

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMACIJOS SISTEMŲ KATEDRA

Donatas Dambrauskas

**Informacinių sistemų tezauro modelis ir jo
įgyvendinimas internete**

Informatikos mokslo magistro baigiamasis darbas

Darbo vadovė: doc. dr. Lina Nemuraitė

KAUNAS 2004

TURINYS

SANTRAUKA.....	4
SUMMARY	5
1. ĮVADAS	6
1.1. Darbo tikslas	6
1.2. Sistemos vartotojai	7
1.3. Informaciją apie projekto užsakovą	7
2. TURINIO VALDYMO SISTEMŲ KŪRIMO METODŲ ANALIZĖ	8
2.1. Dalykinės srities modeliais grindžiami metodai	8
2.2. UML profilis interneto svetainėms projektuoti	10
2.3. Semantinio žiniatinklio metodai	18
2.4. Esamų metodų trūkumai ir siekiamos savybės	21
2.5. Analizės išvados.....	23
3. SIŪLOMAS TURINIO VALDYMO SISTEMŲ KŪRIMO METODAS 25	25
3.1. Kūrimo etapai	25
3.2. Dalykinės srities modeliavimas	26
3.3. Navigavimo projektavimas.....	27
3.4. Pateikties elementų tipai.....	28
3.5. Svetainės architektūros projektavimas	29
3.6. Paieška.....	30
3.7. Vartotojų teisių valdymas	31
3.8. Daugiakalbystės realizavimas.....	32
4. INFORMACINIŲ SISTEMŲ TEZAURO MODELIS.....	35
4.1. Veiklos kontekstas	35
4.2. Panaudojimo atvejų modelis.....	36
4.3. Klasių diagramos.....	37
4.4. Panaudojimo atvejų sekų diagramos	39
4.4.1. Vartotojo registravimasis sistemoje	39
4.4.2. Vartotojo prisijungimas prie sistemos.....	39
4.4.3. Reikalingų duomenų paieška	40
4.4.4. Dokumentų siuntimas.....	41
4.4.5. Dokumentų parsisiuntimas	41
4.4.6. Terminų žodyno panaudojimas.....	42
4.4.7. Sistemos sąsajos tvarkymas	42
4.4.8. Daugiakalbystės tvarkymas	43
4.4.9. Statistikos peržiūra	43
4.4.10. Sistemos parametrų nustatymas	44
4.5. Objektų būsenų kaitos diagramos	44
4.5.1. Vartotojas.....	44
4.5.2. Dokumentas	45
4.6. Kuriamos sistemos funkcijų sąrašas.....	45
4.7. Tinklapių žemėlapis	55
5. PROGRAMINĖ REALIZACIJA INTERNETE	56
5.1. Apibendrintas architektūros modelis.....	56
5.2. Sistemos struktūra	58

5.3.	Paieškos realizavimas.....	59
5.4.	Daugiakalbystės realizavimas.....	60
5.5.	Vartotojo prieigos realizavimas.....	61
5.6.	Paskirstymo diagrama.....	62
6.	IŠVADOS.....	63
7.	LITERATŪROS SĄRAŠAS.....	64
8.	TERMINŲ IR SANTRUMPŲ ŽODYNAS.....	65

SANTRAUKA

Didėjant web taikomųjų programų galimybėms informacijos publikavimo uždaviniams spręsti, vis daugiau švietimo ir mokymo organizacijų savo sukauptą informaciją talpina ir publikuoja žinių portaluose. Toks sprendimas leidžia efektyviai pasinaudoti visa organizacijoje sukaupta metodine bei praktine medžiaga.

Dauguma universitetų visame pasaulyje renkasi specializuotus žinių portalus, internetinius tezaurus, kurie leidžia pateikti turimą informaciją įvairiais pjūviais bei susieti vienos tematikos duomenis tarpusavyje. Tokio sprendimo dėka reikalinga informacija gali būti pasiekama iš bet kurio planetos taško, bet kuriuo paros metu, o tai leidžia efektyviai išnaudoti nuotolinio mokymo galimybes.

Šiame darbe yra pateikiamas pavyzdys, kaip internetinis tezaurus gali padėti publikuojant metodinę ir praktinę medžiagą, kaupiant teorinius terminus. Atliekama egzistuojančių tokių svetainių kūrimo metodų analizė bei aprašomas sukurtas specifinis metodas tezaurui analogiškoms web taikomosioms programoms kurti. Nurodomi sprendimai, leidžiantys realizuoti daugiakalbystės funkciją, efektyviai išnaudoti sistemos resursus, sukurti naudingą paieškos sistemą bei kontekstinį meniu, sudaryti naudingą turinio valdymo mechanizmą.

SUMMARY

Nowadays possibilities of creation of web applications dedicated for information presentation a rapidly increasing, and more universities and other education organizations prefer to publish the information they gathered in specialized information portals. This solution allows effective use of all methodical and practical material, collected in organization.

More universities in all over the world made their decision on specialized portals of knowledge, internet thesaurus. These web applications allows present information in different views and connect related data. Using this solution gives possibility to search for useful information from all over the world, 24 hours a day, so this gives chance to improve possibilities of distant learning.

In this work web thesaurus is presented as an example, how specialized web application can help to publish methodical and practical information and gathering terms of theory and their descriptions. Also presented analysis of existing methods for creating specialized web applications and description of method, created for modeling web applications analogical to web thesaurus. There are special solutions described for creating of multilingual web applications, effective system resource using, creating useful search system and context menu or content management mechanism.

1. ĮVADAS

Metodinės ir praktinės medžiagos publikavimas internete šiuo metu yra labai paplitęs didžiausiuose Europos bei Amerikos universitetuose. Toks metodas leidžia maksimaliai išnaudoti web taikomųjų aplikacijų tiekiamus privalumus – informacijos pateikimą įvairiais pjūviais, paieškos mechanizmo efektyvumą ir nepertraukiamą pasiekiamumą, kuris yra įtakojamas duomenų ieškančio asmens buvimo vietos.

Egzistuoja įvairūs sprendimai specializuotų internetinių portalų kūrimui ir pasirinkti tinkamą metodiką bei sukurti naudingą programinį produktą galima tik atlikus išsamius analizės, projektavimo bei realizavimo darbus. Būtent toks darbas ir buvo paskirtas šio dokumento autoriui, kurio rezultatas turi būti veikiantis ir naudojamas programinis produktas – Informacijos sistemų katedros žinių portalas.

1.1. Darbo tikslas

Pagrindinis šio magistrinio darbo tikslas buvo sukurtas Informacinių sistemų tezaurą, veikiantį interneto aplinkoje. Informacinių sistemų tezauras – tai specifinis žinių portalas. Pagrindiniai uždaviniai, kurie buvo keliami ir įgyvendinti šiame projekte yra daugiakalbystė, leidžianti informaciją portale pateikti keliomis kalbomis, vartotojų teisėmis grindžiama prieiga prie sistemoje pateikiamų resursų, įgalinanti kontroliuoti publikuojamus duomenis bei jų koregavimą, terminų žodynas ir jame pateiktų terminų susiejimas su sistemoje saugomais dokumentais, lokalsios bei išorinės paieškos sistemos ir kontekstinis meniu, leidžiantys lengvai rasti reikiamus duomenis.

Sukurto programinio produkto administravimo modulis leidžia lengvai kontroliuoti vartotojų atliekamus veiksmus, rinkti statistiką bei keisti portale pateikiamos informacijos turinį. Siekiant pagreitinti sistemos veikimą ir taip palengvinti vartotojo darbą buvo realizuotas pakartotinis sistemos resursų panaudojimo mechanizmas, grupuojant turinio formavimo elementus ir juos kešuojant.

Siekiant kuo geriau įgyvendinti šį darbą, buvo atlikta išsami turinio valdymo sistemų kūrimo metodų analizė, kurios metu išnagrinėti jau egzistuojančių metodologijų privalumai ir skirtumai bei sudarytas savitas turinio valdymo sistemų kūrimo metodas, kuris perimta geriausias nagrinėtų metodų savybes ir išplečia Informacinių sistemų tezaurą bei analogiškų internetinių sistemų kūrimo galimybes. Tokiu būdu sukurta kūrimo metodika specifinėms, tačiau šiuo metu aktualioms ir sparčiai besivystančioms internetinėms taikomosioms programoms, kurioms yra keliami vis platesni reikalavimai, bei pateikta šios metodikos pagrindu sukurtas programinis produktas – Informacinių sistemų tezauras.

1.2. Sistemos vartotojai

Numatomi kuriamos sistemos vartotojai – Kauno technologijos universiteto informacijos sistemų katedros dėstytojai bei studentai, o taipogi kitų universitetų bei šalių informacinių sistemų srities specialistai. Jiems keliami minimalūs ir elementarūs žinių apie internetines taikomas programas ir interneto naršyklės reikalavimai.

Jei sistema susidomės kitų mokslo sričių atstovai ir ji bus diegiama kitose katedrose, vartotojų skaičius ir įvairovė gali smarkiai padidėti.

Lietuvoje nėra analogiškų sistemų, sukurtų ir naudojamų mokslo įstaigose, todėl tikėtina, jog sukūrus visus užsakovo reikalavimus atitinkantį ir efektyviai dirbantį programinį produktą, jis gali būti reikalingas kitoms mokslo įstaigoms ir nepelno siekiančioms organizacijoms. Visi programinio produkto platinimo už Kauno technologijos universiteto ribų klausimai turi būti suderinti su sistemos autorinių teisių savininke docente dr. Lina Nemuraite.

1.3. Informaciją apie projekto užsakovą

Programinio produkto užsakovas yra Kauno technologijos universiteto Informacijos sistemų katedra, kurią atstovauja docentė dr. Lina Nemuraitė. Jai suteikiamos teisės platinti ir diegti sukurtą sistemą Kauno technologijos universitete, jai priklauso ir pačio produkto autorinės teisės.

2. TURINIO VALDYMO SISTEMŲ KŪRIMO METODŲ ANALIZĖ

2.1. Dalykinės srities modeliais grindžiami metodai

Dauguma esamų projektavimo modelių yra pagrįsti dalykinės srities modeliavimu ir pagrindinį dėmesį skiria hipermedijos programų projektavimo gyvavimo ciklo architektūros etapui [2]. Visi šie modeliai akcentuoja palaipsnių ir interaktyvų projektavimo procesą ir dažniausiai susideda iš keletos ontogonaliųjų modeliavimo lygmenų. Dažniausiai programos projektavimo metu yra naudojami šie modeliavimo lygmenys: konceptualus arba struktūrinis lygmuo (nusakantis informacijos srities struktūra), hiperteksto lygmuo (parodantis kuriamos programos kompoziciją bei navigacijos struktūrą), vaizduojamasis arba prezentacinis lygmuo (nusakantis vartotojo sąsają) personalizacijos lygmuo (individualus dizainas), įgyvendinimo arba realizacijos lygmuo.

Kiekvienas iš nagrinėjamų modelių skirtingai dengia visus šiuos lygmenis, tačiau dauguma jų akcentuoja ir didžiausią dėmesį kreipia į tris lygmenis – konceptualų, hiperteksto ir vaizduojamąjį [5].

Konceptualiajame lygmenyje informacijos sritis yra užfiksuojama ir modeliuojama pasinaudojant trimis architektūros technologijomis:

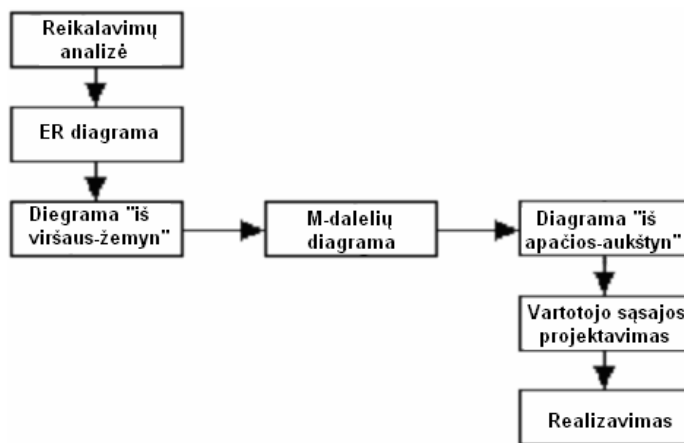
- ✦ Esybių sąryšių metodika – informacijos objektai ir duomenų struktūra yra nusakoma esybių bei jų ryšių reikšmėmis.
- ✦ Objektiškai orientuota metodika – informacijos objektai yra modeliuojami kaip objektai arba klasės.
- ✦ Ontologija pagrįsta metodika – informacijos objektai yra modeliuojami kaip ontologijos klasės.

Įvairių požiūrių ir perspektyvų koncepcija yra naudojama hiperteksto lygmenyje siekiant modeliuoti įvairių tipų programas, nagrinėjant struktūrinio lygmens analizės ir projektavimo rezultatus. Keletas metodų, tokių kaip WSDM, OOHDM ir WebML suteikia galimybę lankstesnėms personalizacijos ypatybėms nagrinėti (turinio, nuorodų, struktūros ir konteksto individualiam pritaikymui). Modeliuojamos taikomosios programos kaip sistemos komponavimas ir navigacijos struktūra yra sudaroma mazgų (puslapių, navigacijos elementų, turinio elementų) ir įvairių ryšių (perspektyvos, struktūrinių, sisteminių bei kt.) tarp jų pagrindu. Navigacijos elementai (mazgų vienetai) yra susieti su koncepciniais elementais (esybėmis arba klasėmis) tam kad būtų galima pateikti informaciją bei duomenis vaizdavimo metu [7].

Nesunku pastebėti kad projektavimo metodai ir modeliai, naudojami modeliuojant internetines programas, turi pakankamai daug panašumų. Pagrindinis jų skirtumas yra nagrinėjamų lygių išsamios analizės apimtis, projektavimo metu suteikiamų galimybių lygmuo skirtinguose veiklos etapuose.

HDM (angl. Hypermedia Design Model) yra vienas iš ilgiausiai naudojamų projektavimo modelių, paremtas esybių ryšių modeliavimu ir padedantis nustatyti didelių ir tik informaciją pateikiančių (bet nepriimančių (tik skaitymui) hipermedijos sistemų sąveikas ir struktūrą. Šis modelis yra tinkamas sritims pasižyminčioms aukštu organizavimo, moduliavimo ir vientisumo lygiu. Naudojant šį modelį pagrindinis dėmesys yra skiriamas hierarchiniam informacijos objektų aprašymui. Šie objektai apibrėžiami kaip esybės sudarytos iš komponentų, savyje talpinančių informacijos vienetų bei navigacines struktūras, neatsižvelgiant nuo jų realizacijos. Navigacinė struktūra yra sudaroma iš perspektyvinių ryšių tarp vienetų, struktūrinių ryšių tarp komponentų, taikomosios programos ryšių tarp esybių, komponentų arba vienetų [4]. Vaizdavimo projektavimas yra sudaromas informacijos vienetų bei karkasų (informacinių vienetų grupių) pagrindu.

RMM (angl. Relationship Management Methodology) yra esybių ryšiais pagrįstas internetinių taikomųjų programų modelis, tinkamas struktūrizuotoms hipermedijos programoms [3]. Pats projektavimo procesas yra suskaidytas į septynis žingsnius (matomus pav. 1): esybių bei jų ryšių projektavimas, sluoksnių projektavimas (esybių atributų grupavimas vaizduojant juos mazgais bei vaizdavimo vienetais, vadinamais dalelėmis arba M-dalelėmis (modeliavimo dalelėmis), navigacijos projektavimas (prieigos tipo nustatymas organizuojant nuorodas, meniu, indeksus, supažindinimo kelius), protokolo konvertavimo projektavimas (verčiant projekto komponentus į fizinius objektus), vartotojo sąsajos projektavimas (vaizdavimo maketavimas), veikiančios sistemos funkcionavimo projektavimas, konstravimas ir testavimas.



Pav. 1. RMM procesų seka

OOHDM (angl. Object Oriented Hypermedia Design Model) yra objektiškai orientuotas projektavimo modelis kuris leidžia specifikuoti hipermedijos programos kaip navigacijos pjūvius konceptualiaame modelyje. Projektavimo procesas čia susideda iš keturių lygmenų: konceptualaus modeliavimo, navigacinės sistemos architektūros, abstrakčios sąsajos projektavimas ir realizavimas, prie kurių visai

nesenai buvo pridėti reikalavimų rinkimas ir personalizacijos projektavimas. Navigacijos mazgai yra susiejami su konceptualiomis klasėmis, o tik skaitymui skirtų internetinių taikomųjų programų generavimui yra sukurtas specialus CASE įrankis vadinamas OOHDM-Web.

SOHDM (angl. Scenario-based object-oriented Hypermedia Design Methodology) yra kitas objektiškai orientuotas požiūris į internetinių programų modeliavimą. Šis modelis didžiausią dėmesį skiria į procesą orientuotoms hipermedijos sistemoms, kurios padeda palaikyti organizacijoje vykstančius procesus [9]. Scenarijai yra sukuriami nagrinėjant organizacijos veiklą ir tai tampa pagrindu objektiniam ir modeliavimui ir navigacijos projektavimui. Projektavimas susideda iš šešių fazių: aplinkos analizės, objektiškai orientuoto modeliavimo, pjūvių kūrimo, navigacijos sukūrimo, realizacijos architektūros sudarymo, konstravimo.

WDSM (angl. Web Site Design Method) yra į vartotoją nukreiptas projektavimas, kadangi modeliuojant yra panaudojamas vartotojų modelis, sukuriamos vartotojų klasės ir jų ypatybės bei pjūviai. Dizaino procesas susideda iš trijų stadijų: vartotojų modeliavimo, konceptualaus projektavimo, realizacijos architektūros sudarymo, konstravimo.

EORM (Enhanced Object-Relationship Model) – objektiškai orientuotas projektavimo modelis, kurio pagrindinė savybė yra ta, kad čia ryšiai tarp objektų yra apibrėžiami kaip atskiri objektai. Visi šie ryšio bei sąveikos objektai yra išreiškiami kaip atskiros klasės. Metodas yra paremtas trimis karkasais – klasių karkasu, kompozicijos karkasu, grafiniu vartotojo sąsajos karkasu. Šio metodo savybe taip pat galima pavadinti ankstyvą vartotojo sąsajos projektavimą.

2.2. UML profilis interneto svetainėms projektuoti

Modeliavimas padeda lengviau suprasti kai kurias sistemos detales. Modeliavimo pasirinkimas turi didžiulę įtaką sprendžiamos problemos supratimui ir pačio sprendimo formai. Internetinės taikomosios programos, kaip ir kitos programinės įrangos sistemos, yra pavaizduojamos pasitelkiant modelius: panaudojamumo modelis (angl. use case), realizacijos modelis (angl. implementation model), išdėstymo modelis (angl. deployment model), saugumo modelis (angl. security model) ir kt. Išskirtinis internetinių programų modelis yra tinklapio žemėlapis, kurį sudaro abstraktus web puslapių bei navigacijos kelių, galimų esamoje sistemoje, rinkinys.

Sprendžiant kaip ir ką modeliuoti, labai svarbu nustatyti abstrakcijos ir detalizavimo lygmenį, rasti optimalų variantą sukuriant rezultata, kuriuo bus patenkinti modelio naudotojai. Paprastai geriausia ir naudingiausia modeliuoti sistemos artefaktus, t.y. tas realaus gyvenimo esybes, kurios bus sukurtos ir kuriomis bus manipuluojama siekiant realizuoti galutinį produktą - išbaigtą sistemą [11]. Web serverio vidaus komponentų modeliavimas ar interneto naršyklės detalių modeliavimas neduos

laukiamų rezultatų ir nepalengvins sistemos architektų bei projektuotojų darbo. Tinklapių modeliavimas, nuorodų tarp jų išskyrimas, dinaminio turinio, pateikiamo juos sudarant ir perduodamo klientui analizė yra labai svarbi. Tinklapiai, nuorodos tarp jų, dinaminio turinio elementai serverio bei kliento pusėje – tai ir yra tie sistemos artefaktai, kuriuos privalu modeliuoti ir analizuoti.

Norint atlikti modeliavimą, reikia sistemos elementus susieti su modelių elementais. Lengviausiai tai padaroma su nuorodomis, kurios automatiškai virsta ryšiais tarp modelio elementų. Tinklapiai (interneto puslapiai) gali būti suprantami kaip klasės loginiame modelio pjūvyje [7]. Tinklapius apibrėžus kaip klases, visos kodo funkcijos, naudojamos juose automatiškai virsta klasių operacijomis. Tinklapių lygmenyje naudojami programavimo kodo kintamieji yra paverčiami klasės atributais. Problema iškyla kuomet tinklapyje yra naudojama kodo operacijų aibė, kuri veikia serverio pusėje, formuojant dinaminį tinklapių turinį, bei visiškai skirtingi kliento pusės kodo elementai (pvz. JavaScript), valdantys sąsajos elementus ir atliekantys pirminę įeinančių duomenų kontrolę arba dalinį tinklo navigacijos sistemos valdymą. Tai sukelia sumaištį, kadangi vienu atveju turime dar nesuformuotą web puslapį serverio pusėje, kitu atveju turime jau kitą aktyvią dalį, kuri yra pateikiama kliento dalyje. Taigi paprastas web puslapio susiejimas su UML klase nepadės projektuotojams geriau suprasti kuriamos sistemos.

UML kūrėjai numatė atvejį, kada jų pateikta išbaigta UML semantika nesugebės padengti specifinės nagrinėjamų sistemų objektų bei komponentų aibės. Tam tikslui yra naudojamas formalus UML išplėtimo mechanizmas kuris yra naudojamas standartinės UML semantikos išplėtimui. Šis mechanizmas leidžia nustatyti stereotipus, žymėtąsias reikšmes bei apribojimus, kurie gali būti taikomi modelio elementams.

Stereotipas yra elementas suteikiantis galimybę nustatyti naują semantinę reikšmę modelio elementui. Žymėtosios reikšmės yra pagrindinės reikšmių poros kurios gali būti susietos su modelio elementu taip priskiriant reikšmę šiam elementui. Apribojimus sudaro taisyklės, kurios garantuoja kuriamo modelio teisingumą. Šios taisyklės gali būti išreikštos kaip laisvos formos tekstas arba formalizuotos panaudojant Objektų apribojimų kalbą (angl. OCL - Object Constraint Language).

