

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

Rimantas Žukaitis

DOKUMENTŲ VALDYMO SISTEMOS METADUOMENŲ
APDOROJIMO MODELIO SUDARYMAS IR TYRIMAS

Informatikos mokslų magistro baigiamasis darbas

Darbo vadovas
doc. dr. E. Kazanavičius

KAUNAS, 2004

TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
KOMPIUTERIŲ KATEDRA

TVIRTINU

Katedros vedėjas

doc. dr. E. Kazanavičius

2004 05 19

DOKUMENTŲ VALDYMO SISTEMOS METADUOMENŲ
APDOROJIMO MODELIO SUDARYMAS IR TYRIMAS

Informatikos mokslų magistro baigiamasis darbas

Lietuvių kalbos konsultantė

dr. Jurgita Mikelionienė

2004 05

Recenzentas

doc. dr. R. Butleris

2004 05

Vadovas

doc. dr. E. Kazanavičius

2004 05

Atliko

FM – 8/1 gr. stud.

Rimantas Žukaitis

2004 05 18

KAUNAS, 2004

Žukaitis R. Analysis and Development of Metadata Processing Model for Document Management Systems: Master's Work in Informatics/ supervisor assoc. prof. dr. E. Kazanavičius; Computer Department, Faculty of Informatics, Kaunas University of Technology. – Kaunas, 2004. - 56 p

SUMMARY

The increasing usage of personal computers and Internet in organizations made it possible to create, edit and share various documents between different employees of the organization. However, document management becomes very troublesome, especially if several employees can contribute changes to a single document: it is very hard to locate latest document version, or determine which changes to the document were made by which employee.

Document management systems are aimed to solve these problems. However these systems often are highly specialized and very costly to implement, or they are general-purpose and hard to customize and apply to organization business domain. Inabilities to customize often arise from strict and inflexible metadata model, used in document management system.

The aim of this work is propose abstract document metadata definition and processing model, based on XML data definition language and concept of XML data processing pipeline. The proposed model is general-purpose and highly flexible at the same time, enabling to apply model to any business domain and customize it to reflect any features specific to this domain.

TURINYS

TURINYS	4
1. ĮVADAS.....	6
2. DOKUMENTŲ VALDYMO SISTEMOS	8
2.1. Dokumento gyvavimo ciklas.....	8
2.2. DVS organizacijos informacinių sistemų kontekste	9
2.3. Pagrindinės DVS funkcijos.....	12
2.4. DVS principinė schema	15
3. METADUOMENYS IR DOKUMENTŲ VALDYMAS.....	17
3.1. Metaduomenų sąvoka	17
3.2. Metaduomenų panaudojimas DVS	19
3.2.1. DVS metaduomenų tipai.....	19
3.2.2. Pirminiai dokumento metaduomenys	19
3.2.3. Loginės struktūros organizavimo metaduomenys	20
3.2.4. Versijų kontrolės metaduomenys	22
3.2.5. Prieigos teisių kontrolės metaduomenys	23
3.3. DVS metaduomenų apdorojimo realizacijos	25
3.3.1. Pasirinkimas tarp universalumo ir specializuotumo	25
3.3.2. Nuorodų masyvo realizacija	25
3.3.3. Realizacija panaudojant reliacinę duomenų bazę	26
3.3.4. Sudėtingų aukšto lygio sistemų realizacijos specifi­ka.....	28
4. METADUOMENŲ APDOROJIMAS XML TECHNOLOGIJŲ PAGALBA	29
4.1. Naudojamų XML technologijų aprašymas.....	29
4.1.1. XML duomenų aprašymo formatas	29
4.1.2. XPath ir XSLT technologijos.....	29
4.1.3. XML apdorojimo konvejeriai.....	30
4.1.4. XML duomenų bazės	30
4.2. Dokumento metaduomenų aprašymas XML formatu	31
4.2.1. Metaduomenų aprašymas.....	31
4.2.2. Priminių dokumento metaduomenų aprašymas.....	31
4.2.3. Prieigos teisių metaduomenų aprašymas	33
4.2.4. Loginės organizacijos metaduomenų aprašymas.....	35
4.2.5. Versijų kontrolės metaduomenų aprašymas.....	36
4.3. Dokumentų ir metaduomenų saugojimas XML formatu.....	37

4.3.1. Dokumentų saugykla	37
4.3.2. Metaduomenų saugykla	37
4.4. Metaduomenų apdorojimo modelis	38
4.4.1. XML apdorojimo konvejerio komponentai	38
4.4.2. Metaduomenų apdorojimas.....	39
4.4.3. Prieigos teisių apdorojimas.....	41
4.4.4. Dokumento ir jo metaduomenų peržiūra.....	41
4.4.5. Dokumento ir jo metaduomenų patalpinimas	42
4.4.6. Pirminių metaduomenų apdorojimas dokumento paieškos metu.....	43
4.4.7. Metaduomenų apdorojimas dokumento redagavimo metu	44
4.5. Apibendrinimas	45
5. TESTINIS PROGRAMINIS MODULIS.....	47
5.1. Realizacija.....	47
5.1.1. Java programavimo kalba	47
5.1.2. Apache Cocoon XML konvejerių karkasas	47
5.1.3. Apache Xindice XML duomenų bazė.....	48
5.1.4. Realizuota programinė įranga.....	48
5.2. Programinė įranga testavimui.....	48
5.2.1. XML duomenų bazių generavimas	48
5.2.2. Užklausų generatorius	49
5.3. Rezultatai	49
5.3.1. Dokumento skaitymo testavimas	49
5.3.2. Skirtingų dokumentų valdymo operacijų palyginimas	50
5.3.3. Apibendrinimas.....	51
6. IŠVADOS.....	52
7. LITERATŪRA	54

1. ĮVADAS

Daugumos šiuolaikinių organizacijų veiklos pagrindas yra darbas su įvairaus pobūdžio elektroniniais dokumentais. Augant apdorojamos informacijos kiekiui, kartu auga ir cirkuliuojančių įmonėje dokumentų kiekis, tampa vis sunkiau juos suvaldyti. Dažniausiai dokumentai saugomi jų autorių asmeniniuose kompiuteriuose, o kai reikia juos pateikti kolegoms – yra spausdinami ant popieriaus arba siunčiami elektroniniu paštu. Jeigu dokumente atliekami kokie nors pakeitimai, procesas kartojamas iš naujo. Situacija dar labiau komplikuojasi, jei vieną dokumentą turi redaguoti keli autoriai. Taip atsiranda daugybė vieno dokumento kopijų ir praktiškai prarandama svarbi su dokumentu susijusi informacija: kokia yra paskutinė dokumento versija, kokie jame atlikti pakeitimai, kas jų autorius.

