐ Galėdami greičiau gauti leidimus, mes vertiname Jūsų brangų laiką, kurį galėtumėte skirti sau, šeimai ar draugams.
- ☐ Organizuojame ir kontroliuojame visą statybų procesą.
 - ☐ Atsakome už savo darbą tik patys galėdami jį kontroliuoti, todėl visas statybų procesas yra mūsų kontrolėje ir Jūsų žinioje.
- ☐ Garantuojame kokybišką darbų atlikimą.
 - ☐ Statydami laikomės kanadiečių technologijų, kurios tobulinamos daugiau kaip 60 metų. Tik kokybiškos medžiagos ir aukšta mūsų specialistų kvalifikacija įgalina suteikti Jums 10 metų garantiją.

14 metų rinkoje

9.2 pav. Tinklalapio www.bukra.lt vaizdas

9.3.3. Tinklapis www.asa.lt

The screenshot shows the website www.asa.lt, which is a portal for construction and architecture. The header includes the company name 'SLENKANČIOS SISTEMOS' and 'NAUJAS SALONAS VILNIUJE ŽIRMŪNŲ g. 107'. Navigation links include 'Informacijos atnaujinimas', 'Mūsų kainiai', 'Kaip naudotis sistema', 'Reklama Internetete', and 'Susisiekite su mumis'. A search bar is located in the top left, and a sidebar on the left contains a menu with categories like 'Titulinis', 'Katalogas', 'Statybos įstatymai', etc. The main content area features a 'Naujienos' section with several news items dated 2008-01-11, covering topics like the Vilnius city environment competition, state land use efficiency, and a house in Dubai. A 'Temos' (Topics) section lists various subjects such as 'Aplinkotvarka', 'Apsauga', and 'Ekologinė statyba'. A right sidebar contains a 'budma 2008' graphic and a 'Pranešimai visuomenei' section with news about sports commissions and climate change. A bottom banner reads 'TEGU LAIMI GERIAUSIASI! ASA.LT'.

9.3 pav. Tinklapis www.asa.lt vaizdas

9.4. Straipsnis

INFORMACINĖS SISTEMOS, SUKURTOS META MODELIO PAGRINDU, YPATUMAI

Mantas Krugiškis

*Kauno technologijos universitetas
Informacijos sistemų katedra, Studentų 50*

Straipsnyje analizuojamas meta modelių paremtų sistemų valdymas, taikant modelius. Apžvelgiama meta modelio kokybė bei kūrimo ypatumai. Apibūdinami terminų skirtumai tarp skirtingų modeliavimo technologijų. Apžvelgiami kūrimo ypatumai, kuriais reikia remtis norint sukurti geros kokybės meta modelį ir jį pritaikyti tolesniems projektams.

1 Įvadas

Prieš dešimt metų, buvo sunkiai suvokiama, kas yra meta modeliavimas. Žinančių apie tai žmonių buvo mažai.

Šiuo metu nauja meta modeliavimo technologija, rado erdvę pritaikymą. Deja, realiame pasaulyje terminas „meta“ niekada nebuvo visiškai priimtas ir apibrėžtas. Tačiau šiuo metu Duomenų inžinerijoje ir programinės įrangos inžinerijoje modelių naudojimas yra vis labiau rekomenduojamas. Tai turi būti priešpastatoma klasikinei, kodu paremtai, vystymo technologijai. Modelis visuomet yra sietinas su unikaliu meta modeliu. Viena iš šiuolaikinių aktyvių modelių paremtos inžinerijos šakų yra metodas, pavadintas modeliu paremtas kūrimas, pasiūlytas OMG. Šis metodas yra paremtas kalbos utilizavimu parašyti meta modelius vadinamus Meta Object Facility arba MOF. Tipiški meta modeliai pasiūlyti OMG yra UML, SPEM ar CWM. ISO taipogi turi išleidę standartinius meta modelius ISO/IEC 24744.