Bendroji internetinių taikomųjų programų architektūra yra sudaroma analizuojant naršyklių, tinklo bei web serverio galimybes. Naršyklės „pareikalauja“ tinklapių iš serverio. Kiekvieną tokių tinklapių galima apibūdinti kaip turinio bei jo suformavimo taisyklių rinkinį, išreikštą pasinaudojant HTML. Kai kurie tinklapiai naudoja kliento pusės programinius kodus, kurie yra interpretuojami naršyklės. Šie kodo elementai suteikia papildomą dinamiką tinklapių elgsenai bei jo pavaizdavimui, kuri vykdo naršyklė. Dažnai tokie metodai yra naudojami dinamiškai keisti turinio vaizdavimą, papildomas valdykles (apletus, ActiveX komponentus, įskiepus ir kt.) naudojamus internetiniame puslapyje. Web aplikacijos

naudotojas stebi pateikiamą tinklapio informaciją ir sąveikauja su sistema pasinaudodamas jam suteikiamais valdymo elementais – informacijos laukais arba nuorodomis į kitus sistemos tinklapius. Bet kurie vartotojo atlikti veiksmai įtakoja sistemos darbą ir keičia jos veiklos būseną.

Žiūrint į web puslapį iš vartotojo pusės galima teigti, jog tai visuomet bus HTML pagrindu suformuotas dokumentas. Serverio pusėje tas pats puslapis gali keisti savo turinį ir būseną pagal įvairius scenarijus, kurie yra sukuriami pasinaudojant scenarijų rašymo metodologijomis bei kalbomis (CGI, ASP ir kt.).

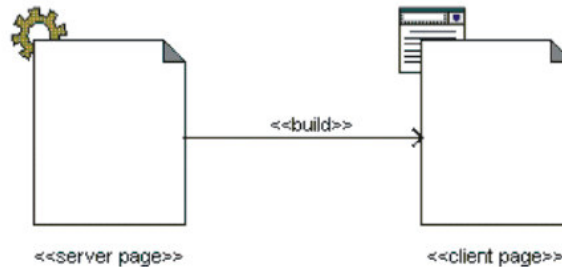
Modeliuojant Web tinklapius kiekvienas interneto puslapis yra susiejami su UML komponentais. Komponentu yra laikoma fizinė ir keičiama kuriamos sistemos dalis. Modelio realizacijos pjūvis (komponentų pjūvis) atvaizduoja sistemos komponentus ir ryšius tarp jų. Internetinių taikomųjų programų atveju, šiame pjūvyje yra atvaizduoti visi sistemą sudarantys tinklapiai ir ryšiai tarp jų (šie ryšiai sudaromi panaudojant nuorodas). Viename lygmenyje web sistemos komponentų diagrama atrodo kaip tinklapio žemėlapis.

Tinklapio komponentai atvaizduoja tik fizines sąsajos dalis, jie nėra tinkami vaizduoti bendradarbiavimą bei sąveiką tinklapio viduje. Toks abstrakcijos lygmuo yra labai svarbus sistemos kūrėjams ir programuotojams ir turi būti modelio dalimi. Kiekvienas web puslapis yra klasė atvaizduojama UML dizaino pjūvyje (loginiame pjūvyje). Atvaizduojant tinklapyje esančius metodus kaip klasės operacijas, o jų naudojamus kintamuosius kaip klasės atributus yra susiduriama su serverio bei kliento pusės metodų nesuderinamumu. Vienas iš šios problemos sprendimo būdų yra panaudoti stereotipus taip kiekvienam klasės atributui arba operacijai nurodant atitinkamą priklausomybę. Tačiau tuomet modeliavimas iš paprasto ir lengvai suprantamo tampa komplikuotu ir sunkiai suvaldomu.

Daug geresnis šios problemos sprendimas yra „interesų išskaidymas“. Mąstant logiškai, web puslapio elgesys serverio ir kliento pusėje yra visiškai skirtingas. Kai tinklapis yra apdorojamas serverio pusėje, jam suteikiama prieiga prie serverio resursų (failų sistemos, duomenų bazių ir kt.). Tas pats puslapis sudarytas iš srauto HTML komandų kliento pusėje yra visiškai kitoks, jei nagrinėsime jo galimus veiksmus bei resursų panaudojimą. Kliento pusėje tinklapis gali naudoti naršyklės resursus ir elementus, pasinaudojant Dokumento Objektų Modeliu (angl. DOM – Document Object Model) ir kiekvienu jame specifikuotu įskiepu, Java apletu arba ActiveX komponento valdykle. Taipogi galimi ryšiai su kitais tuo metu aktyviais puslapiais, esančiais kitoje tinklapio karkaso dalyje arba naršyklės esybėje.

Išskiriant tokius interesus, galima modeliuoti serverio pusės aspektus kaip vieną klasę, o kliento pusės – kaip kitą. Toks išskyrimas į dvi dalis atliekamas pasinaudojant UML stereotipais („serverio

tinklapis“ ir „kliento tinklapis“) ir ikonomis (pav. 2). Taip sukuriami nauja UML semantika modeliuojamiems elementams. Klasė su nurodytu stereotipu gali būti išskirta panaudojant ikonas.



Pav. 2. Kliento ir serverio ikonos

Web puslapių atveju, stereotipas paženklina klasę kaip loginės tinklapių elgsenos abstrakciją serverio arba kliento pusėje. Abi šios klasės yra susiejamos kryptiniu ryšiu, nukreiptu iš serverio pusės tinklapių į kliento pusės tinklapių ir suteikiant ryšiui stereotipą <<sukuria>>, kadangi serverio dalies tinklapis suformuoja kliento dalies tinklapių. Kiekvienas dinaminis tinklapis (toks tinklapis, kurio turinys yra suformuojamas serverio pusėje vykdomų aplikacijų) yra konstruojamas kartu su serverio puslapiu. Dažniausiai vienas serverio puslapis suformuoja vieną kliento pusės internetinį tinklapių, tačiau yra sistemų kuriuose vienas serverio tinklapis formuoja net kelis kliento dalies puslapius.

Paprasčiausias ryšys tarp tinklapių yra nuoroda. Nuoroda web aplikacijoje atlieka navigacijos kelio sistemoje vaidmenį. Toks ryšys tarp tinklapių modeliuose yra vaizduojamas pasinaudojant stereotipu <<nuoroda>>. Tokio stereotipo asociacija visuomet prasideda kliento dalies tinklapyje ir rodo į serverio arba kliento dalies internetinį puslapį.

Nuorodos yra įdiegiamos į sistemą kaip tinklapių paklausa, o web puslapiai yra modeliuojami komponentai, kuomet visi jie vaizduojami realizacijos pjūvyje (Implementation View). Nuorodos pagrindu sudaromas ryšys tarp serverio ir kliento klasių (su nurodytu stereotipu <<sukuria>>) ir paprasta nuoroda sudarytas ryšys (su stereotipu <<nuoroda>>) yra ekvivalentai, kadangi tiek vienu, tiek ir antru atveju šie ryšiai yra tinklapių pareikalavimas.

Žymių reikšmės (tagged values) yra naudojamos aprašyti parametrus, kurie yra perduodami per nuorodas užklausančiam web tinklapiui. Prie nuorodos asociacijos pridėta žymių reikšmė yra sąrašas parametrų vardų, kurie yra naudojami formuojant užklausančiam web tinklapiui.

Stereotipų panaudojimas labai palengvina web tinklapių programinių kodų bei jų naudojamų ryšių modeliavimą. Serverio puslapių stereotipo klasės operacijos tampa atitinkamo tinklapių metodais, o atributai – to tinklapių lygmens globaliais kintamaisiais. Kliento puslapių operacijos tampa kliento pusėje galimais metodais, o atributai – atitinkamo lygmens kintamaisiais. Taip serverio pusės

tinklapiams yra modeliuojami santykiai su serverio pusės resursais, o kliento pusės – su naršyklės elementais, ActiveX valdikliais ir Java apletais.

Vienas iš didžiausių privalumų, pastebimų naudojant klasių stereotipus modeliuojant loginę tinklapio elgseną yra tas, kad jo sąveika ir bendradarbiavimas su serverio pusės komponentais yra išreiškiama taip pat kaip bet kuri kita tik serverio dalies komponentų sąveika. Serverio puslapio stereotipu pažymėta klasė yra tiesiog kita klasė dalyvaujanti sistemos veiklos logikos modeliavime. Daugiau konceptualiame lygmenyje šios serverio lygmens klasės dažniausiai nustatinėja sistemos būsenos valdiklius, taip aktyvuodamos įvairius sistemos veiklos objektus norint sukurti tam tikrą, nuo sistemos einamos būsenos priklausantį, veiklos rezultatą, kurio naršyklės pagalba pareikalavo sistemos naudotojas.

Kliento pusėje vykstančių sąveikų ir bendradarbiavimo modeliavimas gali būti šiek tiek komplikuoatas. Tai įtakoja plati aibė technologijų, kurios gali būti panaudotos šioje dalyje. Paprasčiausias kliento pusės tinklapis yra HTML dokumentas, kuris sudarytas iš valdančiojo navigacinio kodo bei turinio informacijos. Naršyklė naudotojui atvaizduoja tinklapį remdamasi HTML formatavimo instrukcijomis, pateikiamomis jame su kartais panaudojamais vaizdavimo stiliais. Vaizdavimo stilių stereotipas gali būti išskirtas ir suformuota atskira klasė, su priklausomybės ryšiu nuo kliento pusės tinklapio. Tačiau šio stereotipo klasės yra išskiriamos gana retai.

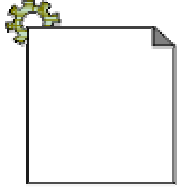
Pagrindinis duomenų įvedimo mechanizmas web puslapiuose yra forma. Formos yra apibrėžiamos HTML tinklapiuose panaudojant `<form>` žymę. Kiekvienai formai yra nurodomas tinklapis, į kurį yra siunčiami surinkti duomenys. Formą sudaro aibė informacijos įvesties elementų, visi jie yra išreiškiami HTML žymėmis. Dažniausiai naudojamos žymės yra `<input>`, `<select>` ir `<textarea>`. HTML žymė `<input>` gali būti daugiareikšmė įvesties elementų formos atžvilgiu, nes ją galima vaizduoti kaip teksto įvedimo lauką, daugybinės arba vienetinės pasirinkties lauką, paslėptas laukas ir kt. Modeliuojant formas sukuriamas dar vienas klasių stereotipas `<<forma>>`. Formos stereotipo klasė neturi operacijų, kadangi bet kokia operacija aprašyta formos žymių viduje yra tinklapio, o ne formos lygmens. Formos duomenų įvedimo elementai yra formos stereotipo klasės atributai. Visiems jiems, atsižvelgiant į duomenų įvesties tipą, yra priskiriamas atitinkamas stereotipas. Formos stereotipo klasė gali turėti ryšių su apletais ir ActiveX komponentais, jei šie yra naudojami kaip informacijos įvesties elementai. Kiekviena forma taipogi turi ryšį su serverio dalies tinklapiu, kuriame yra apdorojami formos surinkti duomenys ir kuris yra nurodytas kaip formos duomenų siuntimo tikslas. Šis ryšys yra parašomas kaip `<<patvirtinimas>>`.

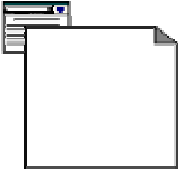
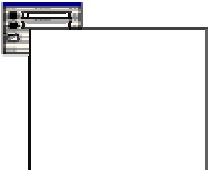
Viena ar kelios JavaScript bibliotekos gali būti aprašytos kaip `<<JavaScript>>` stereotipo klasės. Jų metodai tampa operacijomis, o globalūs kintamieji – atributais.

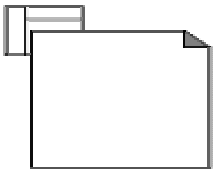
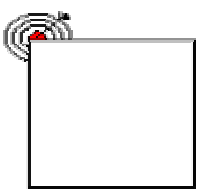
Web aplikacijose vis dažniau yra imami naudoti karkasai (frames), kurie leidžia vienu metu naudotojui matyti kelis tinklapius ir šie tinklapiai gali bendradarbiauti ir sąveikauti tarpusavyje keisdami duomenimis bei aktyvuodami įvairius objektus. Naujausios naršyklių versijos leidžia sąveikauti net skirtinguose naršyklės vienetuose esantiems tinklapiams. Tai yra vykdoma naudojant dinaminio HTML programinį kodą.

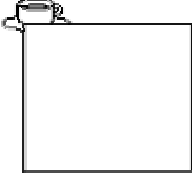
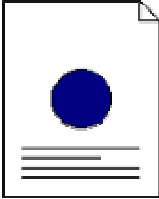
Ar naudoti karkasą web aplikacijoje turi nuspręsti sistemos architektas bei realizuotojas (programuotojas). Jei karkasai yra naudojami, jie turi būti išreikšti ADM (Application Domain Modeling). Modeliuojant karkasų naudojimą atsiranda dar dvi stereotipinės klasės - <<karkasų aibė>> ir <<tikslas>> (frameset, target), bei asociacijos stereotipas <<tikslinė nuoroda>> (target link). Karkasų aibės klasė yra tiesiogiai susieta su HTML <frameset> žyme. Joje yra nurodomi kliento dalies tinklapiai. Tikslų klasė ir yra tinklapis, kuris yra talpinamas karkaso viduje. Jei šie tinklapiai bendradarbiauja tarpusavyje, tai turi būti sukurtas tikslinės nuorodos ryšys. Kaip karkaso dalis yra vaizduojama ne tik vienos naršyklė vaizduojami tinklapiai, bet ir kelių naršyklių tinklapiai, kurie tarpusavyje yra susiję tam tikrais sąveikos mechanizmais. Prie ryšių tarp karkaso ir jo elementų gali būti nurodomos žymimos reikšmės, kurios nusako eilutę bei stulpelį, į kuriuos yra nukreipiamas vaizduojamasis elementas.

Pateikiamose lentelėje 1 yra išskirti visi vaizduojami stereotipų elementai, kurie buvo paminėti bei aprašytas jų vaizdavimas.

Pavadinimas	Metamodelio klasė	Aprašymas	Ikona	Apribojimai	Perduodami parametrai
Serverio tinklapis	Klasė	Serverio tinklapis vaizduoja web puslapį, kuriame yra programinio kodo, kuris vykdomas serverio pusėje. Šis programinis kodas sąveikauja sus serverio resursais (duomenų bazėmis, failais ir kt.). Šios klasės objektai yra programiniam		Serverio tinklapiai gali būti vaizduojami sąveikoje tik su serverio lygmens objektais.	Programinio kodo variklis – kalba arba variklis, kuris turi būti naudojamas programiniam kodui apdoroti (VBScript, PHP, Pearl ir kt.).

		kode aprašytos funkcijos, o atributai – tinklapio lygmens globalūs kintamieji.			
Kliento tinklapis	Klasė	Kliento tinklapio esybė yra HTML pagrindu suformuotas web tinklapis. Kaip ir bet kuris kitas tinklapis, jis yra duomenų, įvykių apdorojimo mechanizmo ir vaizdavimo elementų rinkinys. Šios klasės metodai yra kliento dalies funkcijos, išreikštos tinklapyje, o atributai – tinklapio lygmens kintamieji. Kliento tinklapis gali turėti asociacijų tiek su kliento, tiek ir serverio tinklapiais.			Tinklapių titulas, vaizduojamas naršyklėje. Tinklapių kūno atributai, nustatantys jo turinio stilių.
Forma	Klasė	Formos klasė tiesiogiai atvaizduoja HTML formą. Visi jos atributai yra tinklapių formoje nurodyti informacijos įvesties arba			Metodas, kuris yra aktyvuojamas pasitelkus GET arba POST funkcijas.

		pasirinkimo elementai (teksto įvedimo laukai, pasirinkties laukai ir kt.). Forma neturi operacijų (metodų), kadangi jos lygmens funkcijos aprašomos tinklapio klasės metodais.			
Karkasų aibė	Klasė	Karkasų aibė yra klasė, kurią sudaro tinklapis suformuotas iš kelių dalių. Kiekviena tinklapio dalis atvaizduojama kaip atsiras karkaso elementas, su tikslo asociacija. Ši klasė neturi atributų ir metodų.			Eilutės ir stulpeliai, išskiriami karkasui tinklapio aplinkoje.
Tikslas	Klasė	Tikslas yra esybė, nusakanti kur yra nukreipiamas tinklapio elementas, t.y. tikslas, į kurį nukreipiamas web tinklapis. Jis dažniausiai yra naudojamas su karkasų aibės klasėmis.		Tikslo vardas turi būti unikalus kiekvienam sistemos elementui.	

JavaScript	Klasė	Ši klasė yra JavaScript biblioteka, kuri yra naudojama tinklapyje. Jos operacijos yra bibliotekoje aprašytos funkcijos, o atributai bibliotekos lygmens globalūs atributai,			
Tinklapis	Komponentas	Tinklapių klasė – tai HTML elementas, kurio negalima priskirti nei kliento, nei serverio tinklapiui. Tai web tinklapis, kurio viduje nėra programinio kodo. Dažniausiai tai yra tekstiniai failai, saugomi serveryje.			Kelias, nurodantis tinklapio adresą serveryje.

Lentelė 1. UML profilio metodo elementai

Pasinaudojus visais šiais klasių stereotipais bei asociacijomis tarp jų galima labai detalai ir išsamiai atlikti internetinių taikomųjų programų modeliavimą, neatsižvelgiant į jų struktūros sudėtingumą ir lankstumą. Toks UML profilis interneto svetainėms projektuoti gali būti naudojamas bet kokio tipo web aplikacijoms.

2.3. Semantinio žiniatinklio metodai

Hipertekstas yra tik viena iš technologijų, leidžiančių išplėsti informacijos vaizdavimą iš statiško ir neformuoto teksto su galybe apibojimų į tikslingą ir efektyvų informacijos atvaizdavimą. Hipertekstas naudoja kompiuterio efektus, tokius kaip nuorodos, indeksavimas ir sąveika su vartotojų tam, kad pagerintų komunikaciją tarp žmogaus ir jam reikalingos informacijos srauto. Naudojant šią technologiją yra siekiama, kad duomenys esantys statiniuose HTML failuose, duomenų bazėse būtų

kuo lengviau prieinami ir suprantami žmonėms. Visiškai priešingas tikslas yra keliamas semantinių web sistemų kūrėjams. Tokiose aplikacijose pagrindinis taikynys yra informacijos apdorojimo agentai, t.y. programos, kurios kaip ir žmonės galėtų interpretuoti informaciją, saugomą WWW tinklapiuose. Šios interpretacijos esmė yra ontologija, tai yra pagrindinė ir svarbiausia tokio tipo internetinių taikomųjų programų struktūra, naudojama žinių interpretacijai sistemoje atlikti.

Žinių vadyboje ontologija yra apibrėžiama kaip „konceptualizacijos specifikuojimas“. Galima teigti, jog įprasta ontologija apibrėžia žodyną, kuriam iš agentų (žmonių arba programinės įrangos) yra siunčiamos užklausos ir gaunami patvirtinimai. Ontologija yra pragmatiškas apibrėžimas, žodynas kuris šiuo metu yra naudojamas, konceptai kurie yra naudingi problemos sprendimui [6]. Ontologija yra įrankis kurio kokybė visiškai priklauso nuo jos naudingumo.

W3C konsorciumas nusako ontologiją kaip terminų apibrėžimus, naudojamus atvaizduoti arba nusakyti tam tikrą žinių dalį (dažniau vadinamą aplinka (domain) ir naudojamą žmonių, duomenų bazių bei aplikacijų informacijos kaitos procese. Ontologija tiesiog specifikuota vieną pasaulio supratimo būdą, o skirtingos ontologijos yra naudingos skirtingiems tikslams realizuoti. Yra teigiama, kad gali egzistuoti keletas ontologijų kurios apibūdina tą patį reiškinį visiškai skirtingai, tačiau, tačiau tikslams pasiekti, kuriems jos yra naudojamos, jos yra visiškai tinkamos. Ontologijos panaudojimo web aplikacijose privalumas yra tas, kad reikiama informacija gali būti surasta tuomet, kai kitos sistemos neduos reikiamo rezultato. Pvz. paieškos frazės „baidyklė“ ontologinei sistemai pakanka, kad ji surastu duomenis ir apie „pabaisą“, nors lyginant raides frazės „baidyklė“ ir „pabaisa“ yra visiškai skirtingos. Ontologiją naudojančios sistemos turi privalumą ten, kur informacijos paieška ir panaudojimas įgauna didelę vertę [9].

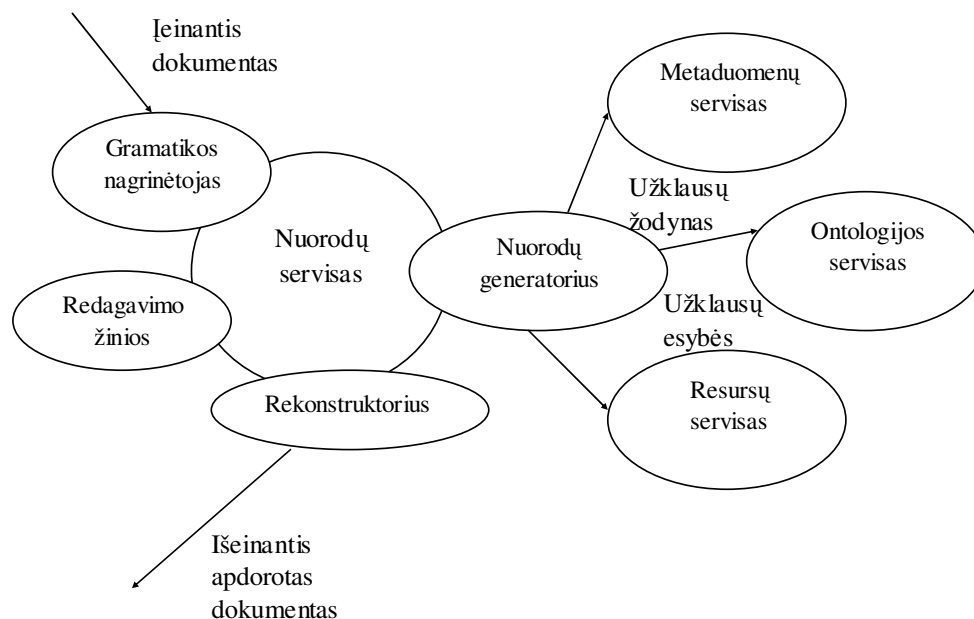
Norint realizuoti tokias sistemas, visai pateikiamai sistemos informacijai yra sukuriamos anotacijos, koncertai, kurių pagalba agentai sugeba susieti formalius teiginius, informacijos sampratą bei turinį ir gauti teisingą rezultatą.

Ontologija yra formalus modelis, kuris leidžia susieti realaus gyvenimo koncertus bei objektus su sudėtingais ryšiais, egzistuojančiais tarp jų. Lengva įsivaizduoti, kad tam yra reikalingos sudėtingos struktūros, leidžiančios apibūdinti ir išaiškinti ryšius tarp hipertekste aprašytų objektų. Įprastinis hiperteksto projektavimas susideda iš numatomos pateikti informacijos analizės, siekiant pateikti ją tinkamu būdu. Visiškai priešingai yra vykdomas ontologija pagrįstų hiperteksto sistemų projektavimas, nes čia sistema išveda savo komponentų struktūrą naudodamasi ryšiais tarp realaus pasaulio objektų.