Šioms problemoms spręsti kuriamos specialios informacinės sistemos – dokumentų valdymo sistemos (DVS). DVS panaudojimas remiasi fizinės ir organizacinės tvarkos įvedimo idėja. Fizinė tvarka įgyvendinama įmonėje sukuriant centralizuotą dokumentų saugyklą, kurioje yra saugomi visi įmonėje apdorojami dokumentai. Organizacinė tvarka įvedama apibrėžiant įmonės darbo su dokumentais politiką, kuri nusako, kaip turi būti kuriami, redaguojami bei peržiūrimi dokumentai, nurodomos įvairių vartotojų prieigos prie dokumentų teisės. Ši politika yra įgyvendinama programiškai realizuojant DVS programinę įrangą (1). Kartu su dokumentu DVS saugoma papildoma su juo susijusi informacija – dokumento metaduomenys. Metaduomenys – tai dokumento informacija, kuri nėra tiesiogiai susijusi su dokumento turiniu, tačiau yra naudojama dokumentui sistemoje apdoroti: dokumento autorius, sukūrimo data, prieigos teisės, sąsajos su kitais sistemoje saugomais dokumentais ir kt.

Siekiant, kad DVS įdiegimas duotų įmonei kuo daugiau naudos, diegiama sistema turi būti specializuota, maksimaliai pritaikyta prie įmonės poreikių. Tokios sistemos kuriamos kiekvienai įmonei atskirai, po ilgų dalykinės srities ir įmonės poreikių tyrimo. Tačiau tokių sistemų trūkumas yra tas, kad jos yra labai brangios. Kita vertus, daug programinės įrangos gamintojų siūlo palyginti nebrangias, masiniam naudojimui skirtas DVS – vadinamąsias „universalias“ DVS. Tačiau tokių produktų trūkumas yra tas, kad jas labai sunku, o kartais net ir neįmanoma pritaikyti prie specifinių įmonės poreikių.

Šio darbo tikslas: *suprojektuoti ir iširti konceptualų DVS metaduomenų aprašymo ir apdorojimo modelį, kuris leistų realizuoti šiuolaikinėms DVS būdingą funkcionalumą*. Šis modelis turi būti pakankamai lankstus bei abstraktus, kad būtų tinkamas taikyti bet kuriai dalykinei sričiai.

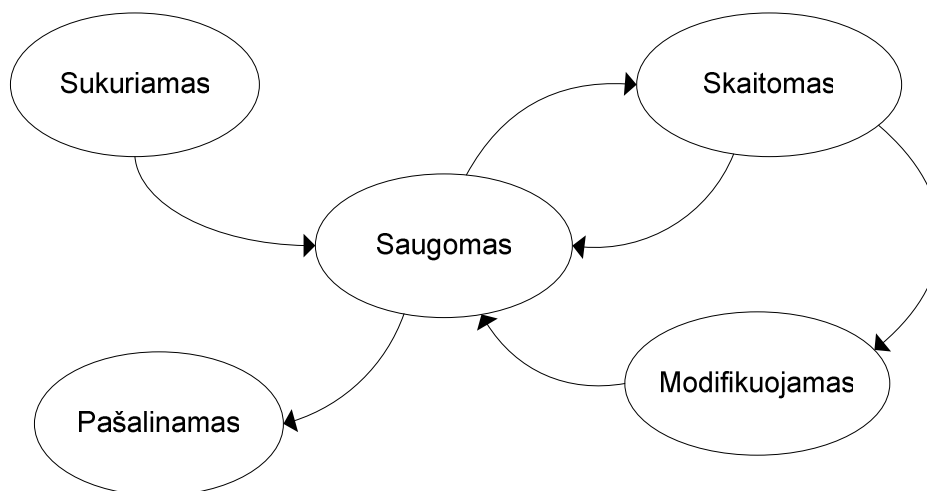
Šiame darbe pristatomas naujas dokumento metaduomenų aprašymo bei apdorojimo metodas, paremtas XML (angl. *eXtensible Markup Language*) duomenų aprašymo kalba. Metaduomenų apdorojimas atliekamas naudojant XML duomenų apdorojimo konvejerius, kurie yra vykdomi kiekvieno vartotojo kreipimosi į DVS metu. Sukurtas testinis programinis modulis, realizuojantis pateiktą metaduomenų apdorojimo modelį. Atlikti eksperimentai, išmatuotas sistemos reakcijos laiko pailgėjimas, susijęs su metaduomenų apdorojimu.

2. DOKUMENTŲ VALDYMO SISTEMOS

2.1. Dokumento gyvavimo ciklas

Prieš pradėdant dokumentų valdymo sistemų apžvalgą, reikia susipažinti su dokumento gyvavimo ciklo sąvoka. Dokumento gyvavimo ciklas apibūdina, kokiose būsenose būna konkretus dokumentas savo egzistavimo įmonėje metu, ir kaip šios būsenos kinta.

Imkime paprasčiausią pavyzdį – tarkime kad įmonėje su elektroniniais dokumentais dirba tik vienas asmuo (pvz., sekretorė) visi dokumentai yra saugomi šio asmens kompiuteryje. Su dokumentu atliekami šie pagrindiniai veiksmai – sukūrimas, išsaugojimas, skaitymas(dokumento turinio peržiūra), modifikavimas bei dokumento pašalinimas (ištrynimasis iš kompiuterio kietojo disko). Šiuo atveju dokumento gyvenimo gyvavimo ciklas atrodo taip:



1 pav. Dokumento gyvavimo ciklo pavyzdys

Šis ciklas atspindi patį paprasčiausią dokumentų apdorojimo būdą. Dokumentas yra sukuriamas, tuo momentu jis yra kūrimo būsenoje. Vartotojas, baigęs kurti pradinę dokumento versiją išsaugo ją kompiuterio diske. Nuo šio momento dokumentas yra saugojimo būsenoje ir prieinamas skaitymui. Jeigu vartotojas sugalvoja koreguoti dokumento informaciją, dokumentas pereina į modifikavimo fazę. Kuomet pakeitimai yra išsaugomi į kietąjį diską ir modifikavimas baigiamas, dokumentas vėl grįžta į saugojimo būseną.