2 Kodėl meta modeliavimas susilaukia tiek daug dėmesio?

Mes galime tik galvoti, tačiau visada reikia remtis tam tikromis priežastimis.

- Atėjo laikas.

Meta modeliavimas prasidėjo nuo 1980 – ūjų. Bet interneto vystymasis, verslo integracija, duomenų integracija yra akivaizdžiai svarbiausia. Meta modeliai yra pagrindas duomenų integracijai, nors jie ne visada yra vadinami meta modeliais.

- Meta modelių paremtų programinių paketų pasirodymas.

Pirmasis meta modelių paremtas programinis paketas buvo CASE priemonė (kaip modeliavimo priemonės) su plečiamumu, arba visiškai konfigūruojamu meta modeliu. Vėliau atsirado ir Meta_Case priemonės. Tai leido vartotojo organizacijoms prisitaikyti programinę įrangą metodologijų tobulinimui, arba industriniams standartams.

Šiuo metu meta modelių paremta programinė įranga yra daugelyje vietų, kaip e-verslo integravimo komplektas.

- Meta modelių paremtų technologijų ir standartų galimybių didinimas.

Labiausiai žinomas pavyzdys šiuo atžvilgiu yra Unifikuota modeliavimo kalba (UML), kuri (daugiausia) apibrėžia meta modelį. UML nebuvo pirmoji, vis dėl to: pirmosios pastangos apima CDIF (CASE įrankio veikimas, paremtas integruotu meta modeliu), PCTE ir IRDS (abu repozitorių standartai), STEP (industrinis informacijos apsikeitimas) ir kiti.

- Poreikis kelti abstrakcijos lygį.

Meta modeliai yra naudingi žemesniuose abstrakcijos veikimo lygiuose, taip pat sprendžiant paskirstymo problemas, abstrakčių duomenų sub-problemas, fizinių duomenų optimizavimo ir kontroliuojamo srauto problemas. Šiuo atžvilgiu, meta modeliai yra idealūs padėjėjai kompleksiniais web servais paremtuose projektuose.

2.1 Meta modeliavimas – kas tai?

Meta modeliavimas pirmiausia yra veikla, kurios rezultatas-metamodelis. Meta modeliavimas yra artimas reliatyviam modeliavimui, t.y. objektiškai orientuotam modeliavimui, arba netgi klimato modeliavimui. Meta modeliai yra sukurti pasaulio apibūdinimui.

Meta modeliai kuriami dviem tikslais:

1. „Bandyti apibūdinti pasaulį“.
2. „Tam tikram tikslui“

Kodėl šie tikslai yra svarbūs? Todėl, kad, nėra modelio, kuris visiškai apibūdina klimatą. Klimato pokyčių modeliai yra aproksimacijos, tokios kaip meta modeliai.

Taip pat yra svarbu suprasti, kad meta modeliai visada duoda tam tikrą rezultatą. Niekada negalima naudoti metamodelio kitu tikslu, nei jis buvo sukurtas autoriaus. Vėl klimato modelio analogas: yra daug skirtingų klimato nustatymo modelių - pavyzdžiui, nuspėti ozono dujų kiekį po 100 metų antradienio popietei prie jūsų namo. Meta modeliai, turintys tą patį subjektą yra visiškai skirtingi, jeigu jie sukurti visiškai kitam tikslui

2.2 Kokybiniai meta modelių ypatumai

- Kaip "schema" semantiniams duomenims, kurie turi būti pakeisti. Tai yra kur meta modeliai turi pagrindą: duomenų apsikeitimui CASE įrankiuose (ir UML/XMI vėliau), STEP, pakeitimui darbinių duomenų ir netgi XML schemas.
- Artimai susiję yra meta modeliai, kaip schemas skirtos semantiniams duomenims, kurie turi būti saugomi, tarkim repozitoriuose.
- Kaip kalba kuri palaiko tam tikrą metodologiją ar procesą, kuris turi tikslą UML meta modelyje. Šiuo atveju, meta modelis leidžia kalbą kūrėjams ar metodologams geriau analizuoti ir suprasti, ką jie iš tikrųjų rašo apie tai metodologijos knygoje.
- Kaip kalba, išreikšti papildomą semantiką, egzistuojančios informacijos. Norint internetinio puslapio informaciją padaryti lengviau naudojamą.