Yra keletas semantinio žiniatinklio realizacijos metodų. Vienas iš tokių yra COHSE (angl. Conceptual Ontological Hyper Systems Environment), kurio panaudojimo rezultatas yra ontologinė hipermedijos

sistema susiejanti atvirą hipermedijos nuorodų servisą su ontologija grindžiamu servisu, taip sistemoje naudojamus dokumentus susiejant nuorodomis, kurios formuojamos konceptų, susijusių su dokumento turiniu, pagrindu. Pagrindinis COHSE tikslas yra apjungti OHS architektūrą su ontologijos modeliu siekiant suformuoti nuorodų į konceptus, atvaizduojamus tinklapiuose, sistemą.

Ontologijos yra naudojamos aprašyti vidinius ryšius tarp konceptų, egzistuojančių dokumentuose, tam kad sukurti vidinių žinių katalogą. Ontologinio hiperteksto aplinkai reikia tam tikro mechanizmo, kurio pagalba būtų galima interpretuoti ontologiją ir formuoti tuos koncertus ir ryšius tikrame gyvenime kaip nuorodas (arba kitus artifaktus) hipertekste. COHSE naudoja standartinę interneto naršyklę kontroliuojamą adaptuotu nuorodų servisu, kuris paeiliui naudoja tris nepriklausomus servigus, manipuliuojančius tinklapio DOM, taip sudarant ontologijos kontroliuojamą hipertekstą.



Pav. 3 Ontologijos serviso valdymo mechanizmas

Galima matyti (pav. 3), kaip ontologijos servisas valdo ontologijas (aibes konceptų, susijusių pagal tam tikrą schemą) ir atsako į specifines užklaudas, susijusias su šiomis ontologijomis. Ontologijos yra viduje atvaizduojamos pasinaudojant DAML+OIL ir užklaudas yra patenkinamos panaudojant FACT suformuotas priešzastis. Pagrindinė šio serviso užduotis yra atsakyti į fundamentalius klausimus, liečiančius ontologijos konceptus, pvz. „kas yra koncepto tėvas“, „kaip šis konceptas atvaizduojamas natūralioje kalboje“, „ar šie du konceptai yra panašūs arba tapatūs“. Skirtingai nuo kitų semantinių web sistemų, COHSE ontologijos serveris nenaudoja specifinių priklausomybės ryšių tam, kad atsakytu į specifinius ontologijos klausimus. Metaduomenų servisas apdoroja dokumento regijonus konceptualiai ir nekuria dokumento anotacijos naudodamas paprasto teksto gabalus. XRodyklė (angl.

XPointer) yra naudojama identifikuoti kiekvieną dokumento regioną. RDF fragmentas, kuris yra susietas su DAML+OIL teiginiais identifikuoja konceptą. Resursų servisas yra paprasta biblioteka naudojama tinklapių paieškai atitinkančių nurodytą konceptą (panaudojamų konceptui iliustruoti).

Kada web puslapis yra pakrautas, ontologijos servisas perrenka visus kalbos terminus panaudotus atvaizduoti konceptams susijusiose ontologijose. Kiekvieno kalbos termino yra ieškoma dokumente ir, jei randama, jis yra asocijuojamas su ieškomu konceptu. Metaduomenų servisas taipogi yra naudojamas nustatyti ar kartais dokumentas neturi regionų, kurių anotacija buvo atlikta ne automatiškai. Jei dokumente yra indentifikuojamas svarbus konceptas, resursų servisas pateikia sąrašą dokumentų, kurie yra šio koncepto esybės.

CREAM (angl. CREAting Metadata) irgi yra ontologiją naudojantis karkasas (framework) naudojamas metaduomenų ir dokumentų kūrimui. Jis yra paremtas Ont-O-Mat įrankiu, komponentais pagrįstu anotacijos ir autorizavimo sistema kuriama apie apie dokumento redaktoriaus ir ontologijos naršykle. CREAM palaiko semantinio tinklo kūrimą pasitelkiant anotacijas. Anotacija gali būti pasiekta užpildant žinių šablonus, kontroliuojamus ontologinės naršyklės. Kas laibai įdomu, dokumentai gali būti išgaunami atliekant atvirkštinę anotaciją. Esybės, esančios ontologijos žinių bazėje yra panaudojamos atkurti tekstą, kuris išlaiko nuorodas bei ryšius su žinių baze.

Pagrindinis CREAM karkaso tikslas sukurti žinių bazę, kurios gali būti panaudotos kuriant semantines internetines taikomąsias programas.

Ontoportalas yra bendras programų karkasas skirtas kurti ontologija pagrįstus web portalus. Jis parodo kaip semantinis ontologijos konceptų ir ryšių tarp jų meta lygmuo gali būti projektuojami į egzistuojančius resursus web portalo generavimui prasmingai nusakant ir susiejant resursus ir jų ryšius. Šis karkasas suteikia šias galimybes: studijavimas (ontologijos nagrinėjimas), žinių kaupimas (turinio kūrimas ir atnaujinimas), paieškos organizavimas (raktinių žodžių paieška metaduomenyse).

Ontoportalas yra ontologinė hiperteksto sistema, todėl ontologijos yra panaudojamos siekiant patobulinti navigacijos priemonės kurios lemia web portalų kokybę.

2.4. Esamų metodų trūkumai ir siekiamos savybės

Atlikus aukščiau paminėtų metodų analizę galima daryti tam tikras išvadas apie pateiktų metodologijų trūkumus ir privalumus, kurie gali būti panaudoti kuriant naują arba išplečiant jau egzistuojančią metodiką.

Dauguma dalykinės srities modeliais grindžiamų metodų yra skirti modeliuoti tik skaitymui skirtas web svetaines (statines), o tai yra didelis minusas, kadangi numatomas kurti tezaurus yra skirtas tiek duomenų peržiūrai, tiek jų redagavimui. Turinio valdymo galimybės šiuose metoduose yra gan

pasyvus, o tai nėra tinkama kuriamai internetinio tezauro programai. Visi šie modeliai turi lygmeninę aplinkos, kuriai yra kuriama internetinė taikomoji programa, nagrinėjimo sistemą. Taip pradžioje yra kruopščiai išnagrinėjama dalykinė sritis, vėliau sudaromas svetainės turinio bei navigacijos, o taipogi realizacijos planas. Tačiau visi šie modeliavimo lygmenys yra silpnai susiję tarpusavyje, nėra konkretaus sektinumo, kada iš vieno lygmens pereinant į kitą būtų jaučiama aiški siekiamų rezultatų tąsa. Visų šio tipo metodų privalumas yra gera dalykinės srities analizė, tačiau jos rezultatai nėra pilnai išnaudojami, nepakanka tiesiog sudaryti sistemos prototipo klases ar esybes, kurios smarkiai kinta pereinant iš vieno lygmens į kitą.

UML profilis interneto svetainėms projektuoti leidžia išsamiai sudaryti modeliuojamos web programos struktūrą. Šios metadologijos silpnoji pusė būtų silpnas dalykinės srities nagrinėjimas, faktiškai programos modeliavimas prasideda nuo navigavimo sistemos sudarymo. Ne visiškai pilnas ir tinklapių elementų padengimas stereotipais, formos elementai nėra vieninteliai ir pagrindiniai tinklapių elementai, svarbūs projektuojant internetinę sistemą. Didelis šio metodo plusas aiškus serverio dalies bei kliento dalies veiklų atskyrimas, leidžiantis aiškiai išreikšti sistemos valdymo ir realizavimo taisykles.

	Projektavimo procesas	Navigacijos struktūra (mazgai ir navigacijos vienetai)	Interaktyvumas	Modeliavimo technologija
HDM	Detalizavimas „iš viršaus žemyn“. Detalizavimas „iš apačios į viršų“.	Esybės, komponentai, navigacijos vienetai	Tik skaityti	E-R
RMM	E-R projektavimas Lygmenų projektavimas Navigacijos projektavimas Konvertavimo protokolo projektavimas Sąsajos vaizdavimo projektavimas Veikimo elgesio projektavimas Konstravimas ir testavimas	Navigacinės sritys, lygmenys	Tik skaityti	E-R
WebML	Struktūrinio modelio sudarymas Hiperteksto modelis Kompozicijos modelis Navigacijos modelis Vaizdavimo modelis Personalizavimo modelis	Tinklapių, turinio vienetai	Skaityti ir rašyti	E-R / OO
OOHDM	Konceptualus projektavimas	Navigacijos klasės	Tik skaityti	OO

	Navigacijos projektavimas Abstrakčios sąsajos projektavimas Realizavimas			
EORM	Klasių šablonas Kompozicinis šablonas Grafinės sąsajos šablonas	Navigacijos klasės	Tik skaityti	OO
SOHDM	Dalykinės srities analizė OO modeliavimas Pjūvių projektavimas Navigacijos projektavimas Realizacijos projektavimas Konstravimas	Klasės	Tik skaityti	OO
WSDM	Vartotojo modeliavimas Konceptualus modeliavimas Objektų modeliavimas Navigacijos projektavimas Realizavimas	Navigacijos klasės	Tik skaityti	E-R / OO
Onto-Webber	Dalykinės srities ontologija Tinklapio pjūvio modeliavimas, navigavimo modelis, turinio modelis, vaizdavimo modelis Personalizacijos modelis Palaikymo modelis	Tinklapiai, kortos (navigacijos vienetai)	Tik skaityti	Ontologija

Lentelė 2. Metodų apibendrinimas

Semantinio žiniatinklio metodai yra labai efektyvūs kuriant dideles ir sudėtingas žinių kaupimo ir publikavimo sistemas. Internetinio tezauro atveju šie metodai nėra visiškai tinkami, kadangi ontologinių sistemų kūrimas perauga į didelius ir labai sudėtingus mechanizmus, kurie šiam numatomam programiniam produktui vargu ar būtų pateisinami. Šių modelių privalumas būtų efektyvus darbas apdorojant turinį ir realizuojant paieškos sistemas, minusas – realizacijos mechanizmo sudėtingumas.

2.5. Analizės išvados

Išnagrinėjus internetinių svetainių kūrimo metodų galimybes gauta išvada, kad tinkamiausias sprendimas internetinio tezauro realizacija būtų UML profilio interneto svetainėms projektuoti išplėtimas ir pritaikymas specifinėms kuriamos programos reikmėms.

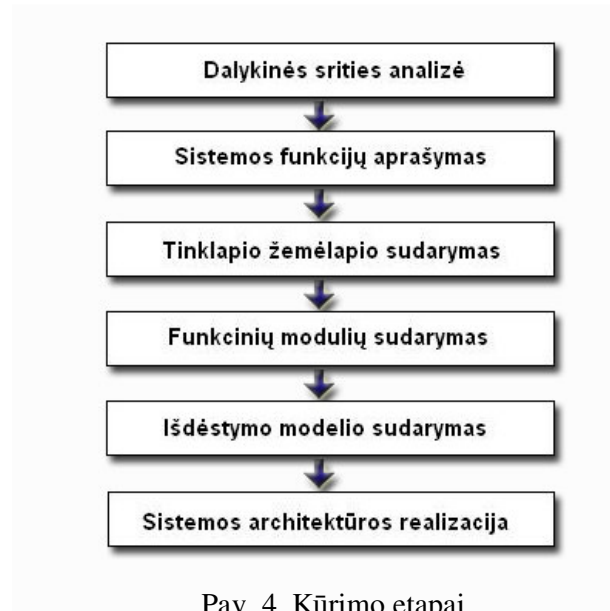
Šis metodas bus išplėstas atliekant išsamią dalykinės srities analizę, kurios rezultatai bus panaudoti veiklos taisyklėms bei prieigos mechanizmo taisyklėms gauti, o taipogi išplėstas navigacijos bei tinklapio žemėlapių modelis, taip sukuriant naujus stereotipus tinklapio elementams išreikšti bei sukuriant galimybę realizuoti daugiakalbystės bei turinio, nepriklausomo nuo jį vaizduojančio objekto.

Analizės metu nagrinėti kiti metodai (lentelė 2) atmesti dėl nepakankamo interaktyvumo kuriamam projektui, sunkiai vaizduojamų perėjimų tarp etapų. Ontologiniai metodai yra per daug sudėtingi ir nebus efektyviai išnaudoti, todėl netinkami kuriamai tezauro programai.

3. SIŪLOMAS TURINIO VALDYMO SISTEMŲ KŪRIMO METODAS

3.1. Kūrimo etapai

Siūlomas metodas turinio valdymo sistemai sukurti susideda iš šešių etapų (pav. XXX). Pirmame etape, dalykinės srities analizės metu, yra kruopščiai išnagrinėjama sritis ir reikalavimai, keliami kuriamai sistemai, o taipogi sudaromos panaudojimo atvejų, klasių, objektų būsenų kaitos, objektų bendradarbiavimo ir panaudojimo atvejų sekų diagramos.



Pav. 4. Kūrimo etapai

Dalykinės srities analizės metu sukauptų duomenų pagrindu yra sudaromas sistemos funkcijų sąrašas, kur yra apibūdinama kiekviena funkcija ir jai priskiriami aktoriai, kurie naudosis šia funkcija. Pagal sistemos funkcijų sąrašą ir panaudojimo atvejų sekų diagramas yra sudaromas tinklapio žemėlapis, kuriame išskiriami visi navigacijos elementai bei objektai, kurie bus atsakingi už sistemos funkcijų realizaciją bei duomenų apdorojimą. Šiame etape yra realizuojama ir sistemos personalizacija pasitelkiant duomenis iš panaudojimo atvejų bei sekų diagramų, kurios yra suprantamos kaip veikimo scenarijai.

Sistemos funkcijos yra apjungiamos į modulius (posistemes) pagal specifiką ir atliekamą darbą. Šios posistemės gali būti aprašytos kaip tinklapio žemėlapyje bei klasių diagramoje išreikštų komponentų junginiai.

Išdėstymo modelis apibendrina sistemos architektūrą bei leidžia aiškiai suskirstyti posistemes į aiškius lygmenis. Sistemos architektūros realizacijos etape yra kuriama programa pagal sudarytą modelį. Siekiant gauti gerą projektavimo rezultatą, kai kurie etapai gali būti detalizuojami jau atlikus paskesnius etapus (pvz. tinklapio žemėlapių korekcijos po išdėstymo modelio sudarymo). Tokiu atveju

yra koreguojami visų tarpe esančių etapų duomenys, atsižvelgiant į žemesnio etapo korekcijas. Tai yra atliekama norint išlaikyti architektūros vientisumą.

Šis metodas yra numatytas taikyti dinaminėms internetinėms programomis, todėl būtina sudaryti duomenų modelį. Duomenų modelis yra kuriamas per kelis etapus. Dalykinės srities analizės dalyje sudaromos klasių diagramos pagrindu yra sukuriamas duomenų modelis, kuris yra koreguojamas (jei iškyla būtinybė) funkcinių modulių sudarymo etape.

Kiekviename etape sukurtų rezultatų (duomenų apie sistemos architektūrą) išsamumas (lentelė 3) įtakoja kuriamos sistemos kokybę.

Etapo pavadinimas	Reikalingi rezultatai
Dalykinės srities analizė (<i>konceptualus lygmuo</i>)	Panaudojimo atvejų diagrama, objektų būsenos kaitos diagramos, klasių diagrama, duomenų bazės schema, panaudojimo atvejų scenarijų diagramos.
Sistemos funkcijų aprašymas (<i>komceptualus lygmuo</i>)	Sistemos funkcijų ir jų vartotojų lentelė
Tinklapio žemėlapių sudarymas (<i>vaizdavimo ir realizacijos lygmuo</i>)	Tinklapio žemėlapių diagrama, stambesnių komponentų detalizacijos, personalizacijos diagramos (atliktos pagal tinklapio žemėlapių diagramą)
Funkcinių modulių sudarymas (<i>realizacijos lygmuo</i>)	Funcinių modelių diagrama, serverio pusės posistemių bei kliento pusės posistemių detalizacijos
Išdėstymo modelio sudarymas (<i>realizacijos ir personalizacijos lygmuo</i>)	Sistemos diegimo, išdėstymo bei paskirstymo diagramos, posistemių diagrama trijų lygmenų (serveris, klientas, dbvs) kontekste
Sistemos architektūros realizacija (<i>realizacijos lygmuo</i>)	Duomenų bazė, sistemos funkcijų apjungtų į modelius realizacija, vartotojo sąsajos realizacija, administratoriaus sąsaja, TVS programinė realizacija

Lentelė 3. Kūrimo etapų rezultatai

3.2. Dalykinės srities modeliavimas

Atliekant dalykinės srities modeliavimą yra sukuriamos panaudojimo atvejų diagrama, klasių diagrama, panaudojimo atvejų sekų diagramos, objektų būsenų kaitos diagramos, duomenų bazės schema (lentelė 4).

Pagal panaudojimo atvejų diagramą yra išskiriamos sistemos vartotojų bei jų atliekamų funkcijų aibės. Vėliau, pasinaudojant šiais duomenimis yra sudaromas sistemos atliekamų funkcijų realizacijos mechanizmas, prieigos kontrolės valdymo mechanizmas ir nustatomos jo taisyklės.

Klasių diagrama šiame etape yra orientuota į duomenis, su kuriais dirba vartotojas ir kurių apdorojimą nori perkelti į sistemą. Klasių diagramos pagrindu yra sudaroma duomenų bazės struktūra bei realizuojamas SQL funkcijų mechanizmas (jei vėliau realizacijai bus naudojama Microsoft SQL serveris). Yra tikslinga sukurti ir pačios sistemos klasių diagramą, kuri vėliau transformuojama į

navigacijos žemėlapiu esybes bei apibendrinant gautas klases gaunamus kuriamos sistemos modulius (posistemes).

Diagramos pavadinimas	Analizės objektas
Panaudojimo atvejų diagrama	Diagramoje yra išreiškiami visi panaudojimo atvejai, kurie numatomi realizuoti sistemoje, bei aktoriai, kurie vykdo išsirtus veiksmus
Klasių diagrama	Klasių diagrama yra sudaroma išreiškiant visus sistemos objektus kaip klases, o ryšius tarp objektų kaip priklausomybes tarp klasių. Klasių diagramoje vaizduojami tiek sąsajos, tiek funkciniai elementai (klasės). Šios diagramos pagrindu yra kuriama tinklapio diagrama.
Panaudojimo atvejų sekų diagramos	Išanalizavus kiekvieną panaudojimo atvejį jiems yra sudaromos seku diagramos, kurios tarnauja kaip veiksmų scenarijai sudarant tinklapio žemėlapi bei personalizacijos struktūrą
Objektų būsenos kaitos diagramos	Išskiriama ir aprašoma būsenų kaita objektams, kurių būseną kinta sistemos veikimo laikotarpiu.
Duomenų bazės schema	Duomenų bazės schema yra išreiškiami kaip klasių diagrama, kur kiekviena lentelė yra atskira klasė, o jos atributai – duomenų įrašų laukų pavadinimai.

Lentelė 4. Diagramų sudarymo apibendrinimas

Panaudojimo atvejų sekų diagramos leidžia nustatyti navigacijos kelius, o taipogi išsiaiškinti sistemoje realizuojamų funkcijų seką. Vėliau šios diagramos gali būti transformuojamos į navigacijos žemėlapiu objektus bei jų pagrindu gali būti koreguojamos prieigos valdymo mechanizmo taisyklės.

Objektų bendradarbiavimo diagramos yra panaudojamos susiejant duomenų bazėse egzistuojančius objektus bei navigacijos žemėlapiu sudarymo metu išskirtas klases bei esybes. Objektų bendradarbiavimo diagramos yra svarbios kartu su būsenų kaitos diagramomis, kurių pagrindu yra sudaroma sistemos būsenų aibė ir sinchronizuojamas veiklos derinimas derinant sistemos architektūrą bei navigacijos žemėlapiu sudaryme gautus duomenis.

3.3. Navigavimo projektavimas

Navigacijos projektavimas yra analogiškas UML profilio interneto svetainėms projektuoti metodui, tačiau yra išskiriami papildomi stereotipai tikslesniam ir detalesniam analizės rezultatui gauti. Skirtingai nei šis metodas yra apibrėžiami ir atskiri tinklapio elementai.

Kiekvienas web puslapis yra suskirstomas į regionus, kurie yra būdingi tam tikros numatomos pateikti arba surinkti informacijos duomenų atžvilgiu. Šiose zonose yra talpinami tinklapio elementai. Regionų stereotipų klasės neturi operacijų, jos atributais yra zoną sudarantys paragrafai arba sub-zonos. Didesnės apimties paragrafai gali būti išreiškiami kaip atskiras regionas. Regionų sudarymas yra atliekamas kartu projektuojant daugybinės vaizduojamųjų elementų paskirties panaudojimą, kada vienas ir tas pats tinklapio elementas informacijos pateikties metu gali būti panaudotas skirtingiems duomenims pavaizduoti, tačiau skirtumas yra tik informacijos pasikeitime, bet ne jos paskirtyje.

Skirtingai nei Conallen metode siūloma išskirti stereotipą „suformuotas tekstas“. Šios klasės atributai yra specifiniai teksto demonstravimo laukai, kuriuose pateikiama informacija, o operacijos – tai metodai, taikomi teksto sričiai aktyvavus tam tikrus vartotojo pusės įvykius. Toks elementas savo svarba nenusileidžia duomenų įvedimo elementams (formos elementams), nes jo būtinumas daugiakalbystės projektavimo atveju yra kritiškas.

Kaip atskirą nuorodų klasių stereotipą yra siūloma išskirti sisteminį meniu. Šio tipo klasėje kaip atributai būtų pateikiami meniu elementai, kaip operacijos išreiškiamos jų apdorojimo bei reakcijos į tam tikrus sistemos būsenų pasikeitimus funkcijos.

Sistemos modelyje pateikiamus tinklapio pateikties elementus (formos elementus) yra siūloma išivaizduoti ne kaip standartines HTML žymes, o kaip tam tikrus JavaScript priemonėmis jau apdorotus objektus, tarsi pusfabrikačius tolesniam jų paskirties rezultatui gauti. Pvz. paprastą įvesties elementą sudaro ne tik duomenų įvedimo `<input>` žymė, bet kartu prie jos konstruojama aiškinamoji konstantą (įvedimo lauko pavadinimas), duomenų pirminio patikrinimo ir apdorojimo funkcija, pagalbos sistemos (jei tokia sistema yra diegiama) indeksas, vaizdavimo stiliaus taikymo taisyklės. Tokie elementai gali būti paruošti vieno iš projekto realizacijos metu, o po to pakartotinai panaudojami kituose analogiškuose projekto realizacijose. Toks HTML žymių perrašymas gali būti labai naudingas sistemos realizacijos stadijoje, kada pateikties kodas yra smarkiai optimizuojamas, o tai automatiškai įtakoja sistemos veikimo greitį ir efektyvumą. Visi šie papildomi elementai yra pateikiami sekančiame pateikties elementų skyrelyje.