Kuomet su elektroniniais dokumentais įmonėje dirba keletas žmonių, o ypač kuomet vieną dokumentą redaguoja keletas darbuotojų, situacija komplikuojasi. Tarkime, kad vienas darbuotojas paruošia pradinę konkretaus dokumento kopiją, ir ją koku būdu (elektroniniu paštu, diskelių pagalba ar kt.) perduoda kitiems suinteresuotiems darbuotojams. Tokiu atveju

dokumento būseną tampa daugiareikšmė, kadangi egzistuoja kelios to paties dokumento kopijos.

Mažose įmonėse, kuomet apdorojamų dokumentų kiekis ir darbuotojų skaičius yra palyginti nedidelis, ši problema dažnai sprendžiama „žodiniu susitarimu“ – t. y. darbuotojai tarpusavyje susitaria, kokia tvarka bus redaguojami vieno ar kito tipo dokumentai. Pavyzdžiui, yra sutariama, kad apsišvietimui dokumentų kopijomis bus naudojamas įmonės failų serveris ir kiekvienam darbuotojui skirtas dokumentas bus perkeliamas į katalogą su jo vardu. Tačiau tokia „sistema“ yra labai trapi ir neatspari klaidoms – nėra jokių galimybių užtikrinti, kad to paties dokumento vienu metu neredaguotų keli žmonės ar kad tam tikram darbuotojui būtų pateikiama tikrai pati paskutinė konkretaus dokumento versija.

Dokumento gyvavimo ciklo sąvoka yra labai svarbi, projektuojant įmonės DVS. Būtent dokumento gyvavimo ciklas nusako, kaip DVS pagalba yra apdorojami dokumentai – kokia dokumentų patalpinimo tvarka, kaip atliekamas dokumento išgavimas (angl. *retrieval*) bei modifikavimas. Konkrečiu atveju, kuomet pasirinkta DVS diegiama konkrečioje įmonėje, dokumento gyvavimo ciklas yra realizuojamas DVS priemonėmis, pritaikant (angl. *customizing*) sistemą prie įmonės poreikių. Kitą vertus, daugeliu atveju dokumento gyvavimo ciklas taip pat turi būti modifikuojamas ir taikomas prie konkrečios DVS realizacijos (3).

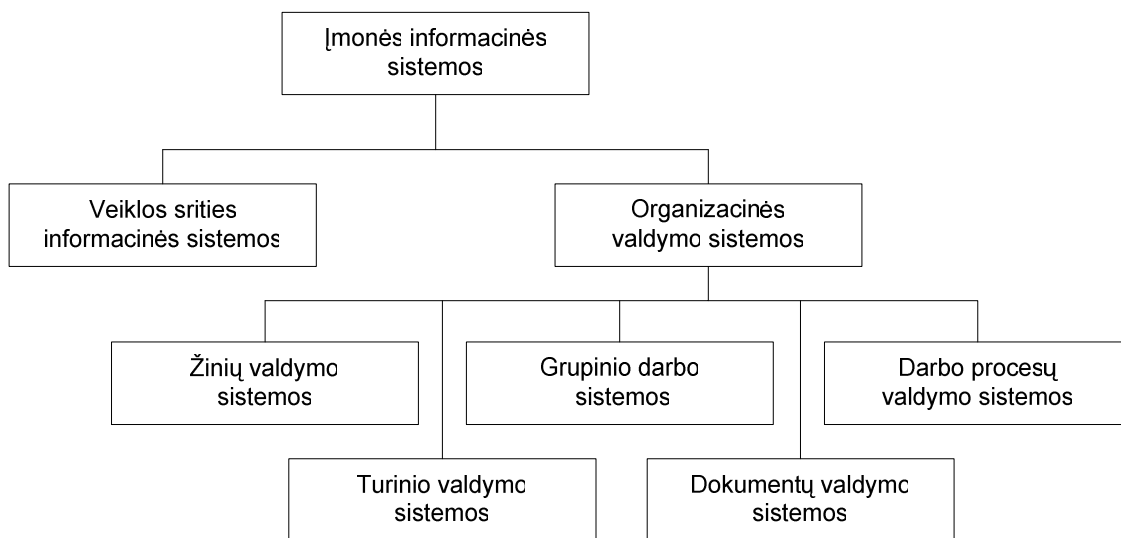
2.2. DVS organizacijos informacinių sistemų kontekste

Greta įvairių informacinių sistemų, skirtų tiesioginiams įmonės veiklos srities (angl. *application domain*) uždaviniams spręsti, tokių kaip įvairios duomenų bazės, apskaitos sistemos, produktų užsakymo sistemos ir kt., egzistuoja organizacinių valdymo sistemų klasė. Tai sistemos, kurios nėra tiesiogiai orientuotos į verslo problemų sprendimą, bet yra skirtos vidinių įmonės procesų valdymui ir pagerinimui. Atsižvelgdamas į tai, kokioms organizacinėms problemoms spręsti jos taikomos, M. Sutton (3) pateikia tokią šių informacinių sistemų klasifikaciją (žr. 2 pav.):

Žinių valdymo sistemos (angl. *knowledge management systems*) – sudėtingos informacinės sistemos, skirtos žinių apie veiklos sritį, sukauptų per visą įmonės veiklos laikotarpį, apdorojimui ir valdymui. Tokių sistemų pagrindinė funkcija – palengvinti sprendimų priėmimą, iškilus kokiam nors standartinei situacijai šios sistemos dažniausiai būna labai specializuotos ir kuriamos atskirai pagal kiekvienos įmonės poreikius ir specifiką (4).

Grupinio darbo sistemos (angl. *groupware*) – šiuo terminu literatūroje vadinamos įvairios programinės sistemos skirtos darbuotojų tarpusavio bendradarbiavimo (angl. *collaboration*) kompiuterinio tinklo pagalba organizavimui ir pagerinimui. Prie šios

kategorijos priskiriamos vidinio pašto bei žinučių (angl. *instant messaging*) sistemos, taip pat failų apsiskeitimo, lygiagreto resursų redagavimo bei versijų kontrolės priemonės (5).



2 pav. Informacinių sistemų klasifikacija

Darbo procesų valdymo sistemos (angl. *workflow management systems*) – skirtos verslo procesų modeliavimui. Procesas suskaidomas į keletą žingsnių, kurie dažniausiai yra atliekami paeiliui skirtingų žmonių. Dažniausiai tai realizuojama persiunčiant formos pavidalo dokumentus ir kiekviename žingsnyje skirtingiems darbuotojams jas pildant. Tokių sistemų specifika yra, kad dažniausiai reikia apdoroti didelius kiekius duomenų, užtikrinti aukštą saugumo lygį bei korektišką kiekvieno žingsnio atlikimą (5).