2.3 Gero meta modelio ypatumai

Pasirinkti kriterijai:

1 lentelė. Meta modelio ypatumai

Kriterijus	Apibūdinimas
Informacijos netektis	Testavimas, pasiimti labiausiai mėgstamą duomenų rinkinį ir pažiūrėti kaip teisingai jį galima išreikšti per meta modelį. Ar visos meta modelio koncepcijos reikalingos? Ar reikia pakeisti duomenis, norint naudoti meta modelį? Ar galima „atgal“ gauti duomenis neprarandant informacijos iš metamodelio?
Techninė kokybė	Meta modelis yra įrankis, pagamintas inžinierių. Ištestuok, „sukrėsk jį“.
Išplečiamumas.	Svarbu vystant programinės įrangos architektūras, kurios gali būti plečiamos ir vėliau plėtojamos. Norint patikrinti, reikia pabandyti norimą meta modelį dirbtinai pakeisti. Viskas gerai, jeigu tai pavyksta ir jei meta modelis pasipildo. Ar tai galima padaryti be kitos dalies panaikinimo? Ar visi buvę komponentai veikia, po išplėtimo?

Dokumentų apibrėžimų kokybė	Atvirkščiai nei programinės įrangos produktui ar API, meta modeliams nėra jokių „įdiegimų nuorodų“. Ką tu matai yra tai, ką gauni, ir viskas ką tu gauni, nors ir apibrėžiama popieriuje ar formalia kalba. Taigi jei tai nėra gerai dokumentuota, tu negali tiesiog įjungti savo įrangos ir „išbandyti“, norint išsiaiškinti ką kas turi reikšti. Pabandykite įsivaizduoti, kad kažkas atsiunčia jums vienos meta modelio koncepcijos įvykį (pvz., bandyti importuojantį programinės įrangos įrankį). Jei tu gauni šio daikto pavadinto „X“ įvykį ar tau visiškai yra aišku ką jis reiškia, net be konteksto?
-----------------------------	--

Meta modelio apimtis

Apibrėžiant meta modelio apimtį ir norint išsiaiškinti ar jis sprendžia jums reikiamą uždavinį, reikia surinkti visas jūsų uždaviniui reikalingas koncepcijas ir palyginti jas su koncepcijomis, esančiomis meta modelyje.

Deja, to gali neužtekti. Panagrinėkime pavyzdį: sakykime, kad jums reikia meta modelio, kuris leistų jums analizuoti klasių reikšmes ir struktūrą. Jūsų koncepcijos atrodytų taip: Klasė, atributas, operacija/metodas, paveldėjimas. Pažiūrėkime, kas bus meta modelyje. Jis turi klases, atributus, operacijas, metodus. Paveldėjimas – tai nėra tai, kas galėtų tapti kuo nors panašiu į meta-klasę. Galbūt meta-ryšių klasė. Iš to seka, vienas paveldėjimas ar daug paveldėjimų? Ar paveldėjimo metodai „virtualūs“ ar „nevirtualūs“? Ar tai galima pasakyti? O kokie vardų konfliktai? Kaip privatus paveldėjimas? (toks kaip c++ kalboje). Kitaip tariant, blogybė yra neakivaizdžiuose dalykuose ir tai daro didelį žymę meta modelio kokybėje.

Geriausias būdas įvertinti meta modelio apimtį yra susidaryti testinių duomenų rinkinį, kuris turi dominančias koncepcijas, kurios yra naudojamos. Taip pat susidaryti sąrašą koncepcijų, kurios šiuo metu nenaudojamos, tačiau kurių ateityje gali prireikti. Tada pažiūrėti ar galima apibrėžti meta modelį modelio terminais. Jeigu galima, tada reikia taip ir aprašyti bei duoti dokumentaciją darbuotojui iš kito skyriaus, kuris nežino kaip jūs galvojate. Leisti jam paaiškinti, ką visi duomenys reiškia, jeigu jis interpretuoja viską teisingai, tada jūsų aprašytas meta modelis yra teisingas ir pirmo lygio.