3.4. Pateikties elementų tipai

Išplečiant UML profilio svetainėms projektuoti stereotipų aibę yra įvedami nauji stereotipai – klasės, kurie padeda išsamiau projektuoti tinklapio žemėlapi ir vartotojo sąsają.

Menui klasė – tai elementas, kurio pagrindu yra sudaromas aktyvaus tinklapio meniu. Jis yra susietas ryšiais su regionu, kadangi privalo priklausyti tinklapio meniu regionui. Jam gali būti nurodomos koordinatės vaizdavimo lango lygmenyje, aktyvavimo, deaktyvavimo bei pelės veiksmų apdorojimo funkcijos, o taipogi stilių palaikymo sąlygos bei matomumo tam tikrais sistemos būsenos atvejais taisyklės.

Suformuoto teksto klasė – tai pateikiamos tekstinės informacijos elementas, kurio turinys yra apdorojamas norint gauti suformatuotą teksto lauką. Jam yra taikomi stiliai, siekiant pagerinti vizualizacijos efektą.

Įvedimo lauko klasė – tai elementas, kuris priklauso formai ir yra skirtas duomenų surinkimui. Skirtumas nuo HTML žymių yra tas, kad šis vienas elementas apima visus informacijos surinkimo

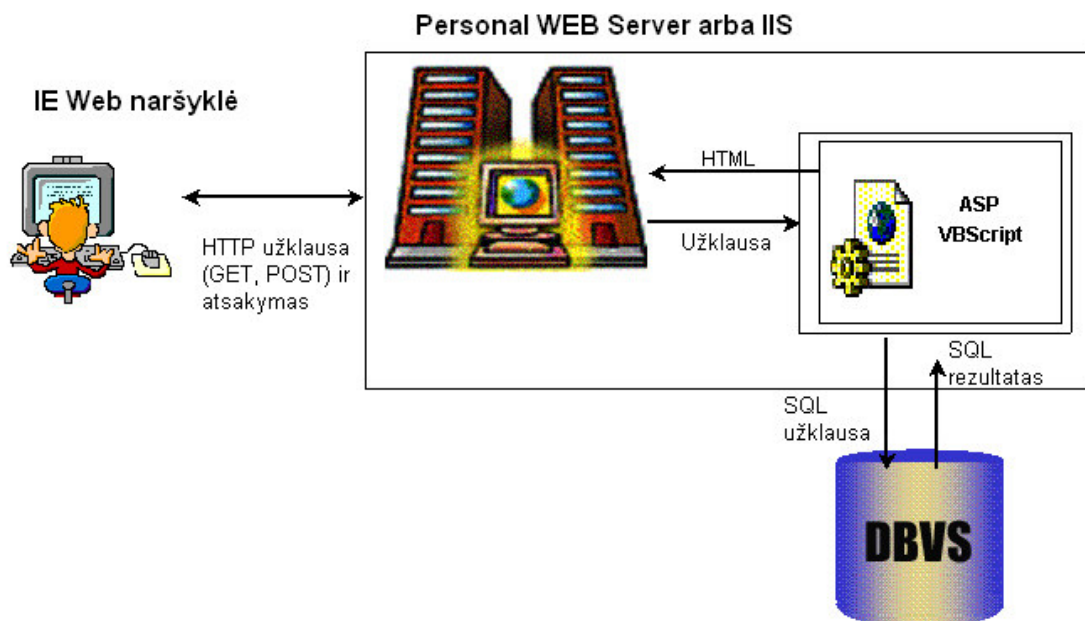
laukų tipus. Jam taikomos pelės veiksmų bei papildomų su juo susijusių įvykių apdorojimo taisyklės ir stilių pritaikymas, o taipogi matomumo sąlygos esant tam tikrom sistemoms būsenoms.

Regiono klasė – tai vidines dalis apibendrinantis elementas, kuriam yra taikomi stiliai ir matomumo, priklausančio sistemoms būsenoms taisyklės.

Paveikslėlio klasė – tai vaizdinės informacijos pateikimo elementas, kurio pagalba yra pateikiama grafinė medžiaga. Jam kartu yra taikomas stilius, nurodomas jo dydis bei funkcijos, reaguojančios į įvykius jo regione.

3.5. Svetainės architektūros projektavimas

Šio metodo siūloma tipinė svetainės architektūra – trijų lygmenų (pav. 5), vartotojas-serveris-duomenų bazė. Pirmajame ir pačiame žemiausiame (duomenų bazės) lygmenyje yra realizuojamos duomenų lentelės bei SQL funkcijos, kurių pagalba vidurinis lygmuo vykdo reikalingų duomenų apskaitą. SQL funkcijos pagreitina duomenų išgavimą iš bazės, o duomenų bazės atskyrimas nuo serverio lygmens yra naudingas, kada norima išgauti didesnę sistemą saugumą. Šis lygmuo ir jo elementai yra projektuojami atlikus dalykinės srities analizę ir išsiaiškinus, kokio tipo duomenys bus naudojami sistemoje.



Pav. 5 Trijų lygmenų internetinių programų architektūra

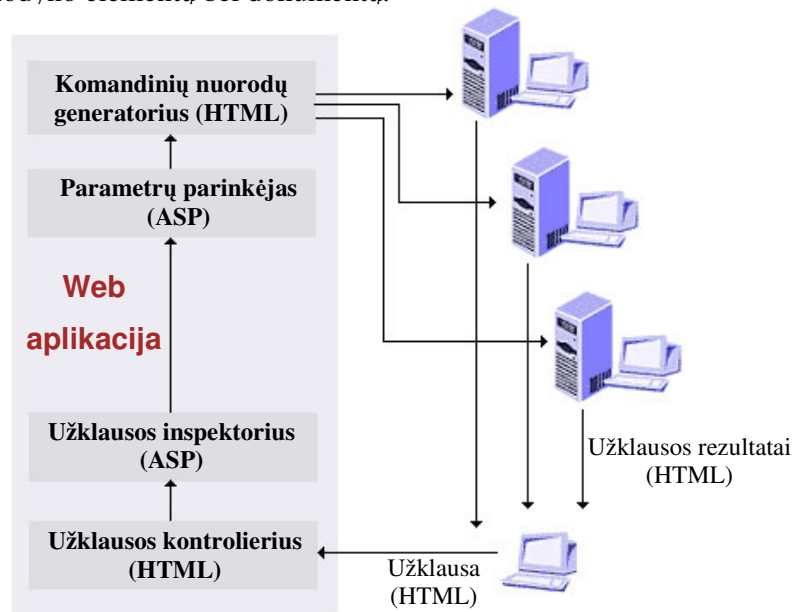
Viduriniajame (serverio) lygmenyje yra išdėstoma visa sistemos funkcionalumo realizacijos dalis, taipogi vartotojų teisių valdymo bei sistemos būsenos kontrolės posistemės. Šio lygmens rezultatas –

virtotojui pateikiami sistemos resursai pagal jo suformuotas užklausas. Visas lygmuo gali būti modeliuojamas atlikus dalykinės srities analizę ir išskyrus funkcijų grupes, kurios bus realizuotos sistemoje, virtotojus, kurie naudosis sistema bei prieigos bei sistemos būsenų sudarymo taisykles.

Aukščiausiam (arčiausiai virtotojo esančiam) lygmenyje yra realizuotas sąsajos valdymo mechanizmas, leidžiantis rinkti informaciją iš virtotojo bei ją pateikti, reaguoti į sistemos būsenos pokyčius, suformuotos viduriniajame lygmenyje. Visa kuriamos sistemos sąsaja yra modeliuojama navigacinio sistemos žemėlapiu sudarymo metu. Jo realizacijos etape yra suformuojami kanalai informacijai apdoroti ir perduoti tarp viduriniojo ir aukščiausiojo lygmens, taipogi išskiriami stiliai ir valdikliai, kurie bus naudojami sistemoje, taip pat kaip ir nuorodų sistemos, meniu elementai bei pateikties elementai, suskirstyti į zonas.

3.6. Paieška

Kuriamai sistemai siūloma realizuoti dvigubą paieškos mechanizmą – lokalią ir globalią. Lokalų paieškos mechanizmą yra naudojamas duomenims ir dokumentams esantiems aktyvioje sistemoje gauti. Siūlomame sistemos projekte paieškos mechanizmas nesinaudoja dokumentų indeksavimu, o kiekvienam sistemos žodyno elementui bei dokumento egzemplioriui leidžia nurodyti praktiškai neribotą kiekį metažodžių, pagal kuriuos ir yra atliekama duomenų paieška bei susiejimas tarp sistemoje naudojamo žodyno elementų bei dokumentų.



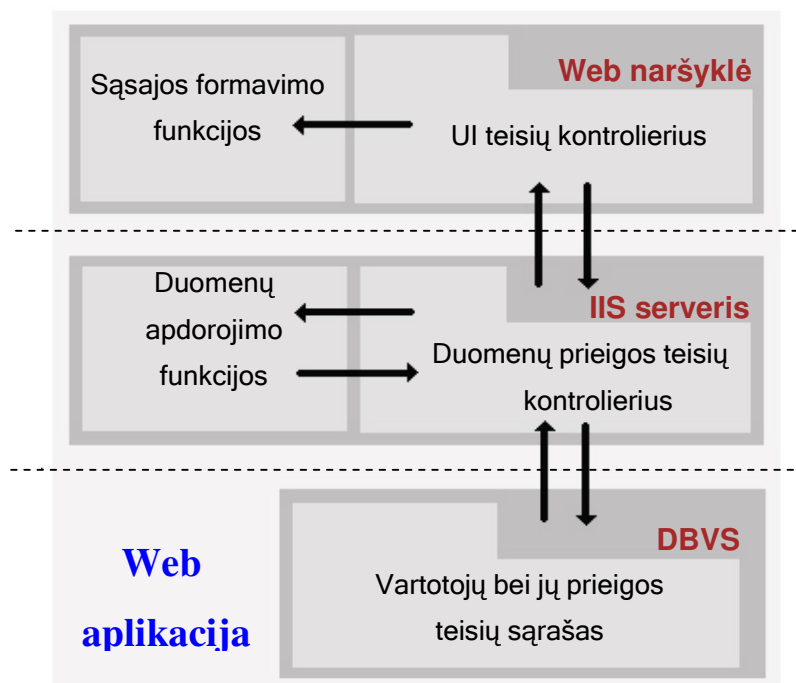
Pav. 6. Globalios paieškos sistemos mechanizmas

Globali paieška yra realizuojama per specifinius parametrus panaudojant išorėje esančių sistemų paieškos mechanizmus bei variklius (pav. 6), kada virtotojo užklausa yra struktūrizuojama,

suformuojami komandiniai parametrai ir aktyvuojama paieškos funkcija. Sistemos struktūra leidžia atlikti konfigūracijos nustatymus praktiškai neribotam skaičiui paieškos sistemų, kurioms pagal nurodytus parametrus yra siunčiamos užklausos ir gaunami atitinkami rezultatai. Šie užklausų formavimo parametrai ir taisyklės yra formuojami sistemos administravimo dalyje.

3.7. Vartotojų teisių valdymas

Vartotojų teisių valdymas yra atliekamas atskiro sistemos modulio, kuris sąveikauja su visais kitais programos elementais (pav. 7). Vartotojų teisės yra sukuriamos jam registruojantis sistemoje, ir jas keisti gali tik sistemos administratorius. Visos vartotojų prieigos taisyklės turi būti sudarytos atlikus dalykinės srities analizę. Šios taisyklės yra realizuojamos serverio pusės tinklapiu elementuose, kurie su vartotojo pusės sistema sąveikauja specialiai tam sukurtais pranešimų sistemos kanalais.



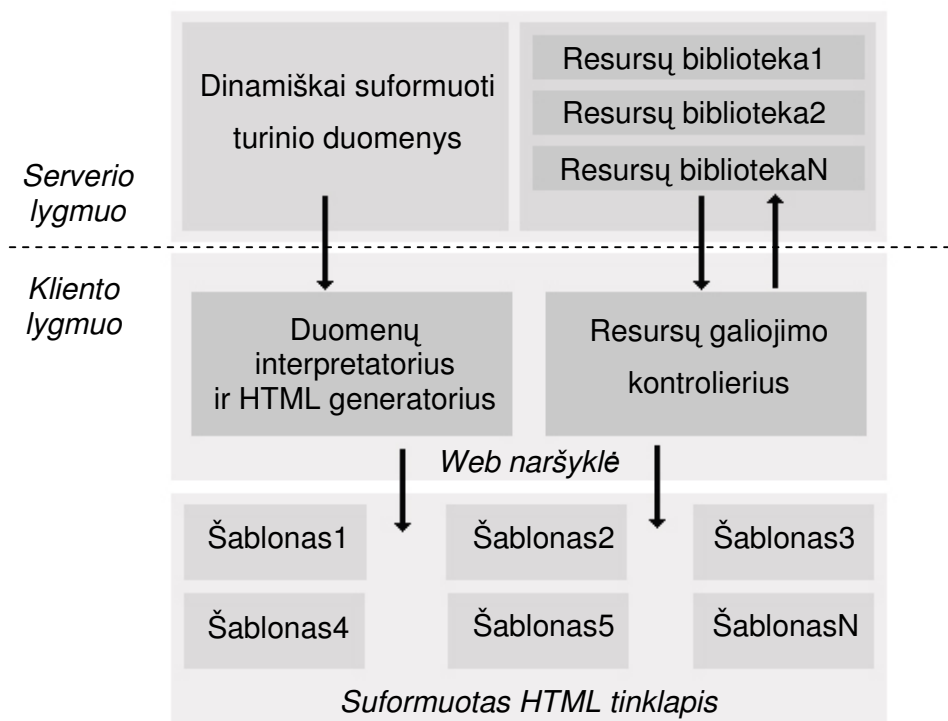
Pav. 7. Vartotojų teisių valdymo mechanizmas

Vartotojų teisių patikra yra atliekama kiekviena kartą atliekant duomenų užklausos veiksmą, tai yra kreipiantis į sistemą nekeičijamiems resursams gauti. Sistemos vartotojų teisių patikros mechanizmas yra realizuojamas interpretuojant sistemos elgesio taisykles, būsenų kaitas bei sistemoje išskirtų aktorių teises į tam tikrų resursų panaudojimą.

3.8. Daugiakalbystės realizavimas

Daugiakalbystė yra realizuojama atskiriant vaizduojamuosius tinklapių objektus nuo jų turinio ir visą sistemos informaciją patalpinus duomenų bazėje kaip konstantas. Šios konstantos nėra paprastas tekstas, yra naudojami specifiniai simboliai kurie vėliau yra interpretuojami kaip teksto formavimo komandos. Konstantų viduje gali būti ir HTML žymių elementai. Tokiu būdu yra išsprendžiamos kelios problemos: skirtingai nei kitose sistemose nedubliuojami dizaino elementai, visa sistemos struktūra ir jos panaudojimas nekinta nepriklausomai nuo pasirinktos kalbos, taipogi toks funkcionalumas leidžia lengvai koreguoti sistemos pateikiamą informacinį turinį ir iš esmės neriboją sistemoje galimų panaudoti kalbų skaičiaus.

Konstantos yra talpinamos į tinklapių žemėlapių dalyje suprojektuotus objektus – šablonus (pav. 8), kurie išdėstomi tinklapių viduje pagal suplanuotą sąsajos vaizdą.



Pav. 8. Daugiakalbystės ir TVS realizacija

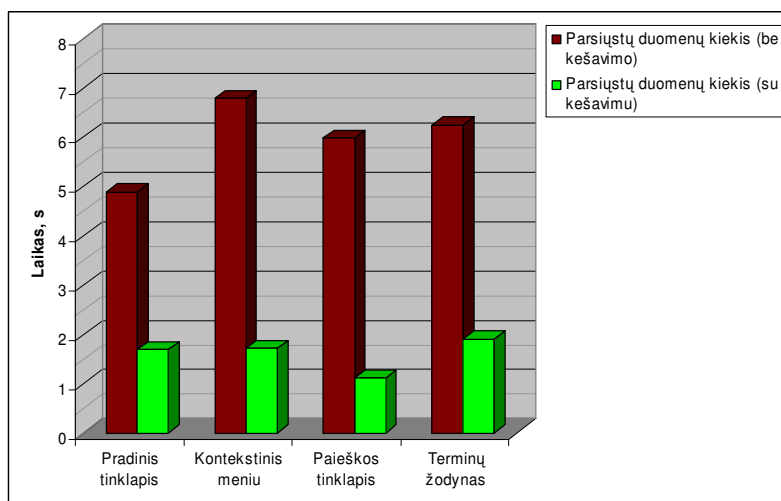
Kadangi visi vaizduojamieji sistemos elementai (o tiksliau jų turinys) yra laikomi bazėje kaip konstantos, kiekvienam tinklapiui suformuoti yra išleidžiama daug brangaus resurso – laiko. Šiai problemai pašalinti yra panaudojamas sparčiosios atmintinės ir sąsajos elementų paskirstymo pagal priklausomybę turiniui mechanizmas. Kiekviena konstanta turi specifinį prefiksą, kuris nusako jos priklausomybę kalbai bei specifiniam programos elementui. Tokios konstantos yra grupuojamos pagal priklausomybę kalbai bei konkrečiai elementų aibei bei formuojamos sąsajos resursų bibliotekos,

kurios patalpinamos sistemoje. Kiekvieną kartą vartotojui užkrovus tinklapį bus naudojami tik tie resursai, kurie jam tuo metu yra reikalingi, taip iki 4 kartų sumažinant laiko sąnaudas (lentelė 5).

Lango pavadinimas	Bandymų skaičius	Parsiūtų duomenų kiekis (be kešavimo)	Laikas (be kešavimo)	Parsiūtų duomenų kiekis (su kešavimu)	Laikas (su kešavimu)
Pradinis tinklapis	50	97KB	4,89s	33KB	1,69s
Kontekstinis meniu	50	212KB	6,78s	64KB	1,72s
Paieškos tinklapis	50	115KB	5,98s	21KB	1,12s
Terminų žodynas	50	168KB	6,25s	80KB	1,91s

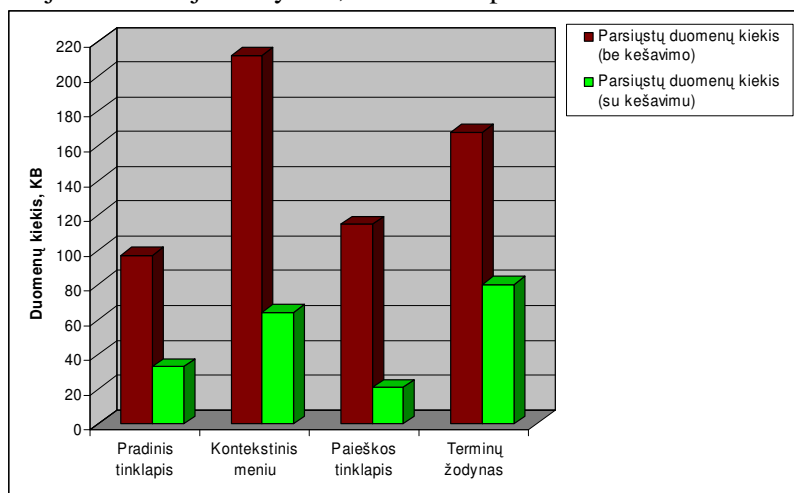
Lentelė 5. Resursų kešavimo efektyvumo testavimas

Kadangi dalis tinklapio vizualizacijos elementų yra formuojami nebe dinamiškai (jų turinio pasikeitimai yra atliekami atnaujinant resursų bibliotekas, tai turi atlikti sistemų administratorius), tai



Pav. 9. Tinklapio sukūrimo greičių palyginimas

šie resursai yra kešuojami vartotojo naršyklės, tokiu būdu padidinant sistemos vaikimo greitį daugiau



Pav. 10. Siunčiamų duomenų kiekio palyginimas

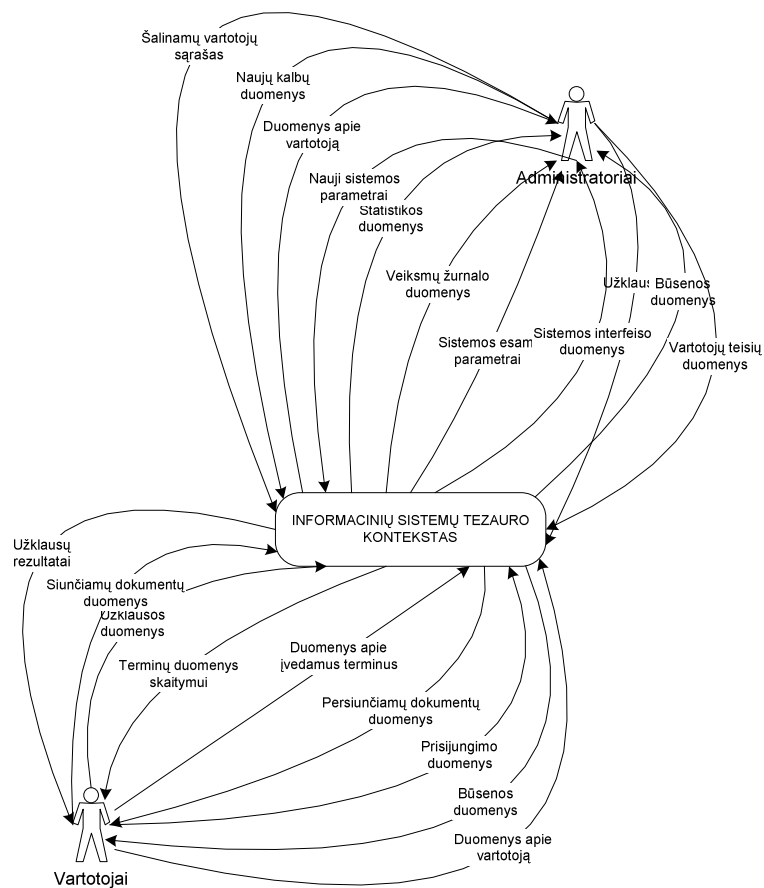
nei vieną kartą kreipiantis į tą pačią programos sritį iki 3 kartų (pav.9 ir pav.10).

Taigi toks sparčiosios atminties ir dinaminio išrinkimo sumažinimo mechanizmų panaudojimas padeda sistemos darbą paversti maksimaliai efektyviu.