Turinio valdymo sistemos (angl. *content management systems*) – sistemos, skirtos informacijos valdymui. Jos yra orientuotos į informacijos publikavimą įmonės interneto svetainėje (arba įmonės vidinio tinklo svetainėse, prieinamose tik įmonės darbuotojams). Pagrindinis dėmesys šiose sistemose skiriamas informacijos įvedimo, dar vadinamo turinio kūrimu (angl. *content creation*) bei informacijos atvaizdavimo priemonėms (6).

Dokumentų valdymo sistemos (angl. *document management systems*) – sistemos, skirtos informacijos, saugomos elektroninių dokumentų pavidalu, valdymui. Pagrindinis dėmesys skiriamas dokumentų saugojimui, paieškai, bei su jais susijusių metaduomenų apdorojimui (6).

Realiose įmonėse, ypač didelėse, retai kada šių tipų sistemos yra realizuojamos atskirai. Dažniausiai įmonėje yra įdiegiama viena sistema, integruojanti keletą ar netgi visų aukščiau išvardintų tipų posistemės. Pavyzdžiui, darbo procesų valdymo priemonės dažnai realizuojamos kaip dokumentų valdymo sistemos arba turinio valdymo sistemos dalis. Grupinio darbo priemonės taip pat dažnai remiasi dokumentų (arba turinio) valdymo sistemos

funkcionalumu darbo grupės nariams prieigai prie bendrų darbo resursų (failų, dokumentų ir pan.) užtikrinti. Vieną sistemą sudaro keletas posistemių, kurios apima reikalingą įmonėje įdiegti funkcionalumą. Tokios integruotos sistemos vadinamos organizacijos valdymo sistemomis (angl. *enterprise management systems*) (3).

Verta pastebėti, kad dažnai (ypač programinės įrangos gamintojų puslapiuose) terminai „turinio valdymo sistema“ bei „dokumentų valdymo sistema“ vartojami vienas vietoje kito. Dažniais atvejais siūlomi produktai, kurie apjungia tiek dokumentų valdymo, tiek turinio valdymo funkcijas. Taip yra dėl to, kad daugelis komercinių turinio valdymo sistemų savyje apjungia ir dalį dokumentų valdymo funkcijų. C. Banayay (6) pabrėžia, kad dokumentų valdymo sistemoje pagrindinis dėmesys skiriamas dokumentui, kaip minimaliam informacijos vienetui, kai tuo tarpu turinio valdymo sistemoje pagrindinis dėmesys yra skiriamas informacijos (turinio) loginės struktūros organizacijai, neakcentuojant, kaip fiziškai yra saugoma informacija.

Verta paminėti, kad DVS sąvoka yra taikoma nusakyti dvejoms skirtingo funkcionalumo sistemų grupėms:

- Fizinį dokumentų valdymo sistemos – sistemos skirtos valdyti dokumentų, kurie yra saugomi popierinėje formoje (taip pat dokumentų, saugomų mikrofilmų ar kitokių neskaitmeninių laikmenų pavidalu). Tai dažniausiai yra liktinės (angl. *legacy*) sistemos arba sistemos, taikomos specifinės veiklos srityje įmonėse (pvz., įvairūs finansinių ar teisinių dokumentų archyvai, kuriuose būtina išsaugoti dokumento originalą). Šiuo atveju informacinėje sistemoje apdorojami tik dokumento metaduomenys.
- Elektroninių dokumentų valdymo sistemos – sistemos, skirtos dokumentų, saugomų elektroniniame pavidale, valdyti. Šiuo atveju sistemoje saugomi ir apdorojami ir dokumento metaduomenys, ir pats dokumentas.

Kadangi šiame darbe yra nagrinėjamos tik elektroninių dokumentų valdymo sistemos, terminas „dokumentų valdymo sistema“ bus taikomas tik šio tipo sistemoms apibrėžti. Taip pat dokumentų valdymo sistemos pagal taikymo sritį yra skirstomos į dvi grupes:

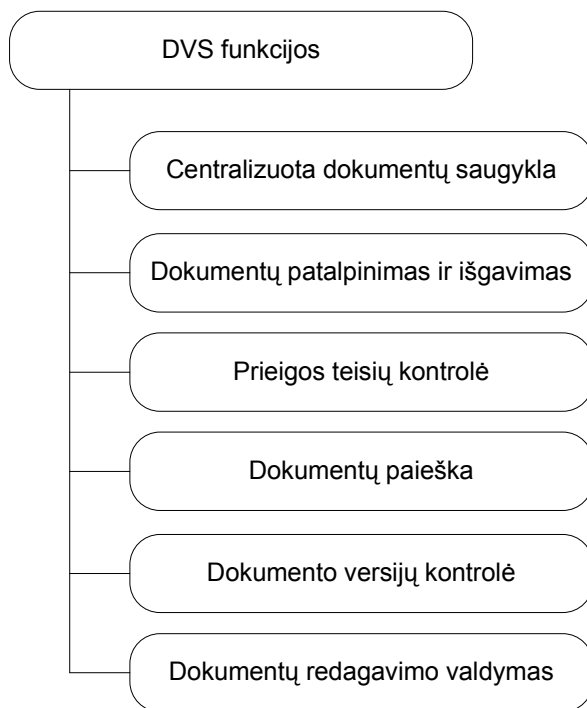
- Dokumentų saugojimo sistemos – skirtos saugoti dideliems dokumentų kiekiams, pvz., dokumentų archyvams, elektroninėms bibliotekoms ir pan. Tokiose sistemose pagrindinis dėmesys skiriamas efektyviam dokumentų saugojimui bei greitiems ir tiksliais dokumento paieškos algoritmams, o dokumentų redagavimo bei loginės dokumentų struktūros organizavimui skiriamas mažesnis dėmesys. Tokio tipo sistemos dar vadinamos dokumentų archyvavimo arba dokumentų išgavimo (angl. *document retrieval*) sistemomis.

- Įmonės dokumentų valdymo sistemos – skirtos organizuoti darbą su įmonėje cirkuliuojančiais dokumentais. Šiuo atveju kaip tik akcentuojama dokumentų loginė organizacija, redagavimo, patalpinimo bei išgavimo priemonės. Tolimesnėje šio darbo dalyje bus nagrinėjamos būtent tokio tipo sistemos.