Meta modelio techninė kokybė

Daug aprašyta, kaip kurti gerą programinę įrangą. O kaip su meta modeliais? Yra tik viena taisyklė juos kuriant, tai kantrybė ir griežtas vertinimas.

Jeigu jūs esate duomenų bazės administratorius: jeigu kas nors jums pasiūlytų naudoti meta modelį kaip konceptualią duomenų bazės schemą jūsų naujam projektui, kurį įmonė tikisi naudoti kitus 10 metų, ar jūs būsite patenkintas?

Jeigu jūs esate programinės įrangos kūrėjas: jūs būsite patenkintas, jeigu meta modelį reikės panaudoti duomenų struktūrose, kiek tai užims jums laiko? Ar viskas veiks, kad realaus pasaulio vartotojas galės normaliai naudotis? Kas nutiks, jeigu vartotojas sumąstys kokią keistą idėją su jūsų programa, ar tai nesukels sunkumų?

Išplečiamumas

Kodėl jūs turėtumėte nerimauti dėl plečiamumo? Yra dvi priežastys:

- Pirma, dažniausiai meta modelių naudojimo blogybė yra tai, kad jie ne visada gali būti naudojami realiame pasaulyje. Taigi juos reikia plėsti, dažniausiai šis papildomas informacijos pridėjimas, jau nebeišreiškia tikrojo meta modelio originalumo.

- Antra, iš esmės meta modeliai turi turėti plečiamumo galimybę, nes jeigu užsakovas norės praplėsti programų galimybes, nereikės perdaryti viso projekto iš pagrindų, o tik pridėti tam tikrą dalį.

Taigi jūs norite patikrinti ar meta modelis gerai išvystytas plečiamumo prasme. Norint tai atlikti jūs turite atsakyti į šiuos klausimus: Ką jūs galite išplėsti? Naujas koncepcijas? Naujus ryšius tarp koncepcijų? Išskirti koncepcijas? Sujungti jas? Pridėti daugiau informacijos prie koncepcijų? Ar praplėsti galima daug, ar tik keliomis koncepcijomis?

Tik nereikia pamiršti koregavimo: jeigu įvairūs žmonės plečia meta modelį, ir bus neišvengiama, kad vieni naudos vieną versiją, kiti kitą lygiagrečiai.

Dokumentų apibrėžimų kokybė

Jeigu jūs manote - „kad, baigtis meta modelio dokumentacijoje yra kaip ir programinės įrangos dokumentacijoje, būtų gerai jei ją turėtumėme“. Meta modelio dokumentacija yra galutinis produktas. Nėra nieko, kas ją pakeistų, pvz., kodas, kuris gali būti paleistas siekiant išsiaiškinti, ko neapibrėžia dokumentacija. Jei jūsų meta modelis nėra gerai apibrėžtas (gerai dokumentuotas), jis yra kaip programinė įranga), kuri nėra baigta, ir jūs ją turite pabaigti (be jokios dokumentacijos). Pažvelkime į blogą pavyzdį, iliustruojantį šį dalyką (tai yra artima mums matytiems kai kuriems realaus pasaulio pavyzdžiams, sukurtiems žmonių, kurie tikrai turėjo žinoti geriau):

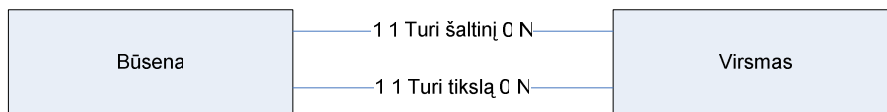


1 pav. Meta modelio fragmentas

Akivaizdu – šis meta modelio fragmentas turėtų išreikšti, kad yra Būsenos ir Virsmai ir kad jie susiję. Šiek tiek sarkastiškai sakant, mes tai būtume žinoję net ir be meta modelio. Jei šis grafikas yra visa duota dalyko būsenų ir virsmų dokumentacija, ji nepadės niekam, nes *ji neišreiškia nieko naudingo*.