4. INFORMACINIŲ SISTEMŲ TEZAURO MODELIS

Žemiau pateiktos diagramos ir modeliai, kurie buvo sukurti siekiant atlikti Informacinių sistemų tezauro dalykinės srities analizę ir išskirti sistemos naudotojus, pagrindines funkcijų grupes, sistemos valdymo mechanizmą bei prieigos prie sistemos funkcionalumo ribojimo taisykles. Šie modeliai bei diagramos yra išskiriami atlikus dalykinės srities analizės, funkcijų aprašymo bei tinklapio žemėlapio sudarymo etapus. Likę trys metodo etapai daugiau siejasi su realizacijos lygmeniu ir yra aprašyti sekančiame skyriuje.

4.1. Veiklos kontekstas

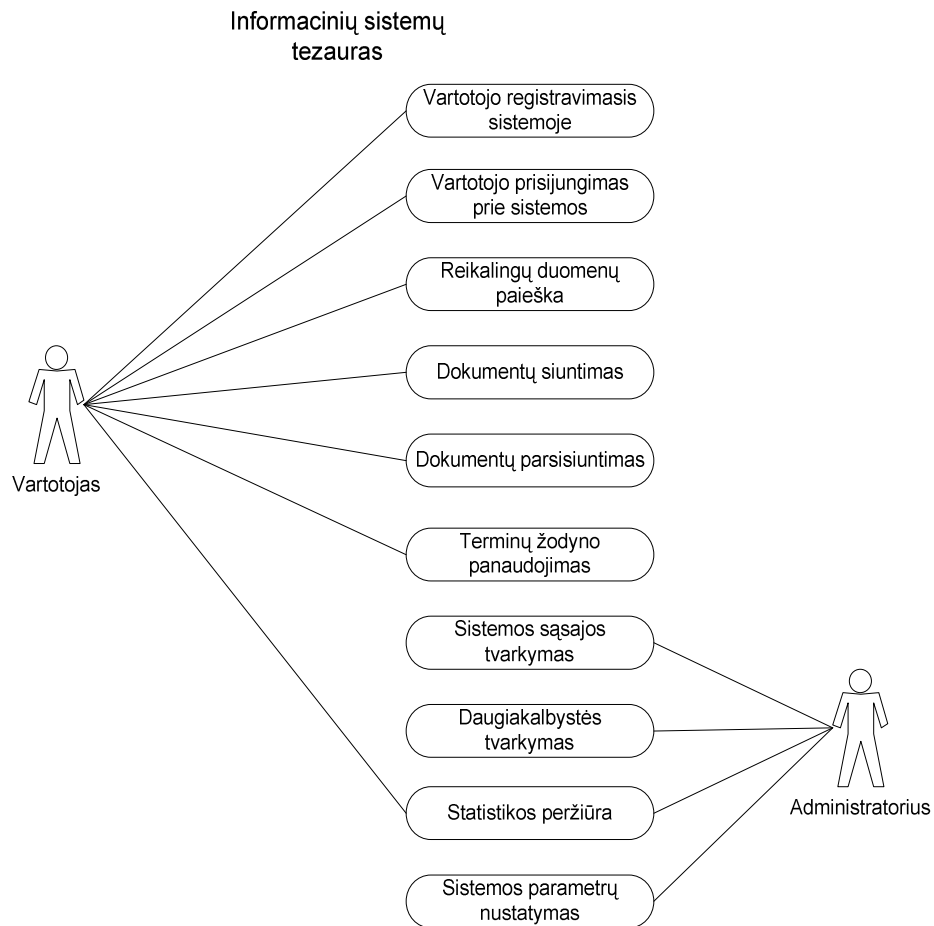


Pav. 11. Veiklos konteksto pavaizdavimas

Analizuojant dalykinę sritį svarbu išsiaiškinti modeliuojamos veiklos kontekstą (pav. 11). Informacinių sistemų tezauro modeliavimo atveju taip yra išsiaiškinami duomenų srautai, kurie turės būti realizuoti sistemoje, vartotojai bei jų atliekamos funkcijos. Siekiant detalizuoti šias žinias yra sudaromas panaudojimo atvejų modelis.

4.2. Panaudojimo atvejų modelis

Panaudojimo atvejų modelio diagrama (pav. 12) detalizuoja veiklos konteksto analizės metu gautas žinias. Sudaryti panaudojimo atvejai bus panaudoti sekančiame etape sudarant funkcijų sąrašą bei jas susiejant su vartotojais. Panaudojimo atvejų modelyje pateikti tik du vartotojų tipai, tačiau pasinaudojant panaudojimo atvejų sekų diagramomis yra išskiriami papildomi vartotojų tipai, kurie aprašomi objektų būsenos kaitos diagramose.



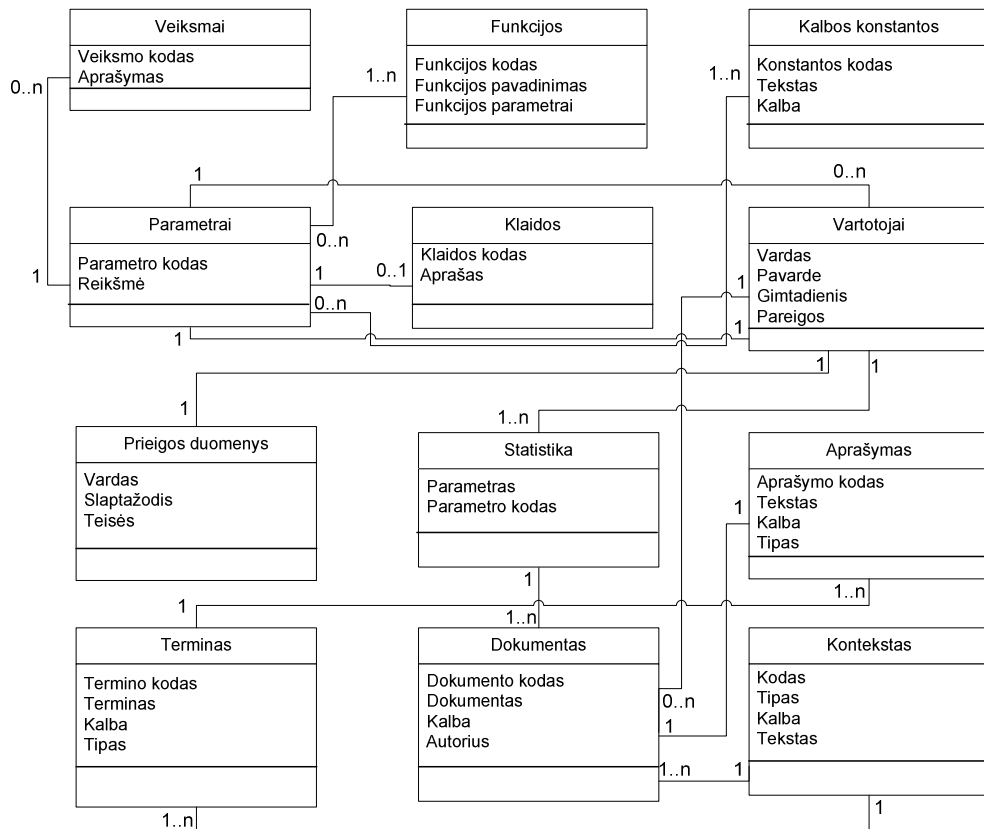
Pav. 12. Panaudojimo atvejų modelis

Kai kurios funkcijos panaudojimo atvejų modelio konvertavimo į funkcijų sąrašą metu gali būti smulkinamos jas detalizuojant arba dėl vartotojų prieigos teisių (saugumo) funkcionalumų.

4.3. Klasių diagramos

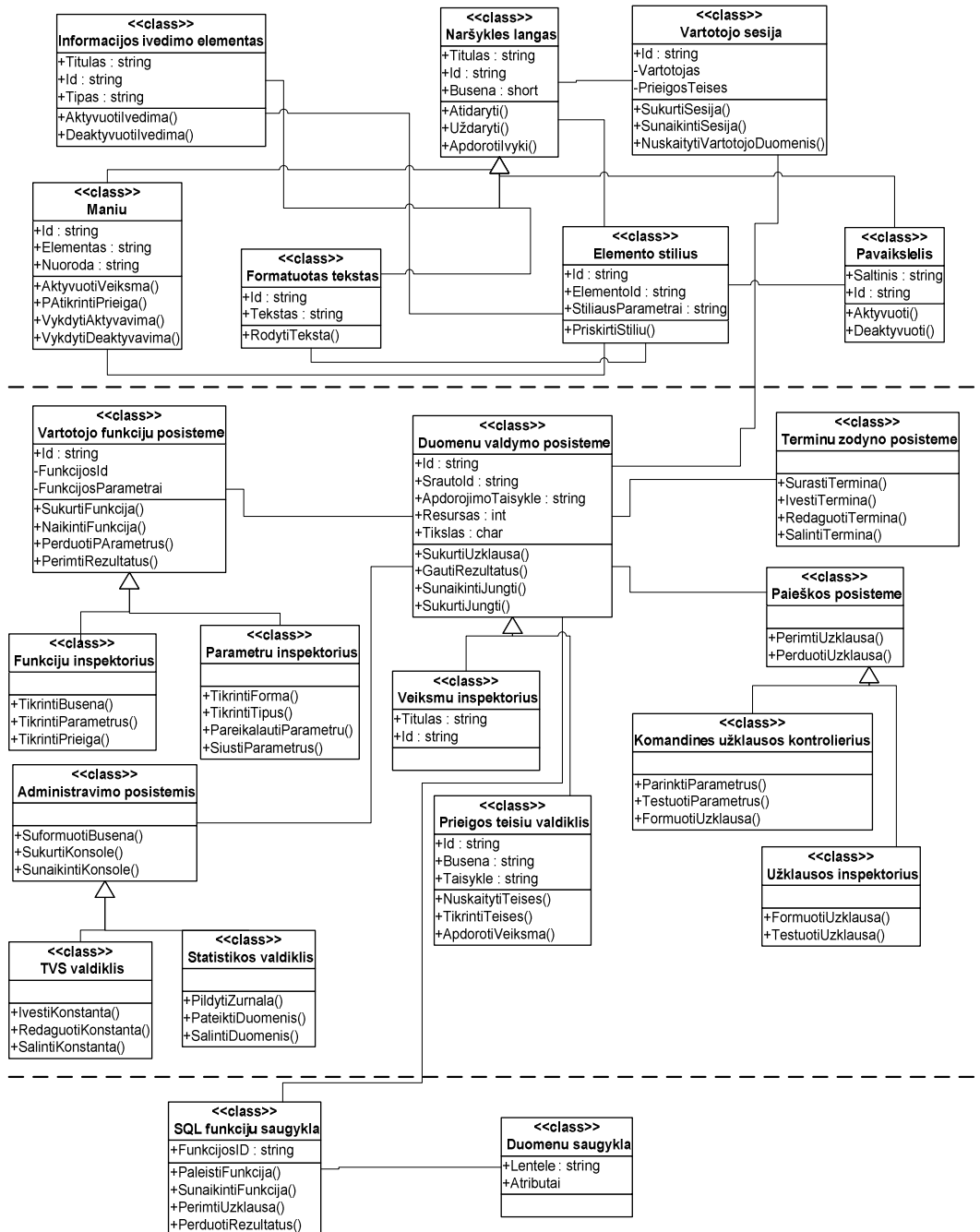
Nagrinėjant sistemos struktūrą ir jos naudojamų duomenų struktūrą išsiskiria pagrindiniai projektavimo tikslai, todėl siekiant abi užduotis atlikti teisingai ir atlikti kuo išsamesnę analizę yra sudaromos dvi klasių diagramos. Pirmoji klasių diagrama (pav.14) yra visos sistemos struktūrinė klasių diagrama. Joje dalyvaujantys objektai aprašo tiek serverio, tiek ir kliento klases.

Antroji klasių diagrama (pav. 13) yra skirta tik duomenų bazės realizavimui, čia pagrindinis dėmesys yra skiriamas duomenų objektams ir ši diagrama gali būti transformuojama į DB, kur kiekvieną lentelę atitinka diagramos klasė, o įrašų laukus – klasės atributai. Įrašų duomenų tipai bus tikslinami sekančiuose etapuose (realizacijos lygmenyje), tuomet gali būti nustatomi įrašų tipai ir koreguojama pati duomenų klasės struktūra.



Pav. 13. Sistemos naudojamais duomenimis pagrįsta klasių diagrama

Aukščiau pateikta dalykinės srities klasių diagrama, pagal kurią sukurtas duomenų bazės modelis. Šias klases naudoja visi funkciniai posistemiai.



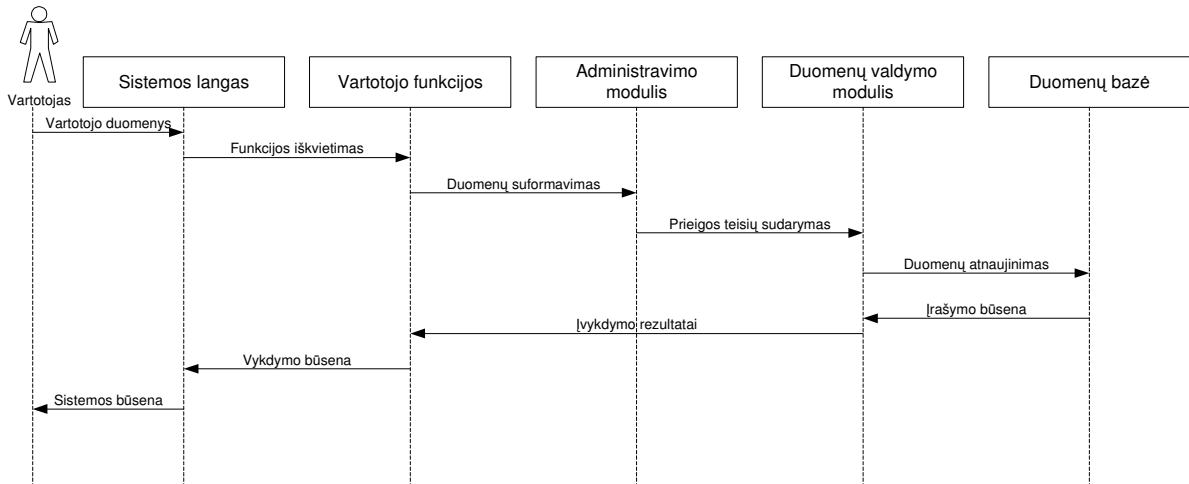
Pav. 14. Sistemos klasių diagrama

Pateiktoje klasių diagramoje išskiriami trijų lygmenų objektai (kliento, serverio, dbvs), atskirti punktyrine linija.

4.4. Panaudojimo atvejų sekų diagramos

Panaudojimo atvejų sekų diagramos (pav. 15, 16, 17, 18, 19, 20, 21, 22, 23, 24) yra sukuriamos kiekvienam panaudojimo atvejui sukurtam panaudojimo atvejui. Kiekviena tokia diagrama yra suprantama kaip funkcijos vykdymo scenarijus, pagal kurį bus formuojama sąsaja ir tinklapio žemėlapis, o taip pat atsižvelgiama realizuojant sistemos funkcionalumą.

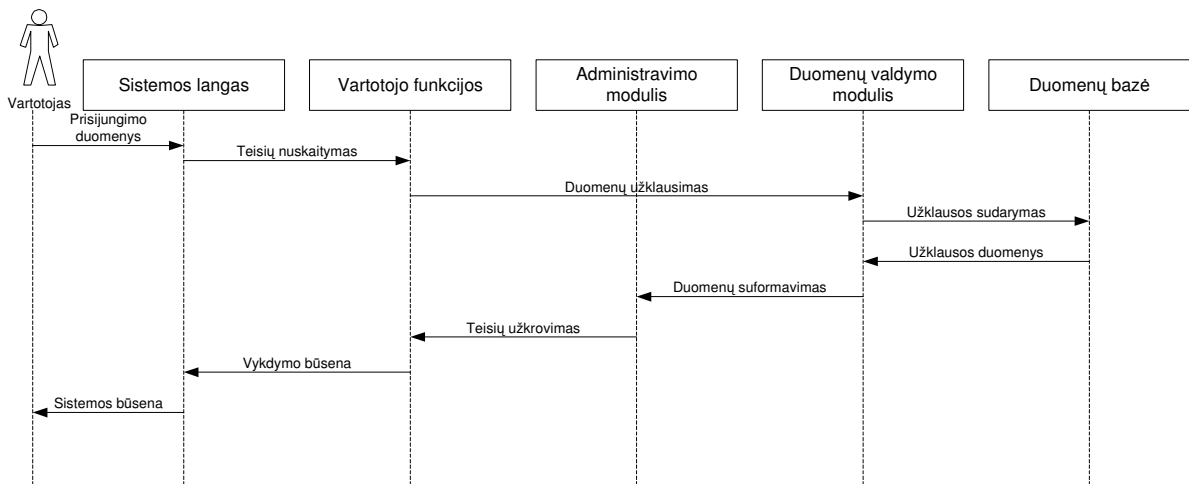
4.4.1. Vartotojo registravimasis sistemoje



Pav. 15. Vartotojo registravimasis sistemoje

Vartotojas registruodamasis sistemoje (pav. 15) iššaukia sistemos langą, kur surinkti duomenys yra apdorojami serverio lygmenyje ir galų gale įrašomi į sistemos duomenų bazę. Sistemos būseną perduodama priešinga kryptimi, vartotoją ji pasiekia pranešimo apie įvykdyto veiksmo būseną pavidalu.

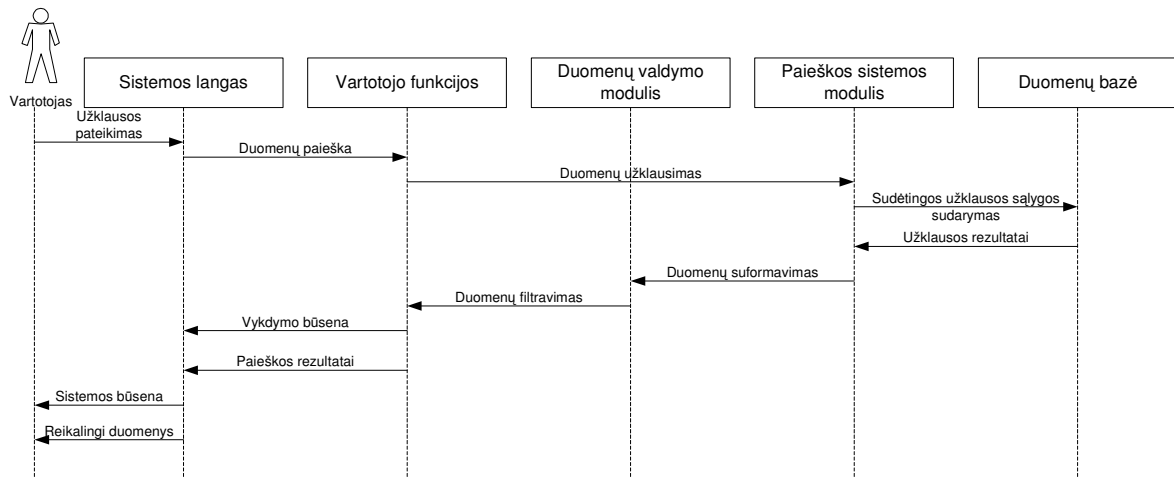
4.4.2. Vartotojo prisijungimas prie sistemos



Pav. 16. Vartotojo prisijungimas prie sistemos

Vartotojas prisijungdamas prie sistemos (pav. 16) iššaukia sistemos langą, kur surinkti prisijungimo duomenys yra apdorojami serverio lygmenyje ir galų gale įrašomi į sistemos duomenų bazę. Sistemos būseną perduodama priešinga kryptimi, vartotoją ji pasiekia pranešimo apie įvykdyto veiksmo būseną pavidalu. Sistemos būsenos inspektorius nustato sesijos duomenis – vartotojo prieigos teises.

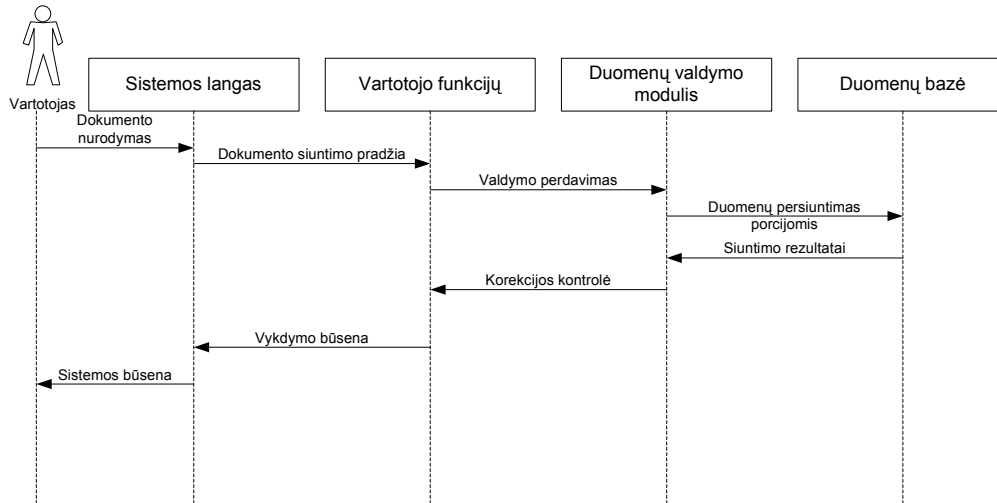
4.4.3. Reikalingų duomenų paieška



Pav. 17. Reikalingų duomenų paieška

Reikalingų duomenų paieškos scenarijus (pav. 17) vaizduoja tik lokalių duomenų paiešką, tačiau analogiškai yra vykdomas ir išorinių duomenų paieškos procesas. Duomenų bazės suformuoti parametrai yra perduodami paieškos komandų inspektoriui, kuris aktyvuoja išorinį duomenų paieškos mechanizmą. Paieškos rezultatai atitinkantys vartotojo pateiktą užklausą yra gražinami į sistemos langą kartu su pakitusia būseną (pranešimu apie veiksmo sėkmingumą).