2.3. Pagrindinės DVS funkcijos

Dokumentų valdymo sistemomis dažnai yra vadinamos įvairaus funkcionalumo ir sudėtingumo lygio sistemos – nuo primityvių failų serverių iki sudėtingų informacinių sistemų, integruojančių dokumentų valdymą kartu su kitais įmonėje vykstančiais organizaciniais procesais. Taip pat dauguma dokumentų valdymo sistemų yra specializuotos ir maksimaliai pritaikytos prie konkrečios įmonės poreikių bei veiklos srities - kadangi įmonė, įdiegdama tokią sistemą, nori iš jos gauti kiek galima daugiau naudos. Kita vertus, universalių dokumentų valdymo sistemų nepopuliarumą galima paaiškinti tuo, kad egzistuojančias universalias sistemas dažnai yra sunku pritaikyti prie konkrečios įmonės poreikių.

Žemiau pateikiamas bazinių universalios dokumentų valdymo sistemos funkcijų sąrašas:



3 pav. DVS funkcijų diagrama

Centralizuotas dokumentų saugojimas – pagrindinė DVS funkcija. Visi dokumentai saugojami vienoje vietoje, griežtai apibrėžtoje fizinėje struktūroje. Tai gali būti vienas ar keli failų serveriai ar duomenų bazės. Sistemos vartotojui pateikiama loginė dokumentų valdymo

sistemos struktūra, kuri dažniausiai skiriasi nuo fizinės saugyklos realizacijos. Tai smarkiai palengvina dokumentų valdymą, kadangi vartotojui pateikimas supaprastintas dokumentų saugyklos vaizdas, paslepiant visas sistemos realizacijos detales. Loginiam atvaizdavimui dažniausiai naudojama hierarchinės struktūros (pvz. katalogų medis).

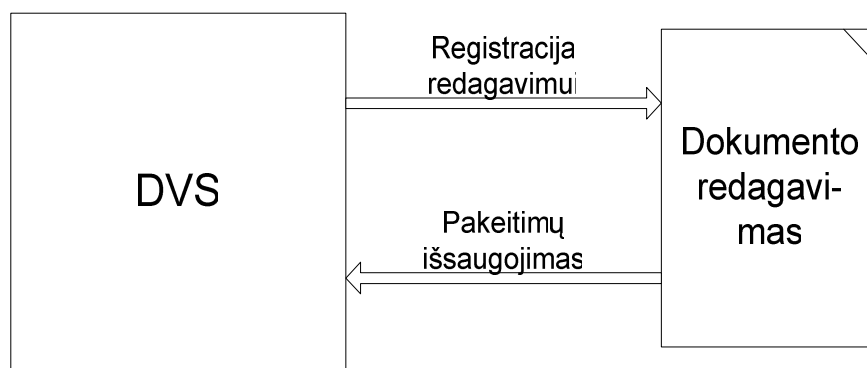
Dokumentų patalpinimo ir išgavimo funkcijos. Šios priemonės priklauso nuo konkrečios sistemos realizacijos. Dažniausiai dokumentų patalpinimas vykdomas interneto naršyklės arba tinklinės failų sistemos pagalba. Retesniais atvejais naudojama specialiai tam tikslui skirta programinė įranga, instaliuojama vartotojo. Aukšto lygio sistemose naudojami specialūs programiniai moduliai, užtikrinantys tiesioginę integraciją su dokumentų kūrimo programiniais paketais, tokiais kaip „*Microsoft Office*“. Dokumentų išgavimas (angl. *retrieval*) taip pat atliekamas šių priemonių pagalba.

Prieigos teisių kontrolė. Visose be išimties dokumentų valdymo sistemose naudojamas vieno ar kito tipo mechanizmas prieigai prie dokumentų užtikrinti. Apibrėžiamos „vartotojo“ (angl. *user*) ir vartotojo teisių (angl. *user permissions*) sąvokos. Vartotojai prisijungimui prie sistemos naudoja vartotojo vardą ir slaptažodį. Darbo su sistema metu, vartotojui atlikti konkretų veiksma (peržiūrėti, modifikuoti dokumentą ar kt.) leidžiama tik tuo atveju, jei tai leidžia jo prieigos teisės. Minimali prieigos teisių tipų aibė susideda iš dviejų veiksmų – „skaityti“ (peržiūrėti) ir „rašyti“ (modifikuoti), tačiau dažnai sistemoje egzistuoja daug daugiau prieigos teisių tipų.

Dokumentų paieška – paieškos modulis ypač svarbus sistemose, kurios skirtos dideliame dokumentų kiekiui saugoti (pvz., dokumentų archyvams, elektroninėms bibliotekoms ir pan.). Paieška dažniausiai atliekama, apdorojant dokumento raktinius žodžius (angl. *keywords*). Aukšto lygio sistemose, siekiant pagerinti paieškos rezultatus, naudojami dokumento teksto indeksavimo arba raktinių žodžių generavimo algoritmai.

Dokumento versijų kontrolė – sistemoje redaguojant dokumentą, po kiekvieno dokumento atnaujinimo išsaugoma ir modifikuota, ir senoji dokumento versija. Tokiu būdu, sistemoje saugomi visi dokumento pakeitimai ir galima pažiūrėti, kaip kito dokumento versijos laikui bėgant. Ši funkcija yra labai svarbi dokumentų valdymo sistemose, naudojamose dažnai redaguojamiems ir besikeičiantiems dokumentams apdoroti.

Dokumento redagavimo valdymas – priemonės, skirtos užtikrinti, kad keli darbuotojai tame pačiame dokumente nepadarytų pakeitimų vienu metu. Jei susidarytų tokia situacija, kad vienu metu dokumentą redaguoja keli žmonės, tai išsaugant pakeitimus į sistemą, būtų išsaugoma tik vėliausiai darbą baigusio darbuotojo pakeitimai. Šioms problemoms išvengti, dažniausiai naudojama „registracijos redagavimui/išsaugojimo“ schema:



4 pav. Dokumentų redagavimo DVS schema

Norėdamas redaguoti sistemoje saugomą dokumentą, vartotojas atlieka registracijos (angl. *check-in*) redagavimui veiksmą. Tuomet sistemoje pažymima, kad šis dokumentas yra redaguojamas, ir kitiems darbuotojams prieiga prie šio dokumento yra negalima. Baigęs redaguoti dokumentą, vartotojas išsaugo pakeitimus, ir sistema „atlaisvina“ (angl. *check-out*) dokumentą, vėl atstatydama prieš redagavimą buvusias prieigas prie dokumento teises.