Klausimai, į kuriuos pateiktas aukščiau fragmentas negali atsakyti, yra: kiek Virsmų eina su kiek Būsenų?; ką dar žinome apie Būsenas: ar jos turi pavadinimus?; ar turi Virsmai pavadinimus?; ar vardai turi būti unikalūs?; ar virsmai iš vienos būsenos į tą pačią būseną yra leidžiami?; kaip dėl 'daugkartinių Virsmų tarp tų pačių dviejų Būsenų?; kiek trunka vienas Virsmas?; ir taip toliau... Geras meta modelis atsako į visus šiuos klausimus ir dar daugiau. Galų gale ar kitaip galėtume suprasti Būsenos reikšmę?

Pažvelkime į vidutinišką to paties dalyko pavyzdį. Tai vis dar nėra net iš tolo adekvatu.



2 pav. Meta modelio fragmentas su papildoma informacija

Dokumentacija taip pat duoda sekančią papildomą informaciją:

2 lentelė. Dokumentacijos papildoma informacija 1

Būsena	Gali būti nedelsiant	
Vardas	Eilutė	Privalomas: Taip
Yra pirminis	Boolean	Privalomas: Ne
Yra galutinis	Boolean	Privalomas: Ne

3 lentelė. Dokumentacijos papildoma informacija 2

Virsmas	Gali būti nedelsiant	
Trukmė	Laikas	Privalomas: Ne
Paleidžiamas įvykis	Eilutė	Privalomas: Taip
Atliktas veiksmas	Eilutė	Privalomas: Ne

Kodėl mes teigiame, kad visa ši detalai atrodanti informacija yra neadekvati? Daug priežasčių, štai kelios: iš kur jūs žinote, kad tai, ką jie pavadino Būseną yra tai, kas jūsų įsivaizduojama Būseną? (visada padeda pervadinti sąvoka „Foo“ argumentu ir tada matyti ar mes vis dar žinome, kas tai yra). Tas pats ir Trukmei ir kitiems atributams. Kur yra meta-ryšių dokumentacija? Ką reiškia „šaltinis“? Jei jūs linkę atsakyti „juk tai akivaizdu, jūs kimbate prie smulkmenų“, mes būtume linkę atsakyti tą kartą, mes radome 12 (!) skirtingų reikšmių aktyviam žodžio „Funkcija“ naudojimui viename vienos įmonės skyriuje. Ir tai buvo aktyviai jų neieškant. Ar kirsite lažybų dėl savo kito projekto sėkmės dėl šito?

Taip pat vienintelis dalykas, kurį žinome apie Būseną ir Virsmą yra tai, kad jie yra nupiešti ankstesnėje diagramoje ir lentelėje. Bet kas tai yra? Kokia yra kalba, kuria mes apibūdiname Būseną ir Virsmą? (Tai yra meta modelio paskirtis. Be gerai apibrėžto meta modelio jūs neturėtumėte supratimo ką visa tai reiškia – vienintelis aiškus dalykas yra tai, kad yra langeliai diagramoje ir lentelės tekste, ir tai nėra geras pagrindas, kuriuo galima remtis.)