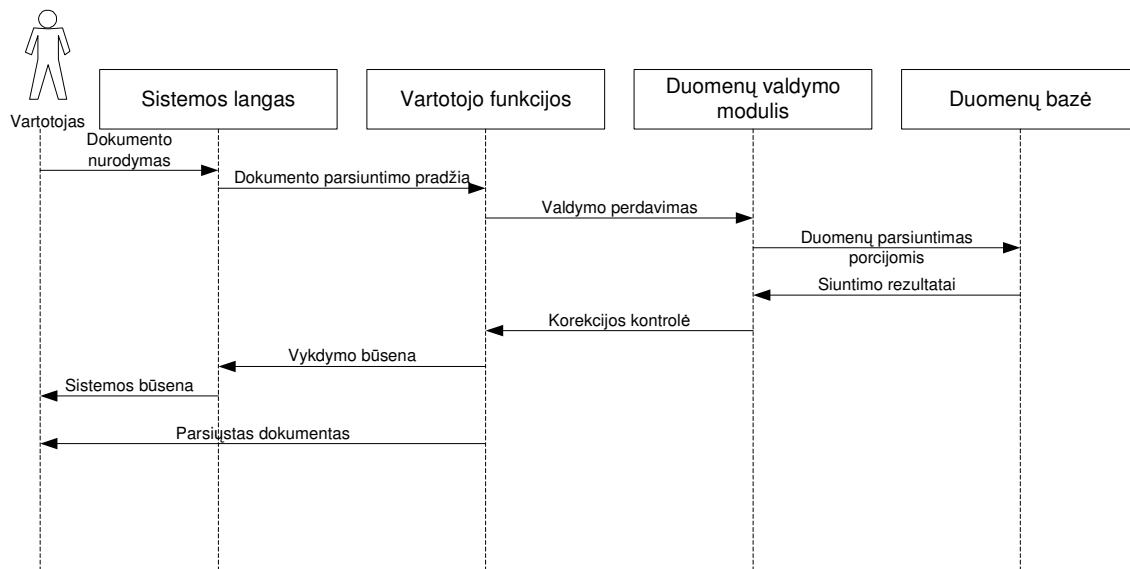
4.4.4. Dokumentų siuntimas



Pav. 18. Dokumentų siuntimo procesas

Dokumento siuntimas sistemoje (pav.18) yra sudėtingesnis, nes be aukščiau išvardintų duomenų apdorojimo procesų pats dokumentas yra talpinamas į bazę ir sukuriama veiksmų žurnalo registracija (tai atlieka duomenų valdymo modulis). Patalpinus dokumentą kiekvieną jo duomenų redagavimo veiksmą (kurio scenarijus sutampa su dokumento talpinimo scenarijumi) į žurnalą įveda duomenų valdymo posistemė (modulis).

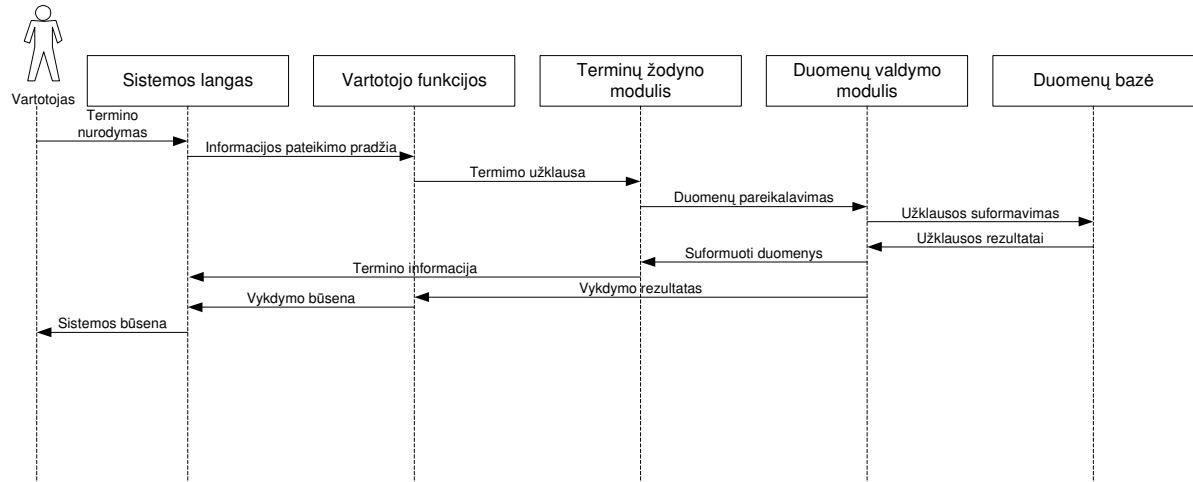
4.4.5. Dokumentų parsisiuntimas



Pav. 19. Dokumentų parsisiuntimo procesas

Dokumento parsisiuntimas iš sistemos (pav. 19) galimas tik radus reikalingo dokumento nuorodą bei prieš tai prisijungus prie sistemos (šie veiksmai buvo aprašyti anksčiau). Šio veiksmo išskirtinumas yra duomenų siuntimo korekcijos kontrolė, kuri tikrina, ar duomenys buvo pilnai parsiųsti iš sistemos.

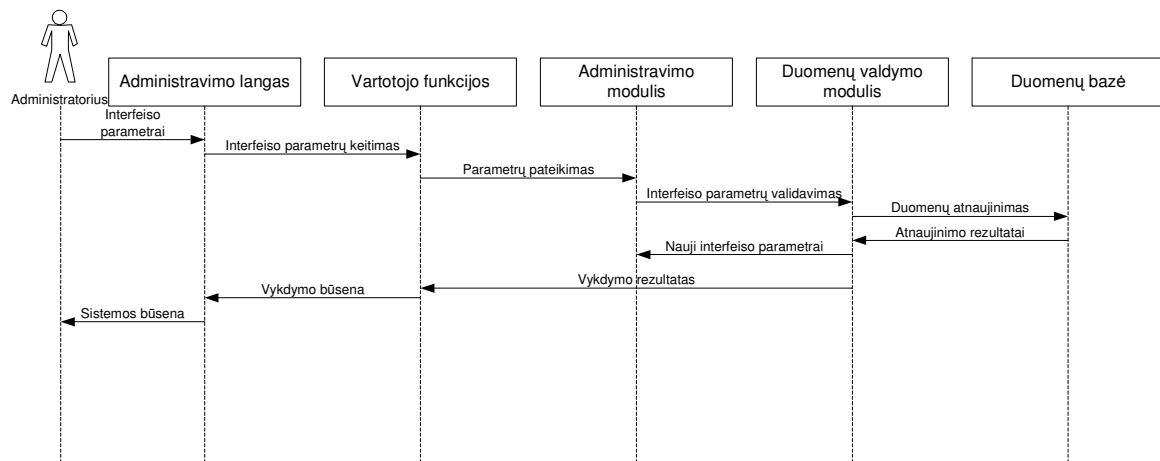
4.4.6. Terminų žodyno panaudojimas



Pav. 20. Terminų žodyno panaudojimas

Atliekant terminų žodyno panaudojimą (pav. 20), reikia būti prisijungus prie sistemos (šis scenarijus aprašytas anksčiau). Terminų žodyno modulis (posistemė) yra atsakingas už teisingą terminų reikšmių atvaizdavimą ir pasirinkimą. Duomenų apdorojimą vykdo duomenų valdymo posistemė, funkcijos rezultatą formuoja vartotojo funkcijų posistemė.

4.4.7. Sistemos sąsajos tvarkymas

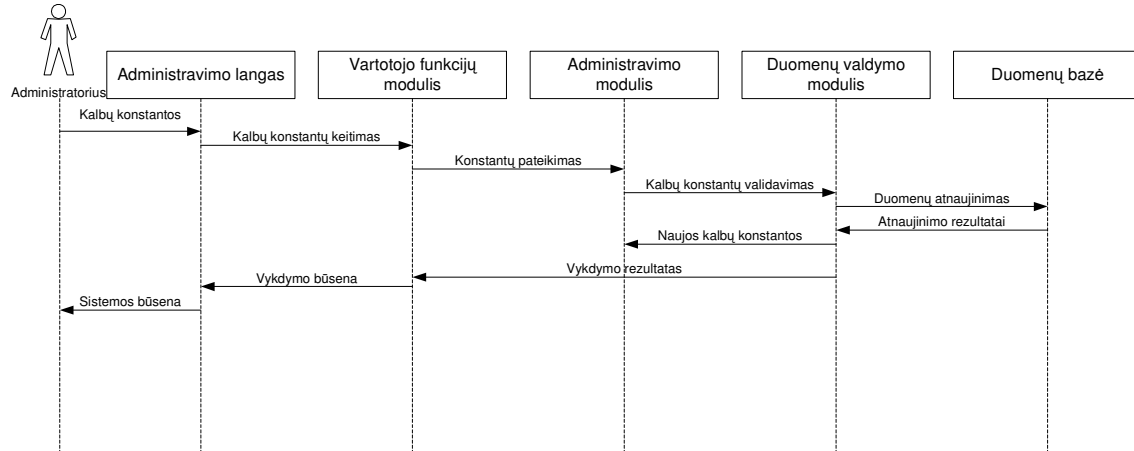


Pav. 21. Sistemos sąsajos tvarkymas

Sistemos sąsajos tvarkymas (pav. 21) yra vykdomas prisijungus prie sistemos ir perėjus į administravimo konsolę. Tai atlikti gali tik administratoriaus teises turintis sistemos vartotojas. Jų

neturintis asmuo negalės aktyvuoti administravimo posistemio funkcionalumą, susijusių su turinio konstantų valdymu.

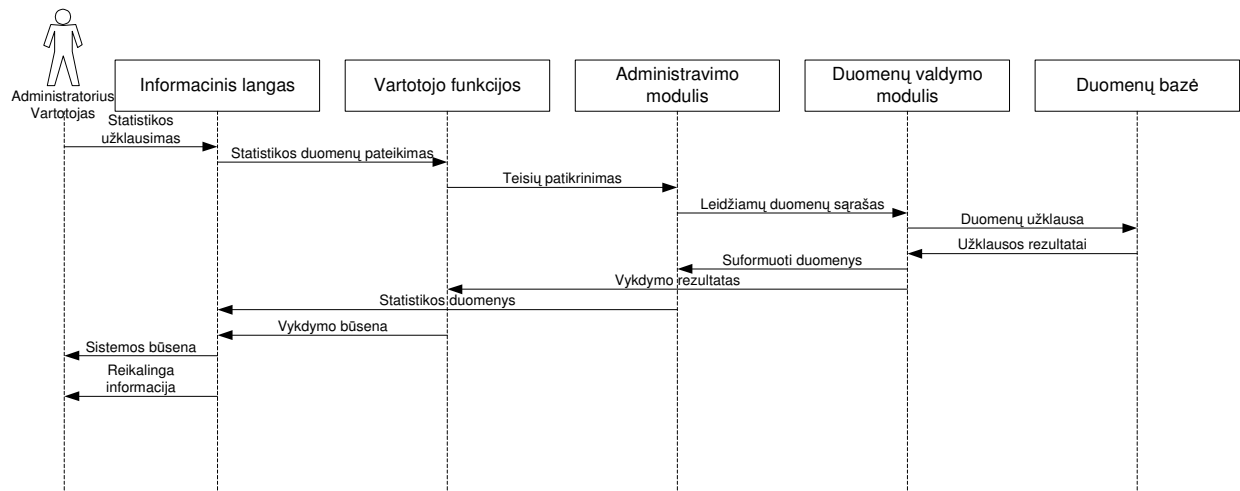
4.4.8. Daugiakalbystės tvarkymas



Pav. 22. Daugiakalbystės tvarkymas

Daugiakalbystės valdymo (pav. 22) scenarijus yra analogiškas sąsajos tvarkymui. Pagrindinis skirtumas yra administravimo modulio funkcijų iškvietime, kadangi yra koreguojamos sistemos kalbos konstantos, o ne atskiri sąsajos elementai. Naujos kalbos įvedimo ir koregavimo scenarijai yra ekvivalentūs.

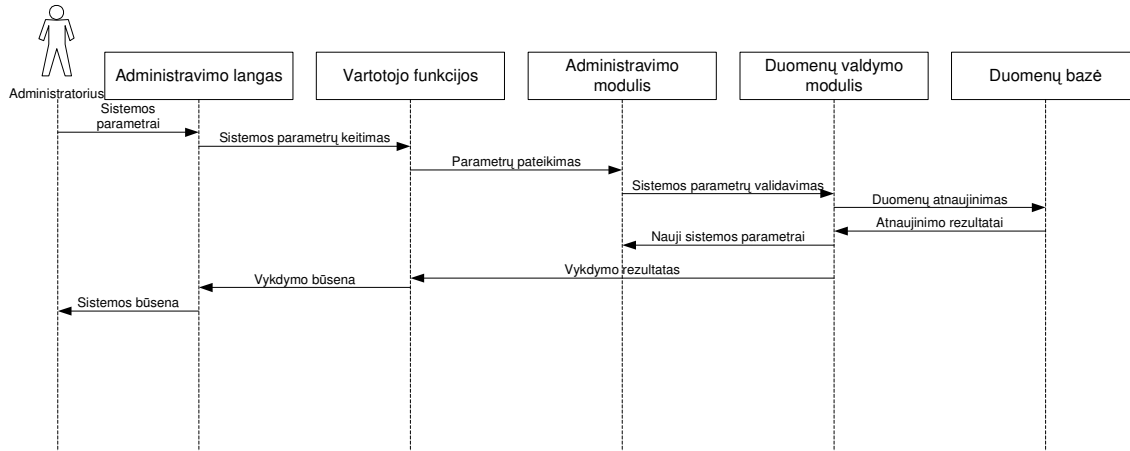
4.4.9. Statistikos peržiūra



Pav. 23. Statistikos peržiūros procesas

Statistikos peržiūros metu (pav.23) yra peržiūrimi lankomumo bei vartotojų atliktų veiksmų žurnalai. Šiuos žurnalus formuoja duomenų valdymo modulis, registruodamas kiekvieną svarbų (duomenų saugumo atžvilgiu) veiksmą.

4.4.10. Sistemos parametų nustatymas



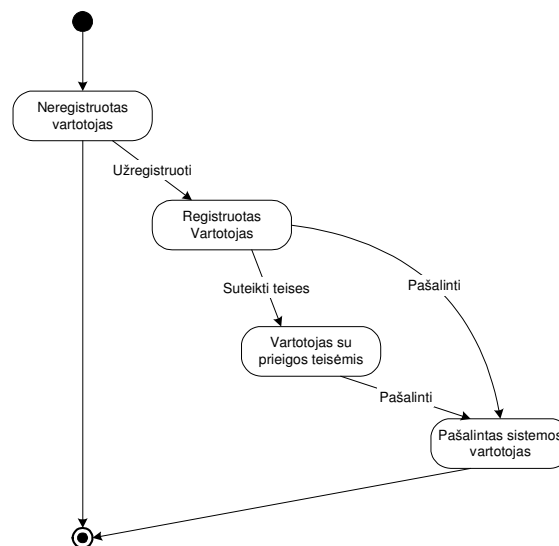
Pav. 24. Sistemos parametų nustatymo procesas

Sistemos parametrai yra nustatomi diegimo metu. Juos nustatinėti galima tik tada, kada internetinėje programoje jau yra suvesta bent viena kalba (daugiakalbystės scenarijus buvo aprašytas anksčiau). Parametų nustatymą gali atlikti tik sistemos administratorius.

4.5. Objektų būsenų kaitos diagramos

Būsenų diagramos leidžia aprašyti modeliuojamų objektų elgesį bei jų būseną tam tikro gyvavimo etapo metu. Kuriamoje sistemoje savo svarba išsiskiria du objektai – tai dokumentas ir vartotojas. Dokumento samprata čia yra išplečiama, nes analogiškai dokumentui yra vaizduojama ir termino duomenų kaita (termino būsenos atitinka dokumento būsenas).

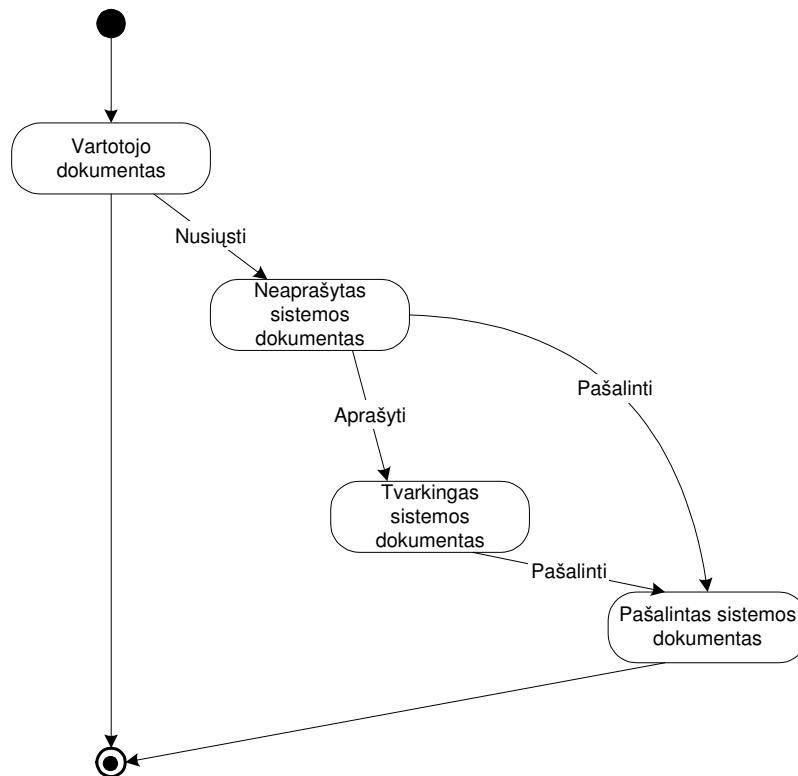
4.5.1. Vartotojas



Pav. 25. Vartotojo objekto būsenų kaita

Vartotojas kaip objektas (pav. 25) pradeda figūruoti sistemoje tik į ją patekęs. Neregistruoto vartotojo matomų duomenų kiekis yra ribojamas. Sistemoje užregistruotas vartotojas įgauna visas teises reikalingas kurti ir redaguoti savo dokumentus bei papildyti terminų žodyną. Sistemos vartotojas gali būti pašalintas administratoriaus, jei netinkamai naudoja jam suteiktas funkcijas arba ilgą laiką nesinaudoja sistemos teikiamomis paslaugomis.

4.5.2. Dokumentas



Pav. 26. Dokumento objekto būsenų kaita

Dokumentas kaip objektas (pav. 26) sistemoje atsiranda vartotojui sukūrus naują dokumentą (bet dar jo neišsaugojus). Parengtas dokumentas tampa matomas ir kitiems sistemos vartotojams, kurie gali jį parsisiųsti. Nebereikalingi ir pasenę sistemos dokumentai yra pašalinami.

4.6. Kuriamos sistemos funkcijų sąrašas

Kuriamų sistemos funkcijų sąrašas yra pateikiamas lentelėse, kiekvieną funkciją apibūdinant atskiroje lentelėje.

Funkcijos nr.	1
Funkcijos pavadinimas	Vartotojo registravimasis sistemoje

Tikslas	Užregistruoti vartotoją sistemos prieigos bazėje ir leisti vartotojui prisijungti prie sistemos suteikiant jam tam tikras prieigos prie informacijos teises.
Aktoriai	Vartotojas
Ryšiai su PA	1
Nefunkciniai reikalavimai	Aiškiai ir nuosekliai išdėstyti reikalavimai veiksmams, kuriuos turi atlikti vartotojas sistemoje norėdamas užsiregistruoti bei lengvai suprantama sąsaja.
Prieš-sąlygos	Vartotojas nebuvo užsiregistravęs sistemoje
Sužadavimo sąlygos	Vartotojas aktyvuoja registravimosi sistemoje funkciją
Po-sąlygos	Vartotojas yra užregistruojamas prie sistemos ir jam yra suteikiamos atitinkamos prieigos teisės, sistemos būsenos modulis informuoja, jog veiksmas įvykdytas be klaidų.
Pagrindinis scenarijus	Norintis prisiregistruoti vartotojas užpildo visus sistemos reikalaujamus rekvizitus ir atlieka klaviatūra įvestų duomenų siuntimo į sistemą veiksmą, po kurio yra informuojamas apie registracijos funkcijos eigos rezultatus.
Alternatyvūs scenarijai	<p>Sistema dėl sutrikimų tinkle negali pasiekti duomenų bazės ir vartotojas nėra užregistruojamas. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.</p> <p>Sistemoje jau yra vartotojas, kuris naudojasi pasirinktu prisijungimo vardu ir sistema prašo pakartotinai įvesti kai kuriuos duomenis. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.</p>

Funkcijos nr.	2
Funkcijos pavadinimas	Vartotojo prisijungimas prie sistemos

Tikslas	Leisti vartotojui prisijungti prie sistemos bei naudotis jos suteikiamu funkcionalumu bei aktyvuoti vartotojo funkcijų modulį bei veiksmų sekimo funkciją.
Aktoriai	Vartotojas
Ryšiai su kitais PA	1
Nefunkciniai reikalavimai	Vartotojui patogi ir suprantama prisijungimo lango sąsaja
Prieš-sąlygos	Vartotojas yra prisiregistravęs sistemoje ir turi prisijungimo vardą bei slaptažodį.
Sužadinimo sąlygos	Vartotojas įveda prisijungimo vardą ir slaptažodį bei aktyvuoja prisijungimo funkciją.
Po-sąlygos	Vartotojas yra prijungiamas prie sistemos, aktyvuojamas vartotojo funkcijų modulis ir veiksmų sekimo funkcija.
Pagrindinis scenarijus	Vartotojo pateikti duomenys yra sulyginami su duomenimis, kurie yra sistemos vartotojų bazėje. Jei randamas reikiamas atitikmuo, vartotojui suteikiamos numatytos prieigos teisės ir aktyvuojamas vartotojo modulis bei pradama vykdyti veiksmų sekimo funkcija. Apie sėkmingą prisijungimą informuoja sistemos būsenos modulis. Vartotojo duomenys registruojami statistikos bazėje.
Alternatyvūs scenarijai	Sistema negali prieiti prie duomenų bazės dėl trikdžių serverio programinėje įrangoje. Vartotojas nėra prijungiamas prie sistemos ir apie tai pranešama pasinaudojant sistemos būsenos moduliu. Sistemos žurnale registruojama klaida.

Funkcijos nr.	3
Funkcijos pavadinimas	Reikalingų duomenų paieška
Tikslas	Rasti duomenis, kurie yra reikalingi vartotojui
Aktoriai	Vartotojas
Ryšiai su PA	1,2

Nefunkciniai reikalavimai	Smulkiai su pavyzdžiais nurodyti kokio tipo užklausas įvesdamas vartotojas gaus jam reikiamus rezultatus bei leisti pasirinkti reikiamus paieškos šaltinius, viską suformuojant paprastais ir efektyviais sąsajos sprendimais. Rezultatai, jei užklaustos atitikimų yra labai daug, rodomi porcijomis, nustatomomis sistemos sąrankoje.
Prieš-sąlygos	Vartotojas įgauna jam skirtas teises.
Sužadinimo sąlygos	Vartotojas suformuoja užklausą ir aktyvuoja duomenų paieškos funkciją.
Po-sąlygos	Vartotojui pateikiami jo prieigos teises atitinkantys paieškos rezultatai.
Pagrindinis scenarijus	Vartotojui aktyvavus užklaustos vykdymo funkciją yra vykdoma paieška vietinėje dokumentų aprašų bei terminų bazėje. Rasti duomenys yra filtruojami pagal vartotojo turimas teises ir pateikiami nuorodų sąrašo pavidalu. Jei vartotojas pažymi, jog paieška turi būti vykdoma ir išorinėse sistemose, duomenų paieška yra atliekama ir nurodytos sistemos bazėje bei visi rezultatai yra pateikiami atskirame lange. Apie sėkmingą veiksmo atlikimą informuoja sistemos būsenos modulis.
Alternatyvūs scenarijai	Sistema dėl sutrikimų tinkle negali pasiekti duomenų bazės ir paieška nėra atliekama. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.