Aukšto lygio sistemose ši problema kartais sprendžiama realizuojant sinchroninio redagavimo priemones. Tokiu atveju sistema seka, kokius būtent pakeitimus dokumento tekste atlieka vienas ar kitas darbuotojas. Kuomet pakeitimus dokumente vienu metu bando išsaugoti keletas vartotojų, panaudojamas apjungimo (angl. *merging*) algoritmas ir kelios skirtingos dokumento versijos išsaugojimo metu apjungiamos į vieną.

Sudėtingos dokumentų valdymo sistemos, naudojamos didelėse organizacijose, dažnai turi papildomų funkcijų. Dažnai sutinkamas pakeitimų stebėjimo (angl. *change tracking*) funkcionalumas – kuomet modifikuojamas konkretus dokumentas (arba pvz. į kažkokį katalogą patalpinamas naujas dokumentas) apie tai pranešama visiems šio dokumento stebėjimo paslaugą užsisakiusiems sistemos vartotojams. Pakeitimų stebėjimo modulis dažniausiai būna integruotas su įmonės elektorinio pašto serveriu arba grupinio darbo sistemomis.

Jau buvo minėtas darbo procesų valdymo palaikymas (žr. poskyrį 2.4). Jam realizuoti dokumento gyvavimo ciklas yra išplečiamas, papildant jį dokumento būsenomis, atitinkančiomis atskirus darbo proceso žingsnius. Dokumento perėjimo iš vienos būsenos į kitą metu dokumentas yra perkeliamas tarp atskirų saugyklos medžio katalogų, susietų su atitinkamais darbo proceso žingsniais.

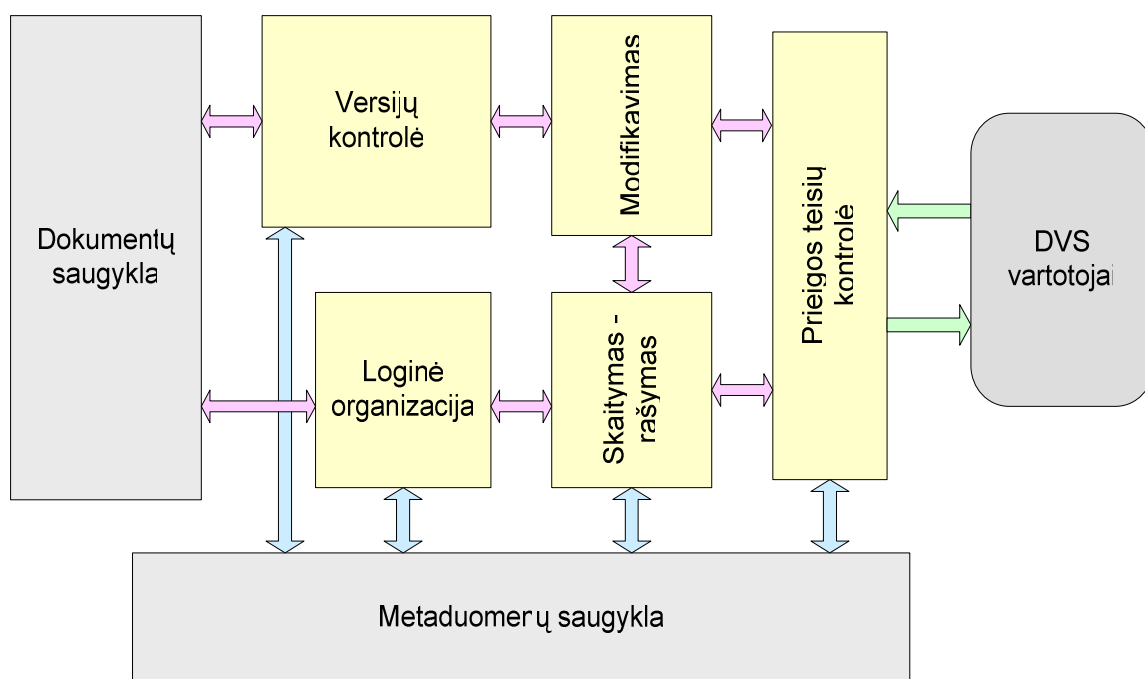
Tam tikra dalis dokumentų, apdorojamų įmonėje yra laikini. Pvz. dokumentas su skyriaus viršininko nurodymu, kad pasitarimas vyks valanda vėliau, nebetenka prasmės praėjus keletui dienų po susitikimo. Tokiems dokumentams apdoroti yra naudinga automatinio pašalinimo funkcija. Dokumento autorius gali nurodyti datą ir laiką, iki kada

dokumentas galioja. Atėjus nurodytam momentui dokumentas yra pašalinamas iš sistemos (arba, priklausomai nuo realizacijos, yra apdorojamas kitu būdu – pvz. perkeliamas į atskirą specialiai tam tikslui skirtą katalogą).

2.4. DVS principinė schema

Kadangi dokumentų valdymo sistema yra skirta naudotis daugeliui vartotojų vienu metu, ji naudoja kompiuterių tinklą savo funkcionavimui užtikrinti ir yra realizuojama kliento-serverio principu. Klientai – sistemos vartotojai – pateikia serveriui užklausas, kur jos yra apdorojamos ir klientui pateikiamas atsakymas.

Pateiksime supaprastintą dokumentų valdymo sistemos veikimo schemą:



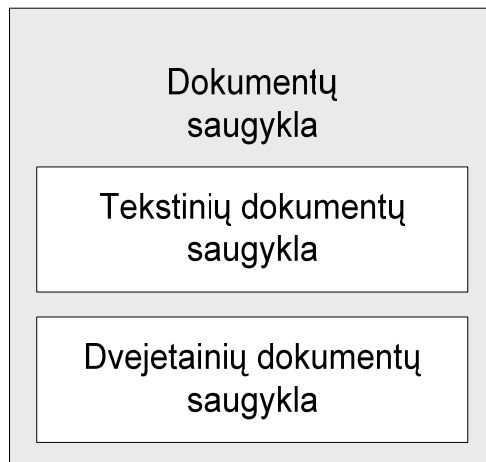
5 pav. Principinė DVS funkcionavimo schema

Kiekvienas vartotojo kreipinys į sistemą pirmiausia apdorojamas prieigos teisių kontrolės modulyje. Prieigos teisių informacija saugoma metaduomenų saugykloje. Jeigu vartotojui leidžiama atlikti konkretų veiksą, kreipinio apdorojimas perduodamas apdoroti atitinkamam sistemos moduliui. Kiekvieno atliekamo veiksmo metu apdorojamas tiek pats dokumentas, tiek su juo susiję metaduomenys.