Kaip pašalinė remarka, kaip ir minėjome anksčiau, meta modelio įvykdomumo kokybė taip pat priklauso nuo to, kam meta modelis yra sukurtas. Pavyzdžiui, ankstesniame paveikslėlyje įsivaizduokite, kad minimalus Virsmas turi Šaltinį. Būseną ir Virsmas turi Tikslą. Būsenos dydžiai yra abu po 1. Tai reiškia, kad Virsmai negali egzistuoti be abiejų, šaltinio ir tikslo būsenų. Tai gali būti gerai, bet įsivaizduokite, kad šis meta modelis yra naudojamas kaip duomenų saugykla, kurioje jūsų įrankis išsaugo savo duomenis, schema. Įsivaizduokite, kad dabar pietų pertrauka, jūs norite išeiti, bet jūs dar nebaigėte su savo būsenos virsmo modeliu; jūs turite Virsmą, kuris laikinai nuveda niekur, nes jūs dar nebaigėte. Šiuo atveju duomenų saugykla atsisakytų išsaugoti jūsų modelį, nes jis pažeidžia dydžių suvaržymą. Tai yra priverstinis jūsų pietų atidėjimas galėjo būti išvengtas, padarant minimalius dydžius abiem meta-ryšiams lygius 0. Bet šiuo atveju jūs negalėtumėte naudoti savo meta modelio automatiniam savo būsenos virsmo modelio tinkamumo tikrinimui, nes jis nežinotų, kad nesujungti Virsmai yra greičiausiai „neteisingi“.

Kitaip pasakius, tai yra sudėtingas dalykas. Taip, meta modeliavimas nėra širdies alpimui. Bet tada, jei panaudota teisingai, tai sumažina augančią sudėtingų projektų riziką didesniu faktoriu, nei bet koks kitas matavimas, nei mes sužinome, kai turime pagrindinę infrastruktūrą, kaip nustatymų valdymą ar klaidų stebėjimą.

2.4 Meta modeliavimo savybės

Meta modeliavimas, kaip bet kuri kita disciplina turi savo terminus. Deja, tai ką mes vadiname meta modeliavimu čia, kitos disciplinos vadina savaip. Tuos pačius metodus, modelius kiekviena disciplina įvardina savaip. Taigi pabandyti apibendrinti visus tuos skirtumus.

Apibrėžiame pagrindines disciplinas, kurios susijusios/sutampa jų kūrimo technologija:

- Objektinis modeliavimas
- Pasaulinis tinklas, paremtas semantika, artėjantis prie dirbtinio proto. Šiuo metu daugiausiai naudojami formatai yra RDF, RDF-Schema ir XML-schema.
- STEP pasaulis, kuris atsižvelgia į duomenų specifiką, duomenų apsikeitimo problemas pramoninėje inžinerijoje.

4 lentelė. Modeliavimo terminų palyginimas

Objektinis modeliavimas	Semantinis tinklas	STEP	Apibūdinimas
meta	(meta)	Nėra atskiro termino	Tipas
Meta modelis	ontologija, schema	Programinis protokolas	Semantinis informacinis modelis.
meta-duomenys	meta-duomenys (ontologiniai)		Meta modelio forma, nėra pilnos informacijos, tačiau gali būti tarkim atributų rinkinys.
meta-meta-modelis	Nėra atskiro termino	Išraiška	Kalba kuria išreiškiami metamodeliai.
Keturių lygių architektūra	Nėra tikslaus atitikmens	Nėra atskiro termino	Meta modeliavimo konceptuali architektūra.
Integruotas meta-modelis (lyginant ir ne integruota meta-modelį)	Nėra atskiro termino	Nėra atskiro termino	Meta modelis, kuris apima keletą disciplinų ir susieja jas integruojant.
meta-klasė, kaip meta-būsena.	Klasė		Objektinio modeliavimo klasės koncepcija, bet pritaikyta meta modeliavimui.
Meta-ryšys, meta-asociacija.	Būsena		Ryšio koncepcija, tačiau pritaikyta meta modeliui.
Meta-atributas	Ypatybė		Objektinio modeliavimo atributo koncepcija, pritaikyta meta modeliavimui.
Meta-objektas			Bet koks objektas, naudojamas meta modeliavime.
Meta-objektų būsenos			Bet koks objektas, kuris meta objekto būsenos.