Funkcijos nr.	4
Funkcijos pavadinimas	Dokumentų siuntimas
Tikslas	Registruotas sistemos vartotojas atsiunčia norimus publikuoti dokumentus ir sukuria jų aprašą bei nustato prieigos teises.
Aktoriai	Vartotojas
Ryšiai su PA	1,2

Nefunkciniai reikalavimai	Patogus ir aiški sąsaja, leidžiantis lengvai atlikti aprašymo veiksmus bei meta duomenų sukūrimą. Aiškūs ir informatyvūs veiksmų įvertinimo pranešimai.
Prieš-sąlygos	Vartotojas buvo užsiregistravęs sistemoje ir yra prie jos prisijungęs
Sužadavimo sąlygos	Vartotojas aktyvuoja duomenų tvarkymo funkciją
Po-sąlygos	Vartotojo dokumentai sukūrus aprašus ir meta duomenis persiunčiami į serverį.
Pagrindinis scenarijus	Sistemos vartotojas persiunčia į serverį dokumentą, atlieka jo aprašymą, kuria ir nurodo meta duomenis, pagal kuriuos jis gali būti rastas, susieja jį su egzistuojančiais terminais. Visi veiksmai yra registruojami veiksmų bazėje. Apie veiksmų atlikimo sėkmingumą informuoja sistemos būsenos modulis.
Alternatyvūs scenarijai	<p>Sistema dėl sutrikimų tinkle negali pasiekti duomenų bazės ir dokumentas nėra perduodamas į serverį bei duomenys apie jį nėra išsaugomi. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.</p> <p>Sistema negali patalpinti siunčiamo failo dėl pernelyg didelio jo užimamos vietos kiekio. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.</p>

Funkcijos nr.	5
Funkcijos pavadinimas	Dokumentų parsisiuntimas
Tikslas	Vartotojui gauti reikiamą informaciją iš serverio ir persisiųsti ją į darbo vietą.
Aktoriai	Vartotojas
Ryšiai su PA	1,2,3

Nefunkciniai reikalavimai	Aiškiai ir nuosekliai išdėstyti reikalavimai veiksams, kuriuos turi atlikti vartotojas sistemoje norėdamas parsisiųsti dokumentus.
Prieš-sąlygos	Vartotojas buvo užsiregistravęs sistemoje ir yra prisijungęs.
Sužadavimo sąlygos	Vartotojas aktyvuoja dokumento siuntimo funkciją
Po-sąlygos	Vartotojui reikiamas dokumentas persiunčiamas į jo kompiuterį.
Pagrindinis scenarijus	Vartotojas iš rastų jam tinkamų dokumentų sąrašo išsirenka jam reikiamus dokumentus ir pradeda siuntimą. Parsiuntus dokumentą tai yra registruojama statistikos bazėje ir vartotoją apie sėkmingai atliktą veiksmą informuoja sistemos būsenos modulis.
Alternatyvūs scenarijai	<p>Sistema dėl sutrikimų tinkle negali pasiekti duomenų bazės ir dokumentas nėra persiunčiamas. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.</p> <p>Sistemai patikrinus parsisųsto ir saugykloje laikomo dokumento dydžius randamas skirtumas. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.</p>

Funkcijos Nr.	6
Funkcijos pavadinimas	Terminų žodyno panaudojimas
Tikslas	Vartotojas randa jam reikiamus terminus bei papildos jų sąrašą naujais ir pateikia jų aprašymus.
Aktoriai	Vartotojas
Ryšiai su PA	1,2

Nefunkciniai reikalavimai	Aiškiai ir nuosekliai išdėstyti reikalavimai veiksams, kuriuos turi atlikti vartotojas sistemoje norėdamas redaguoti terminų duomenis. Patogi valdyti terminų žodyno sąsaja.
Prieš-sąlygos	Vartotojas buvo užsiregistravęs sistemoje ir yra prisijungęs.
Sužadinimo sąlygos	Vartotojas aktyvuoja terminų tvarkymo funkciją.
Po-sąlygos	Terminų žodynas papildomas naujais duomenimis arba yra pakoreguojamas.
Pagrindinis scenarijus	Vartotojas pasirenka norimų terminų aprašymus bei juos papildo arba redaguoja (savo). Taipogi įveda naujus arba peržiūri senus terminus bei jų aprašymus bei autorius. Apie veikimo sėkmingumą informuoja sistemos būsenos modulis, visi veiksmai ir pakeitimai yra registruojami žurnale.
Alternatyvūs scenarijai	Sistema dėl sutrikimų tinkle negali pasiekti duomenų bazės ir veiksmai su terminais nėra galimi. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.

Funkcijos Nr.	7
Funkcijos pavadinimas	Sistemos sąsajos tvarkymas
Tikslas	Keisti sistemos sąsajos turinį redaguojant turinio teksto konstantas bei pranešimų žinutes.
Aktoriai	Administratorius
Ryšiai su PA	1,2
Nefunkciniai reikalavimai	Visa sistemos sąsajos informacija yra pateikiama redagavimui patogia forma ir lengvai suprantami sistemos reikalaujami veiksmai.
Prieš-sąlygos	Vartotojas buvo užsiregistravęs sistemoje ir yra prisijungęs bei turi administratoriaus teises.
Sužadinimo sąlygos	Administratorius aktyvuoja sąsajos tvarkymo funkciją.

Po-sąlygos	Pakeitimai atlikti turinio konstantoms yra pateikiami sistemoje.
Pagrindinis scenarijus	Administratorius pasirenka norimos kalbos tekstines turinio konstantas ir keičia jas. Galimi grafikos elementų pakeitimai. Po pakeitimų administratorius išsaugo sąsajos duomenis ir jie yra pateikiami sistemoje. Apie veiksmų teisingumą informuoja sistemos būsenos modulis.
Alternatyvūs scenarijai	Sistema dėl sutrikimų tinkle negali pasiekti duomenų bazės ir veiksmai su turinio konstantomis yra negalimi. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.

Funkcijos Nr.	8
Funkcijos pavadinimas	Daugiakalbystės tvarkymas
Tikslas	Keisti sistemos sąsajos kalbą.
Aktoriai	Administratorius
Ryšiai su PA	1,2,7
Nefunkciniai reikalavimai	Visa sistemos sąsajos kalbų informacija yra pateikiama redagavimui patogia forma ir lengvai suprantami sistemos reikalaujami veiksmai.
Prieš-sąlygos	Vartotojas buvo užsiregistravęs sistemoje ir yra prisijungęs bei turi administratoriaus teises.
Sužadinimo sąlygos	Administratorius aktyvuoja daugiakalbystės tvarkymo funkciją.
Po-sąlygos	Pakeitimai atlikti daugiakalbystės bazėje pateikiami sistemoje.
Pagrindinis scenarijus	Administratorius pasirenka norimą kalbą sistemai ir papildo esamą kalbų sąrašą nauja kalba, vėliau užpildydamas reikalingas konstantas arba jas redaguodamas. Visi veiksmai yra registruojami, apie veiksmų teisingumą informuoja sistemos būsenos modulis.

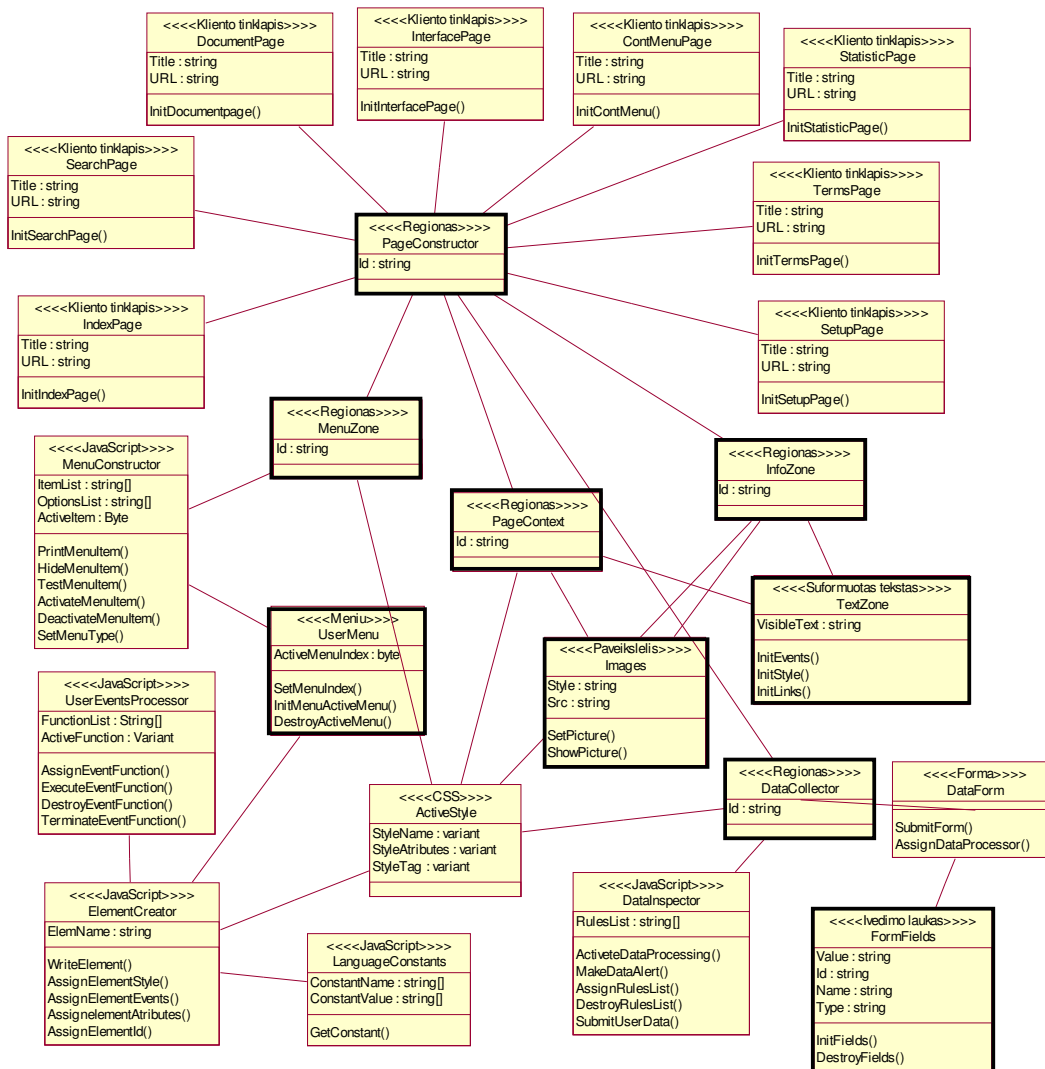
Alternatyvūs scenarijai	Sistema dėl sutrikimų tinkle negali pasiekti duomenų bazės ir veiksmai su daugiakalbystės funkcijomis yra negalimi. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.
-------------------------	--

Funkcija nr.	9
Funkcijos pavadinimas	Sistemos sąsajos tvarkymas
Tikslas	Keisti sistemos sąsajos turinį redaguojant turinio teksto konstantas bei pranešimų žinutes.
Aktoriai	Vartotojas, administratorius
Ryšiai su PA	1,2
Nefunkciniai reikalavimai	Visa statistinė informacija yra pateikiama patogiai ir yra maksimaliai informatyvi.
Prieš-sąlygos	Vartotojas buvo užsiregistravęs sistemoje ir yra prisijungęs.
Sužadinimo sąlygos	Vartotojas aktyvuoja statistikos peržiūros funkciją.
Po-sąlygos	Vartotojai gauna jiems reikiamą informaciją pagal įvairius juos dominančius parametrus ir prieigos teises.
Pagrindinis scenarijus	Vartotojas pasirenka norimą ir jo prieigos teisių leidžiamą statistikos apie duomenis arba lankytojus rūšį ir peržiūri jį dominančią informaciją. Apie veiksmų teisingumą informuoja sistemos būsenos modulis.
Alternatyvūs scenarijai	Sistema dėl sutrikimų tinkle negali pasiekti duomenų bazės ir statistinių duomenų peržiūra yra negalima. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.

Funkcijos nr.	10
Funkcijos pavadinimas	Sistemos parametrų nustatymus.
Tikslas	Keisti parametrus pagal kuriuos yra atliekami vartotojo veiksmai.

Aktoriai	Administratorius
Ryšiai su PA	1,2
Nefunkciniai reikalavimai	Visa parametrai yra pateikiami patogioje formoje ir lengvai suprantama jų reikšmė sistemoje.
Prieš-sąlygos	Vartotojas buvo užsiregistravęs sistemoje ir yra prisijungęs, bei turi administratoriaus teises.
Sužadinimo sąlygos	Administratorius aktyvuoja parametru keitimo funkciją.
Po-sąlygos	Sistemoje yra naudojami naujai nustatyti parametrai.
Pagrindinis scenarijus	Administratorius nustato sistemos vykdomos paieškos parametrus, kai kurių veiksmų parametrus, statistikos bazės išvalymą ir kt. Pabaigęs nustatyti parametras jis tvirtina juos ir šie yra pritaikomi sistemoje. Apie atliktų veiksmų teisingumą informuoja sistemos būsenos modulis, visi veiksmai yra registruojami veiksmų bazėje.
Alternatyvūs scenarijai	Sistema dėl sutrikimų tinkle negali pasiekti parametru duomenų ir bet kokie pakeitimai yra negalimi. Apie tai informuoja sistemos būsenos modulis. Sistemos žurnale registruojama klaida.

4.7. Tinklapių žemėlapis



Pav. 27. Tinklapių navigacijos žemėlapis

Tinklapių žemėlapyje (pav. 27) galima matyti, kaip yra suprojektuotas informacinių sistemų tezauro vartotojo sąsajos formavimas. Visi tinklapiai yra formuojami iš metodikoje pasiūlyto stereotipo regionas elementų. Kiekvienas regionas yra specifinis ir yra apibūdinamas pagal savo paskirtį. Vienas iš tinklapių regionų yra paskiriamas meniu. Šiame regione panaudotas meniu stereotipas iliustruoja nepriklausomą nuo turinio tinklapių formavimo mechanizmą. Panaši struktūra matoma ir duomenų surinkimo zonoje, kur kiekvienas elementas formuojamas įvedimo lauko esybės pagrindu. Paveikslėlis ir suformuotas tekstas – tai stereotipai, kurie padeda formuoti informacinę tinklapių zoną. Kiekvienas elementas yra formuojamas priskiriant stiliaus (CSS) parametrus. Conallen metodiką išplečiantys stereotipai žemėlapyje pavaizduoti tamsesniu rėmeliu.

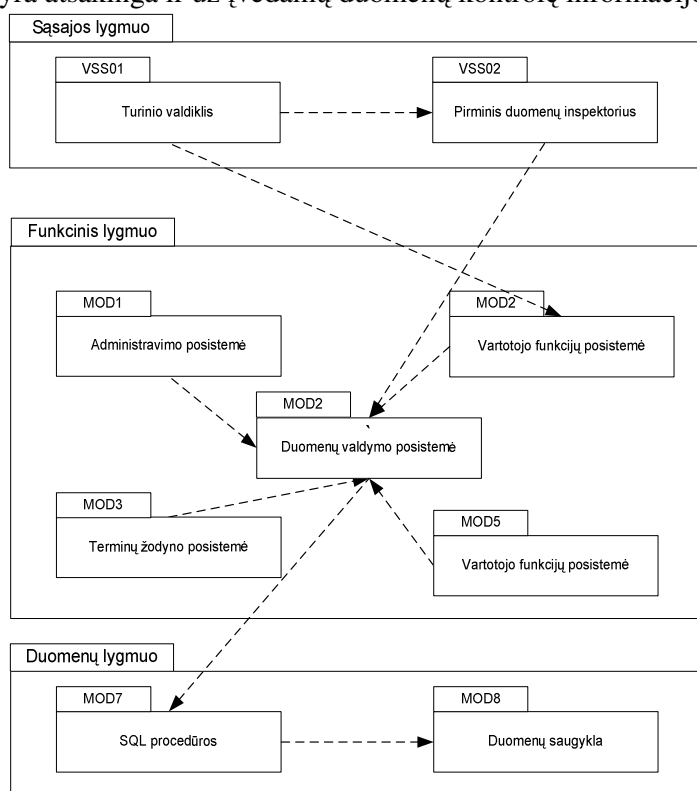
5. PROGRAMINĖ REALIZACIJA INTERNETE

5.1. Apibendrintas architektūros modelis

Sistemos architektūra yra kuriama numatant kad Informacinių sistemų tezauro realizacija internete bus skirta vartotojams, kurie prie sistemos jungsis per atstumą interneto pagalba ir visas jiems reikiamas funkcijas sistema pateiks remdamasi ją sudarančių ir tarpusavyje bendradarbiaujančių modulių realizacija.

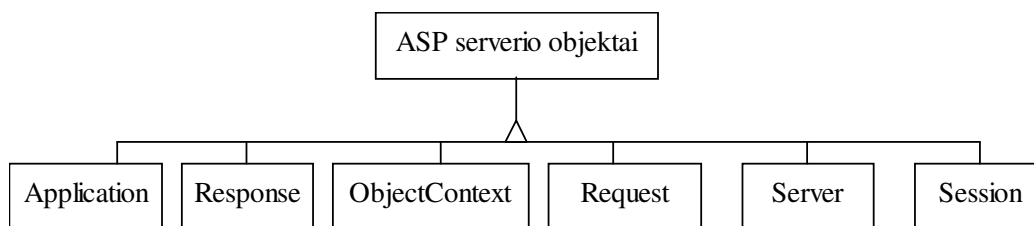
Visa vartotojo sąsaja (pirmasis sistemos lygmuo) yra realizuota bendrosiomis funkcijomis (nepriklausomomis nuo vartotojo prieigos) ir yra tik grafinis pateikimas duomenų, kurie yra gaunami iš sistemos modulių (apsikeitimo duomenų struktūra pateikiama kaip ir duomenų bazės lentelėse).

Vartotojo sąsaja yra atsakinga ir už įvedamų duomenų kontrolę informacijos įvedimo languose.



Pav. 28. Kuriamos sistemos posistemų diagrama

Pagrindinis principas (pav. 28) yra šis: visa sistema yra suskaidyta į programines posistemas, kurios bendradarbiaudamos tarpusavyje pateikia bendrą rezultatą. Kiekviena posistemė yra atsakinga už specifinę sritį, tačiau negali pilnai funkcionuoti, jeigu nėra tvarkingai įdiegtos ir veikiančios likusios sistemos posistemės.



Pav. 29. ASP serverio objektai

Posistemės viena kitą papildo. Vartotojas vienu metu gali naudotis kelių posistemų paslaugomis to nė nejausdamas. Toks sistemos struktūros sudarymas leidžia sugrupuoti sistemoje realizuotas funkcijas pagal posistemas ir vėliau, atnaujinant arba patobulinant jau esamą programinę įrangą lengvai atlikti pakeitimus, kadangi sąlyginai specifinės funkcijos nėra priklausomos vienos nuo kitų (tik nuo įėjimo parametru), todėl keičiant kelias ar daugiau funkcijų nereikia perdarinėti likusių sistemos dalių.

Kiekviena posistemė gali sąveikauti su likusiomis posistemėmis, todėl paskirstymas lygiais yra sunkiai įmanomas, tačiau kaip ir daugumoje internetinių projektų galima išskirti tris sistemos lygius (lentelė 6).

Grafinė vartotojo sąsaja (Vartotojo lygmuo, realizuojamas JavaScript)		
Duomenų patikrinimo/pateikimo funkcijos	Grafinių vaizdų apdorojimo/pateikimo funkcijos	Konstantų pateikimo funkcijos
Duomenų apdorojimo funkcijos (Serverio lygmuo, realizuojamas ASP VBScript)		
Duomenų įrašymo/redagavimo funkcijos	Klaidų aptikimo bei būsenos nustatymo funkcijos	Paieškos vykdymo bei konteksto sudarymo funkcijos
Dokumentų siuntimo/persiuntimo funkcijos	Prieigos valdymo/suteikimo funkcijos	Statistikos rinkimo/pateikimo funkcijos
Duomenų saugojimo funkcijos (DBVS lygmuo, realizuojamas SQL)		
Duomenų redagavimas	Duomenų paieška	Duomenų šalinimas

Lentelė 6. Funkcinis pasiskirstymas architektūros lygiuose

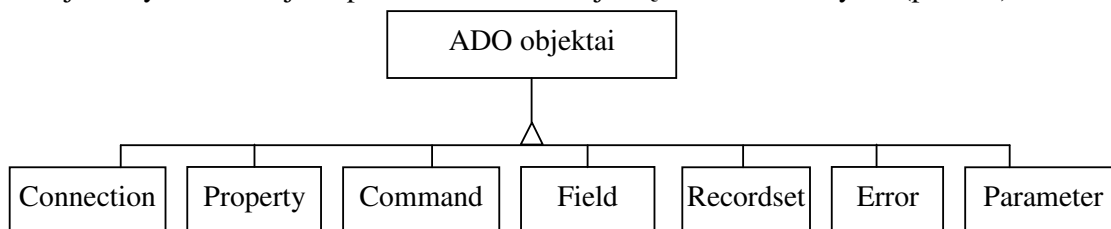
Tokia architektūra leidžia lengvai ir greitai pasinaudoti sistemos suteikiamomis funkcijomis iš nutolusių darbo vietų pasinaudojant internetu. Vartotojui užtenka turėti naršyklę, kuri apdoroja JavaScript kalbą parašytas funkcijas. Visi duomenys yra apdorojami serveryje ir vartotoją pasiekia jau

apdoroti. Taip sumažinamas duomenų srautas ir nelieka perteklinių duomenų, kurie apkrauna tinklą darbo su sistema metu ir lėtina funkcijų atlikimo eigą.

Kiekvienas su duomenų baze susijęs veiksmas yra realizuotas SQL komandų pagalba, todėl duomenys yra apdorojami labai greitai ir sistema vienu metu gali lengvai patenkinti vienu metu prisijungusių vartotojų poreikius, pernelyg neapkraudama DBVS ir taupydama serverio resursus.