Dokumentų valdymo sistemose apdorojamus dokumentus yra priimta skirstyti į dvi grupes – tekstinius dokumentus ir dvejetainius dokumentus. Tekstiniai dokumentai – dokumentai, kuriuose saugoma tekstinė informacija, aprašyta žmogui suprantamais simboliais. Dvejetainiai dokumentai – juose informacija yra saugoma dvejetainėje formoje.

Tai įvairūs paveiksliukai, garsai, filmuota medžiaga, failų archyvai, o taip pat įvairių programinių paketų duomenų failai. Tekstinių dokumentų apdorojimas automatizuotomis priemonėmis (pvz., indeksavimas ar raktinių žodžių generavimas) yra daug paprastesnis, todėl, rekomenduojama informaciją dokumentų valdymo sistemose saugoti tekstinių dokumentų pavidalu. Daugelis aukšto lygio dokumentų valdymo sistemų pateikia priemones konvertuoti populiarių programinių paketų pagalba kuriamus dokumentus į vidinį sistemoje naudojamą formatą (kuomet tai yra įmanoma). Dažniausiai šios priemonės yra integruojamos į dokumentų patalpinimo/išgavimo modulius.

Kadangi tekstinių ir dvejetainių dokumentų tiek apdorojimo tiek saugojimo specifiška gana smarkiai skiriasi, dažnai dokumentų valdymo sistemose naudojama dvilypė fizinė dokumentų saugykla:



6 pav. Dvilypė dokumentų saugykla

Reikia paminėti, kad tokiu atveju saugykla dvilypė yra tik fiziniu požiūriu – tai yra dokumentų saugojimo priemonių realizacijos požiūriu. Sistemos vartotojui, matančiam tik loginį saugyklos vaizdą, šis dvilypumas yra visiškai nematomas (angl. *transparent*). Tai yra užtikrinama programinėmis priemonėmis.

3. METADUOMENYS IR DOKUMENTŲ VALDYMAS

3.1. Metaduomenų sąvoka

Kas yra metaduomenys? Dažniausiai sutinkamas atsakymas į šį klausimą – metaduomenys yra duomenys apie duomenis. Metaduomenys yra aprašomojo pobūdžio informacija apie objektą, su kuriuo jie yra susieti. Metaduomenys gali būti panaudoti aprašyti praktiškai bet kam – objektams, su kuriais dirba konkreti informacinė sistema, atskiroms lentelėms ar net ištisoms duomenų bazėms, įvairioms programinėms sistemoms ar jų teikiamoms paslaugoms, žmonėms ir jų rolėms informacinėse sistemose. Pavadinimas „metaduomenys“ yra palyginti naujas ir pradėtas naudoti informacinių technologijų srityje, atsiradus vis didėjančiam poreikiui integruoti atskiras informacines sistemas.

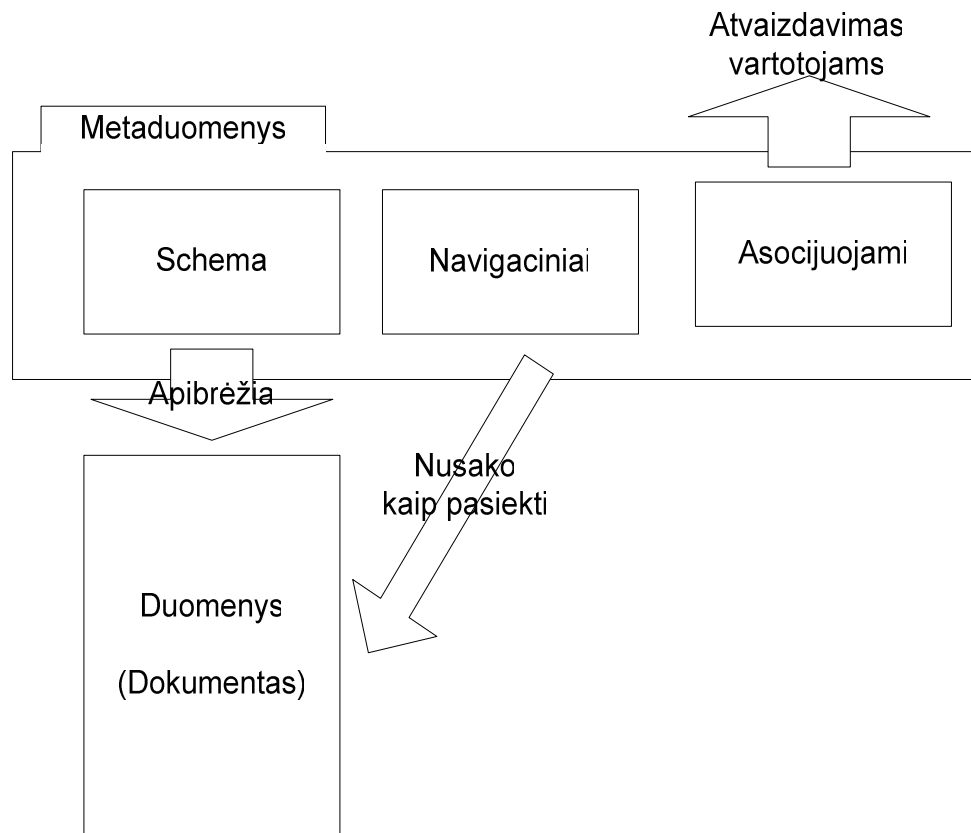
Kadangi metaduomenų standartizavimo procesas dar tik prasideda, kiekvienoje taikymo srityje kuriami atskiri metaduomenų aprašymo bei panaudojimo standartai. Kaip pavyzdžius būtų galima paminėti geografinių metaduomenų (8), meteorologinių metaduomenų (9) ar interneto resursų aprašymo standartą RDF (angl. *Resource Description Framework*) (10).