3. Meta modelių pagrįstas portalas

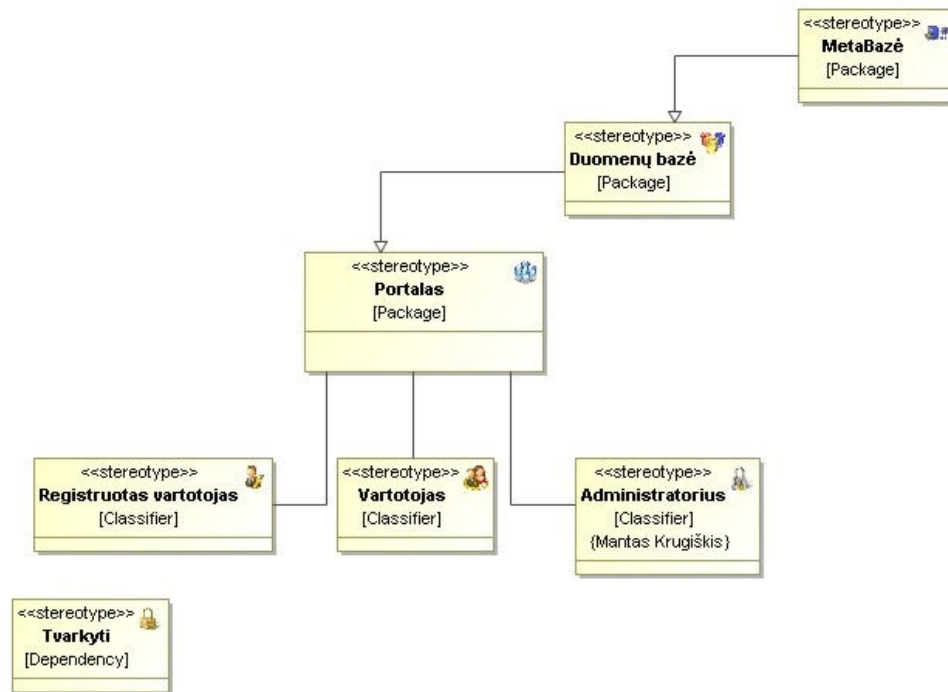
3.1 Esama situacija

Šiuo metu daugumas portalų yra kuriami naudojantis paprasčiausia turinio valdymo sistema. Tačiau ar tai leidžia tolesniam vartotojui lengvai specifikuoti portalą? Patogiausias būdas išspręsti šią problemą yra pasinaudoti meta duomenimis ir meta modelių. Turint paprastą duomenų bazę, reikia sudaryti naują bazę, tik aukštesnio lygio, kurioje būtų specifikuoti visi duomenų elementai, galintys dalyvauti portale. Tokiu atveju ateityje, norint pakeisti portalą specifika, tereikės į meta bazę įkelti naujus elementus, o nereikės perkurti visos struktūros. Taip bus sutaupomas laikas ir žinoma pinigai.

3.2 Meta modelio kūrimas, pasinaudojant MagicDraw

Kuriant portalą, pirmiausia reikėjo sukurti modelį, pagal kurį būtų jis projektuojamas. Buvo pasirinkta projektavimo priemonė MagicDraw, kuria naudojantis sukurta struktūra, kuri aprašyta stereotipais. Naudojantis stereotipais galima savarankiškai sukurti UML plėtinių stereotipus, diagramas bei diagramų stilius.

Susikūrus stereotipus, buvo išpildyta portalo struktūra, pavaizduota 3 paveiksle.