5.2. Sistemos struktūra

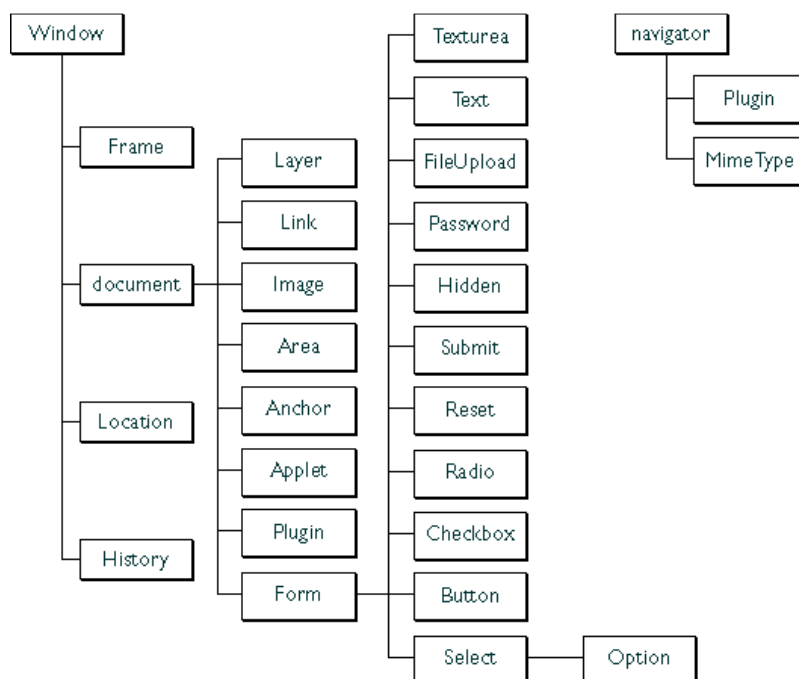
Sistemos architektūra yra kuriama numatant kad Informacinių sistemų tezauro realizacija internete bus skirta vartotojams, kurie prie sistemos jungsis per atstumą interneto pagalba ir visas jiems reikiamas funkcijas sistema pateiks remdamasi ją sudarančių ir tarpusavyje bendradarbiaujančių modulių realizacija. Jie yra realizuojami pasitelkiant ADO objektų metodus ir savybes (pav. 30).



Pav. 30. ADO objektai

Vartotojų sąsajos realizacijai pasitelktas HTML ir JavaScript kodas bei DOM modelis (pav. 31).

Vartotojo sąsaja yra atsakingas ir už įvedamų duomenų kontrolę informacijos įvedimo languose.



Pav. 31. DOM modelis

Sistemą sudaro penkios posistemės: administravimo, duomenų valdymo, vartotojo funkcijų, terminų žodyno ir paieškos.

Administravimo modulis (posistemė) yra atsakingas už visas funkcijas, kurias atlieka administratorius – tai prieigos teisių suteikimas/atėmimas, įvairių sistemos parametrų nustatymas bei peržiūra, statistinių duomenų peržiūra, galimų lokalių klaidų sprendimo funkcijos ir kt. Šis modulis naudojami ir kitų modulių funkcijomis arba jų pagalbiniais metodais.

Administravimo modulis padeda valdyti sistemą, stebėti jos dabartinę eigą bei vartotojų atliktus veiksmus. Naudodamasis šiuo moduliu administratorius gali atlikti įvairių statistinių duomenų peržiūrą, duomenų bei vartotojų šalinimo veiksmus, teisių suteikimo funkcijas bei kalbos konstantų (daugiakalbystės) funkcijų tvarkymą ir bendrų sistemos parametrų nustatymą.

Duomenų valdymo modulis yra atsakingas už sistemos apsaikimą duomenimis tarp vartotojo ir bazės, kuomet yra siunčiamas dokumentas, duomenų tinkamumo, tikslumo bei teisingumo kontrolę bei duomenų apsaikimą tarp visų šešių modulių bei DBVS bei grafinio interfeiso. Dauguma duomenų valdymo modulio funkcijų yra realizuotos SQL komandomis ir dirba DBVS lygmenyje.

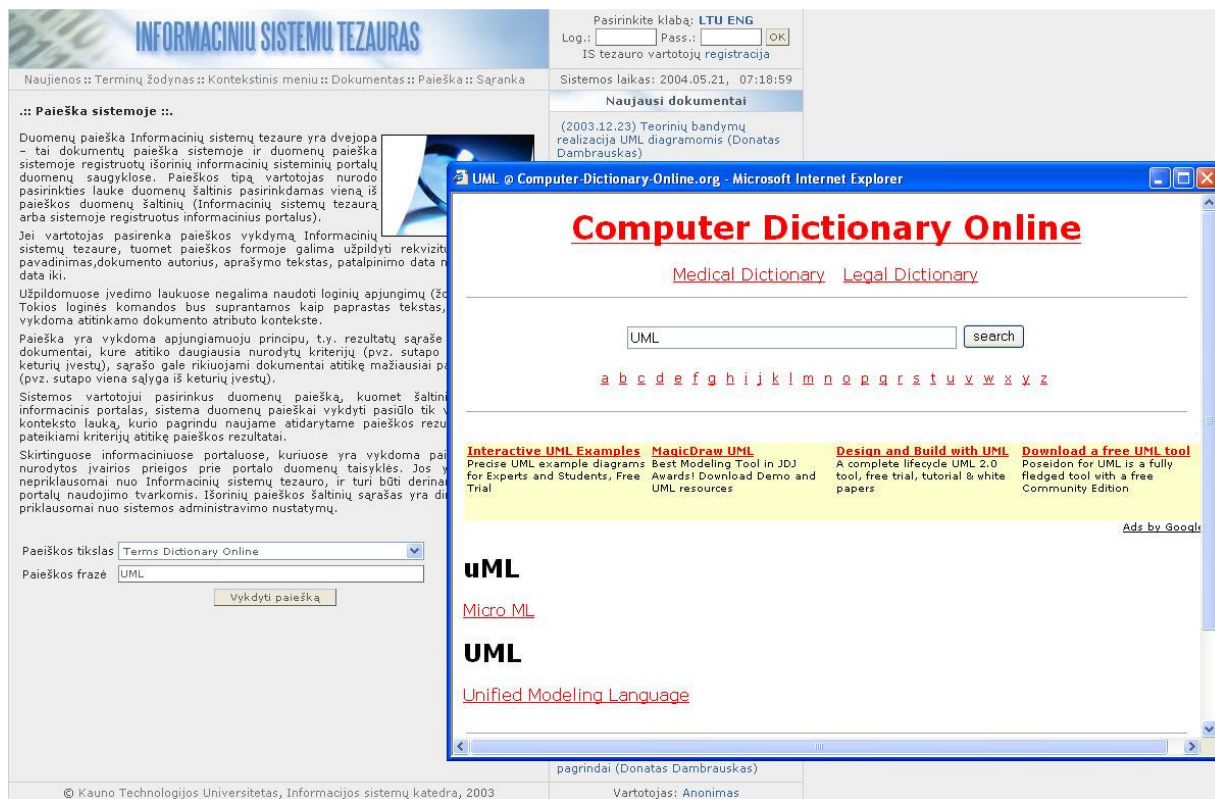
Terminų žodyno modulis yra atsakingas už funkcijas, kurios yra susijusios su terminų žodynu. Tai terminų įvedimo ir redagavimo bei šalinimo funkcijos, termino informacijos bei nuorodų suformavimo funkcijos.

Vartotojo funkcijų modulis yra atsakingas už tam tikro prie sistemos prisijungusio vartotojo aptarnavimą pagal jam priskirtas teises. Vartotojo funkcijų modelis daugumą funkcijų iškviečia iš kitų modulių, pagrindinė jo funkcija yra vartotojo užklausių vykdymas ir jo darbo eigos kontrolė.

Paieškos sistemos modulis atlieka duomenų paieškos sistemos bazėje pagal priklausančias teises funkciją bei kontekstinio turinio suformavimo funkciją. Vartotojui duomenys yra atrenkami ir pateikiami tik leidžiant jo prieigos teisėms (pvz. vartotojas negali matyti dokumentų jei nėra užsiregistravęs). Paieškos funkcijos yra realizuotos SQL komandomis ir vykdomos DBVS lygmenyje. Paieškos parametrus vartotojas įveda sąsajos lygmenyje.

5.3. Paieškos realizavimas

Paieškos mechanizmas yra realizuotas pagal 3.6 skyriuje pateiktą schemą. Vartotojui yra pateikiamas paieškos užklausių įvedimo langas (pav. 32). Įvedus užklausių yra aktyvuojamas JavaScript pagrindu sukurtas pirminis užklausių inspektorius, kuris patikrina ar paieškos užklausių neturi neleistinių simbolių ir ar teisinga jos struktūra. Klaidos atveju veiksmas nutraukiamas ir vartotojui išvedamas pranešimas apie nepavykusį veiksmą.



Pav. 32. Sistemos paieškos langas

Jei įvesta užklausa yra teisinga, tai JavaScript ui inspektorius aktyvuoja serverio duomenų apdorojimo bei paieškos vykdymo posistemas, kurios pagal vartotojo nurodytą paieškos tikslą parenka parametrus tikslinės web programos komandinei nuorodai sukurti. Sudaryta nuoroda yra aktyvuojama naujame naršyklės lange, kuriame ir yra pateikiami paieškos rezultatai.

Visi parametrai, reikalingi paieškos aktyvavimo nuorodai sukurti turi būti įvesti administratoriaus. Iš jo suvesto nuorodų sąrašo vartotojas renkasi paieškos tikslinę programą (išorinę paieškos sistemą).

5.4. Daugiakalbystės realizavimas

Daugiakalbystės ir TVS veikimas yra realizuotas pagal schemą, aprašytą 3.8 skyriuje. Formuojant HTML tinklapį (pav. 33), kuris yra pateikiamas klientui, vietoje konkretaus teksto ar paveikslėlių yra rašomos konstantos. Konstantų pavadinimai yra sudaromi pagal tam tikrą struktūrą, iš kurios būtų aišku, kur ši konstanta bus naudojama ir kokiai kalbai ji priklauso. Kaip pvz. konstanta LIT_SEARCH_WLCM yra sudaryta iš kalbos identifikatoriaus (LIT_), tinklapio arba funkcijos identifikatoriaus (SEARCH_), ir individualaus vardo (WLCM). Ši konstanta yra naudojama pateikti sveikinimo tekstą lietuvių kalba paieškos lange. Formuojant tinklapį yra nurodoma konstantos dalis be kalbos identifikatoriaus (LT_), kuri suformuoja duomenų valdymo posistemę priklausomai nuo sistemoje parinktos kalbos. Konstantos pavadinimas turi būti unikalus. Jei sistemoje yra naudojamos

lietuvių, rusų ir anglų kalbos, tai konstantų saugykloje bus įrašytos LIT_SEARCH_WLCM, ENG_SEARCH_WLCM ir RUS_SEARCH_WLCM konstantos ir jų reikšmės.

The screenshot shows the 'INFORMACINIŲ SISTEMŲ TEZAUROS' (Information Systems Thesaurus) web application. The interface is in Lithuanian. At the top, there is a header with the title and a login section for 'LTU ENG'. Below the header, there is a navigation bar with links like 'Naujienos', 'Terminų žodynas', 'Kontekstinis meniu', 'Dokumentas', 'Paieška', and 'Sąranka'. The main content area is titled 'Kontekstinis meniu' and contains text explaining the context menu feature. To the right of the text is a small image of a keyboard. Below the text is a tree view of categories, with options to 'Išskleisti visas kategorijas' (Expand all categories) and 'Suskleisti visas kategorijas' (Collapse all categories). The right sidebar shows a list of documents, including 'Teorinių bandymų realizacija UML diagramomis' and 'Informacinių sistemų teorijos pagrindai'. The footer contains copyright information for Kauno Technological University and the user name 'Anonimas'.

Pav. 33. Suformuotas sistemos tinklapis

Šias reikšmes gali koreguoti tik administratorius. Jis tai daro atnaujindamas naujienas arba įvesdamas naują sistemos kalbą. Kadangi sistemoje yra naudojamas kešavimo mechanizmas, tai konstantų kalbos identifikatoriaus bei tinklapio arba funkcijos identifikatoriaus pagrindu yra sudaromi resursų (konstantų ir jų reikšmių) failai. Vartotojas kraudamas tinklapį užkrauna tik jam tuo metu reikalingas konstantas. Antrą kartą kraunant tą patį tinklapį iš serverio yra siunčiama tik kintama dinaminė tinklapio dalis, visi kiti resursai jau yra imami iš laikinųjų vartotojo darbo vietos saugyklų. Administratoriui atlikus pakeitimus konstantose visi resursų failai yra perkuriami.

5.5. Vartotojo prieigos realizavimas

Vartotojo prieiga yra realizuojama teisių suteikimo ir tikrinimo pagrindu. Kiekvienam vartotojui, kuris yra prisiregistravęs sistemoje yra suteikiamos teisės. Šios teisės yra pakraunamos į naršyklės sesiją.

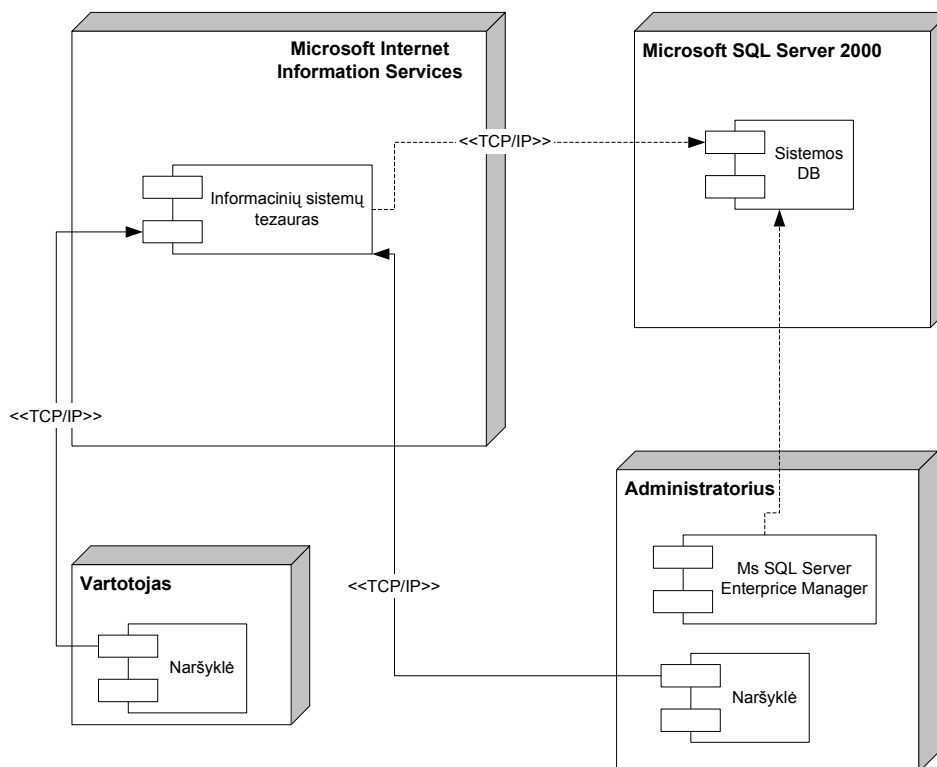
Vartotojui formuojant sistemos tinklą yra tikrinama, kokias teises jis turi ir ar visus veiksmus gali atlikti. Tų veiksmų, kurių vartotojas atlikti negali, sistemos langas nerodo.

Kadangi visi veiksmai iš vartotojo sąsajos yra aktyvuojami komandinėmis nuorodomis, tai atsiranda galimybė sudaryti savo komandinę nuorodą ir atlikti neleistiną veiksmą (pvz. svetimo dokumento šalinimą). Norint tokių situacijų išvengti yra vykdomas antrinis teisių tikrinimas serverio lygmenyje prieš pradėdant aktyvuoti komandinės nuorodos reikalaujamą funkciją. Kadangi prieigos taisyklės yra paprastos, toks antrinis tikrinimas sistemos veikimo efektyvumui įtakos faktiškai nedaro.

Vartotojo teisių talpinimas naršyklės sesijos objekte dar efektyvus tuo, kad vartotojui palikus darbo vietą ir neatsijungus nuo sistemos po tam tikro laiko tarpo (kurį nustato administratorius serverio parametruose) sesija yra paskelbiama nebegaliojanti, ir pašalinis asmuo nebegali atlikti neleistinių veiksmų.

5.6. Paskirstymo diagrama

Paskirstymo diagramoje (pav. 34) matome kaip sistemos elementai yra paskirstomi panaudojant techninius komponentus. Visi jie bendrauja tarpusavyje naudodami TCP/IP protokolą.



Pav. 34. Paskirstymo diagrama

6. IŠVADOS

1. Darbo tikslas buvo sukurti internete veikiančių informacinių sistemų tezaurą, kuris veiktų kaip daugelio vartotojų plečiamas žodynas (turinio valdymo sistema), skirtas informacinių sistemų terminams bei žinioms kaupti.

2. Kuriant šią sistemą, buvo išanalizuoti trijų tipų interneto sistemų projektavimo metodai:

- Metodai, paremti dalykinės srities modeliavimu;
- UML profilis interneto sistemoms projektuoti (arba Conallen metodas);
- Semantinio žiniatinklio metodai.

3. Conallen metode siūlomos praktiškos priemonės kuriamos sistemos architektūrai formuoti, tačiau neskiriama dėmesio dalykinės srities analizei, kuri yra svarbi daugelio interneto sistemų kūrime. Todėl šiame darbe naudojama projektavimo metodika susideda iš Conallen metodo ir jį papildančios dalykinės srities analizės. Semantinio žiniatinklio metodai yra naujoviškesni, tačiau esamu momentu jie dar nėra gerai technologiškai aprūpinti ir šio darbo tikslams pasiekti buvo per daug sudėtingi ir brangūs.

4. Conallen metodas papildytas regiono, meniu, suformuoto teksto, paveikslėlio ir įvesties lauko stereotipais, kurie leidžia detaliau atlikti internetinių informacinių sistemų projektavimą

5. Sukurtas informacinių sistemų tezauro modelis, kuris įgyvendintas naudojant ASP, VBScript, JavaScript, HTML, MS SQL technologijas.

6. Sudarytas modelis leidžia palaikyti daugiakalbystę bei redaguoti sistemos pateikiamo turinio informaciją. Tai pasiekta atskiriant objektų pateiktį nuo jų turinio ir aprašant kalbas konstantų reikšmėmis.

7. Sistemos efektyvumui pagerinti buvo panaudotas ir praktiškai išbandytas sparčiosios atmintinės bei sąsajos resursų paskirstymo mechanizmas, kuris padėjo pagerinti sistemos veikimo rodiklius.

7. Sukurta sistema gali būti naudojama ir kitų organizacijų įvairaus turinio valdymui nepriklausomai nuo organizacijos tipo ar kalbos.

8. Sukurtas tezauro modelis yra universalus, turinio elementus galima išplėsti, nekeičiant likusios sistemos dalies, užtikrinančios daugiakalbystę, turinio įvedimą bei atnaujinimą, vartotojų prieigos valdymą. Sudarytas modelis ir projektavimo metodika gali būti naudinga kitiems projektuotojams, kuriantiems panašias sistemas.

7. LITERATŪROS SĄRAŠAS

1. Brusilovsky, P., Schwarz, E., and Weber, G. (1996) ELM-ART: An intelligent tutoring system on World Wide Web. In C. Frasson, G. Gauthier, & A. Lesgold (Ed.) p. 261-269. Berlynas: Springer Verlag (2001).
2. T. Bryant and A. Evans, OO Oversold - Those objects of obscure desire, *Information and Software Technology*, vol. 36 nr. 1 (1994) p.35-42.
3. N. Davies, M. Davy, G. Blair, and J. Mariani, Object invocation and management in the distributed multimedia information system, *Information and Software Technology* vol. 35 nr. 5 Gegužė (1993) p.259-266.
4. B. Gaines and M. Shaw, WebMap: Concept Mapping on the Web, *World Wide Web Journal*, vol. 1 nr.1 (1995) p.171-183.
5. M. J. Jacobson, C. Maouri, P. Mishra, and C. Kolar, Learning with Hypertext Learning Environments: Theory, Design, and Research, *Journal of Educational Multimedia and Hypermedia*, vol 4(4) (2002), p.321-364.
6. C. Kacmar and J. Legget, PROXHY: A Process-Oriented Extensible Hypertext Architecture, *ACM Transactions on Information Systems* vol. 9 nr. 4, October (1991) p.399-419.
7. D. B. Lange, An Object Oriented Design Method for Hypermedia Information Systems, *Proceedings of the 27th Annual Hawaii Int. conf. on System Sciences* (1994) p.366-375.
8. H. Mauer (Ed.), *Hyper-G now Hyperwave : The next generation Web solution*, Addison-Wesley Longman, (1996).
9. J. Nanard and M. Nanard, Hypertext Design environments and the Hypertext Design Process, *Communications of the ACM*, vol. 38, nr. 8, (1995) p.49-56.
10. R. Nkambou and G. Gauthier, Integrating WWW resources in an Intelligent Tutoring System, *Journal of Network and Computer Applications*, vol. 19 (1999) p.353-365.
11. J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen, *Object-oriented Modeling and Design*, Prentice Hall Inc. (1991).
12. G. Rossi, D. Schwabe, C. Lucena, D. Cowan, An Object Oriented Model for Designing the human-Computer Interface of Hypermedia Applications, *Proceedings of the International Workshop on Hypermedia Design (IWHD'95)*, (1999) p123-143.

8. TERMINŲ IR SANTRUMPŲ ŽODYNAS

CSS – Cascading Style Sheet (internetinių stilių aprašymai).

Dialogo langas – ekrane rodomas langas, kuriame vartotojas gali pateikti tolesniam sistemos darbui reikalingą informaciją.

Duomenų bazė – priemonė duomenims ir jų apdorojimo funkcijoms laikyti.

Forma – rinkinys html elementų duomenims susrinkti.

Java apletas – mažos, savarankiškos taikomosios programos, kurias galima įtraukti (embed) į formas.

JavaScript – Scenarijų rašymo kalba, leidžianti dirbti su dokumento objektiniu modeliu (DOM).

Naršyklė – grafinė sąsaja, skirta vartotojams bendrauti su internetu.

Serveris – kompiuteris, kuriame laikomi failai bei programos ir suteikiama galimybė į failus, programas ir išteklius kreiptis klientams.

Vartotojo sąsaja – ekrano aplinka, suteikianti vartotojui galimybę matyti ir valdyti taikomosios programos veiksmus.

HTML – hiperteksto žymėjimo kalba.

UML – unifikuota modeliavimo kalba.

URL – failo arba resurso interneto adresas.

IS – informacinė sistema.

WWW - pasaulinis kompiuterių tinklas.

HTTP – hiperteksto perdavimo protokolas.

ADO – ActiveX duomenų objektas.

DB – duomenų bazė.

DBVS – duomenų bazių valdymo sistema.

SQL – struktūrizuota užklausų kalba.

CGI – scenarijų rašymo technologija.