3 pav. Portalo stereotipai, aprašyti MagicDraw

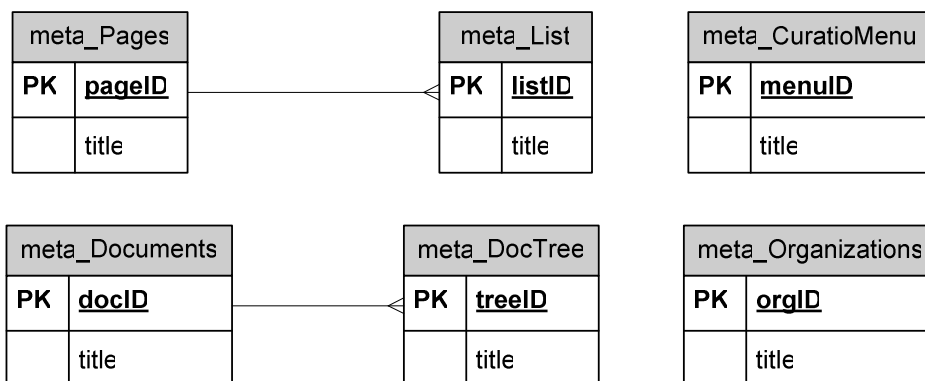
Iš stereotipo matyti esminis dalykas, kad buvo sukurta duomenų bazė ir virš jos, aukštesniame lygyje buvo sukurta meta bazė, kurioje yra esminė portalo struktūra – portalo meta modelis.

Buvo išpildytos šios vartotojui teikiamos paslaugos:

- Vartotojo registracija
- Šamatos skaičiavimas
- Šamatos saugojimas duomenų bazėje
- Portalo valdymas
- Portalo koregavimas
- Vartotojui paslaugas teikiančių įmonių sąrašas.

Šiom paslaugom įgyvendinti buvo sudaryta duomenų bazė iš 15 lentelių, kurių 12 yra susietų tarpusavyje.. Likusios nesusietos ryšiais, nes jos tarpusavyje nepriklausomos. „Pages“ ir „List“ tarpusavyje susietos vienas su daug ryšiu, pagal puslapio „pageID“ identifikatorių.

Kuriant meta duomenų lygį buvo sudarytas aukštesnis duomenų bazės lygis – meta lygis.



4 pav. Meta bazės struktūra

4. Išvados

Straipsnyje supažindinama su meta modeliu, jo standartais bei naudojimo būdais. Išanalizuotos panašios kūrimo technologijos ir meta modelius. Pasiūlytas portalo kūrimo principas, pasinaudojant metamodeliu. Meta modelio naudojimas leidžia daug efektyviau valdyti portalą: galima lengviau specifiikuoti, pertvarkyti, koreguoti esančius duomenis. Mūsų atveju sukurto portalo nauda yra:

- Supaprastintas portalo valdymas, nes portalo struktūra yra apibrėžta meta duomenų lygyje.
 - Greita informacija.
 - Taupomos eksploatacinės išlaidos.
1. Meta duomenys ir meta modelis – patogiausias metodas portalams kurti.
 2. Meta modeliai – duomenų integracijos pagrindas. Jie leidžia programinę įrangą pritaikyti industriniams standartams.
 3. Meta modelių paremta programinė įranga yra e-verslo integravimo kompleksas.
 4. Meta modeliai yra idealūs padėjėjai kompleksiniais web servais paremtuose projektuose.

Literatūros sąrašas

1. Lexbe. New Approaches Required for e-Discovery and Metadata. [žiūrėta 2007-11-14]. Prieiga per internetą: <http://www.lexbe.com/hp/indepth-e-discovery-rule-metadata.htm>
2. Making a List of Data About Metadata and Exploring Information Cataloging Tools. 1998 [žiūrėta 2007-11-14]. Prieiga per internetą: <http://www.fortunecity.com/skyscraper/oracle/699/orahtml/dbmsmag/9803d05.html>
3. R. Todd Stephens (2003). Utilizing Metadata as a Knowledge Communication Tool. Proceedings of the International Professional Communication Conference 2004. Minneapolis, MIN: Institute of Electrical and Electronics Engineers, Inc.
4. Knowledge Systems, AI laboratory. Stanford University. What is an Ontology? [žiūrėta 2007-11-14]. Prieiga per internetą: <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>

PECULIARITIES OF INFORMATIVE SYSTEM BASED ON META MODEL

This article analysis the operation of meta model-based systems using models. It also surveys the quality of meta model and peculiarities of its creation; as well as describes the differences in terms between various ways of modelling techniques to be used in creation of a good quality meta model and adjusting it to further projects.