Kadangi egzistuoja tiek daug skirtingų metaduomenų taikymo sričių ir su jomis susijusių metaduomenų aprašymo būdų ir standartų, pas terminas „metaduomenys“ tampa nevienareikšmiu. K. Jefferey (7) pateikia tokį metaduomenų apibrėžimą: metaduomenys – tai su pagrindiniais sistemos apdorojamais duomenimis susijusi ir juos tam tikru apsektu aprašanti informacija. Taip pat išskiriamos tokios metaduomenų grupės (pagal panaudojimo paskirtį):

- Metaduomenys, skirti integracijai (angl. *data exchange metadata*) – naudojami aprašyti duomenims, kurie naudojami apsikeitimui informacija tarp skirtingų sistemų. Kadangi informacinių sistemų integracija tampa vis aktualesne problema, ši metaduomenų grupė yra labai svarbi – dėl šios priežasties netgi pati metaduomenų sąvoka literatūroje dažnai sutapatinama su šio tipo metaduomenimis.
- Metaduomenys, skirti užklausų (angl. *query*) optimizavimui. Į šią grupę įeina metaduomenys užklausų formavimo bei užklausų rezultatų generavimo pagerinimui. Šio tipo metaduomenys yra labai svarbūs paieškos funkcijas turinčioms sistemoms.
- Metaduomenys, palengvinantys pagrindinių duomenų apdorojimą. Į šią grupę patenka metaduomenys, aprašantys su pagrindiniais duomenimis susijusią informaciją, kuri užtikrina korektišką jų interpretavimą ir apdorojimą.

Taip pat metaduomenų aprašoma informacija skirstoma į tris rūšis (11):

- Metaduomenų schema (angl. *metadata schema*) – metaduomenų aprašymo formatas. Apibrėžia kaip ir kokia informacija yra nurodoma metaduomenyse.
- Navigaciniai metaduomenys (angl. *navigational metadata*) – nusako kaip pasiekti metaduomenimis aprašomus pradinius duomenis (pvz., RDF atveju navigaciniai duomenys yra aprašomo resurso URI – angl. *Uniform Resource Identifier*). Prie navigacinių metaduomenų taip pat priskiriamos prieigos prie aprašomų duomenų teisės.



7 pav. Metaduomenų rūšys (7)

- Asocijuojami metaduomenys (angl. *associated metadata*) – papildoma informacija, kuri susiejama su pradiniais duomenimis. Aprašoma metaduomenų schemos apibrėžiamu formatu.

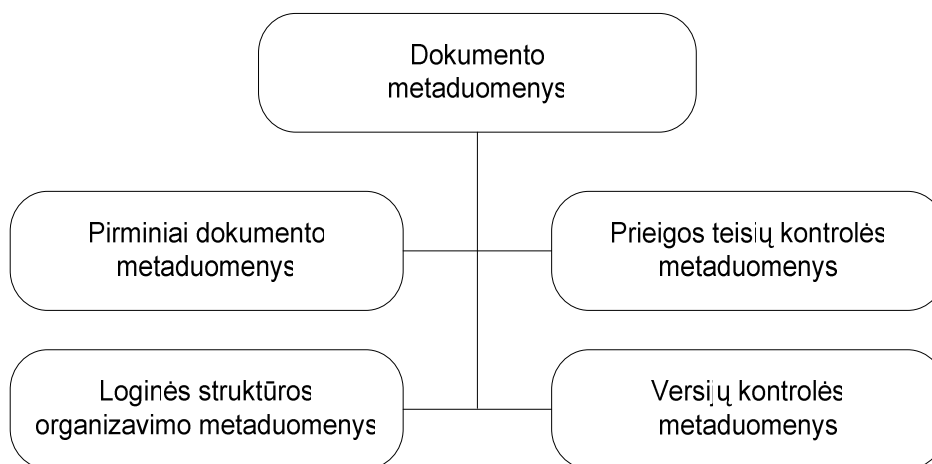
Šių trijų rūšių metaduomenų tarpusavio sąryšis pavaizduotas 7 pav. Priklausomai nuo taikymo srities, realizuojamoje sistemoje naudojama vienos, kelių ar visų rūšių metaduomenys. Metaduomenų schema įtraukiama į konkretaus objekto metaduomenų aprašą

tik tuo atveju, kuomet metaduomenys skirti duomenų apskaitimui tarp skirtingas metaduomenų schemas naudojančių sistemų.

3.2. Metaduomenų panaudojimas DVS

3.2.1. DVS metaduomenų tipai

Kadangi dauguma dokumentų valdymo sistemų yra specializuotos ir pritaikytos konkrečiai sričiai, skiriasi ir jose naudojamų metaduomenų tipai bei metaduomenų apdorojimo modeliai. Metaduomenų panaudojimo laipsnis, jų aprašymo metodas bei apdorojimo modelis iš esmės priklauso nuo sistemoje realizuojamo funkcionalumo. Jeigu nagrinėtume universalią dokumentų valdymo sistemą, kurios funkcionalumas buvo pavaizduotas 3 pav. (žr. poskyrį 2.3), būtų galima išskirti tokius naudojamų metaduomenų tipus:



8 pav. DVS naudojamų metaduomenų tipai

Dokumentų valdymo sistemos vartotojui tiesiogiai prieinami tik pirminiai dokumento metaduomenys (angl. *document primary metadata*) ir (iš dalies) prieigos teisių metaduomenys: pavyzdžiui, patalpindamas į sistemą naują dokumentą, vartotojas apibrėžia prieigos prie to dokumento teises.

3.2.2. Pirminiai dokumento metaduomenys

Pirminiai dokumento metaduomenys – aprašo tiesiogiai su dokumento turiniu susijusią papildomą informaciją, kuri nėra aprašoma pačiame dokumente arba jos išgavimas iš dokumento yra per sudėtingas. Priklausomai nuo dokumentų valdymo sistemos taikymo srities, šiuose metaduomenyse dažnai aprašoma skirtinga informacija – pavyzdžiui, medicininių dokumentų valdymo sistemoje kiekvienam dokumentui (ligos istorijai)

priskiriamas atsakingas gydytojas, palatos numeris ir kita panaši, vien tik ligos istorijų aprašymui skirta informacija. Kitoje taikymo srityje – pavyzdžiui meteorologinių stebėjimų dokumentų valdymo – taikymo sričiai specifiniai duomenys gali būti visai kitokie.

Deja, dar nėra apibrėžto ir visuotinai pripažinto pirminių dokumento metaduomenų aprašymo standarto, skirto universalioms dokumentų valdymo sistemoms. Skirtingose tokių sistemų realizacijose naudojamos skirtingos pirminių metaduomenų elementų aibės. Kai kuriose realizacijose yra adaptuojami metaduomenų standartai iš kito tipo sistemų – pavyzdžiui *Dublin Core Metadata* (12) standartas interneto resursų pirminių metaduomenų aprašymui.

Pateiksime pavyzdinę dokumento pirminių metaduomenų elementų aibę, kuri sutinkama daugumoje universalių sistemų:

- pavadinimas – dokumento pavadinimas, naudojamas sistemoje;
- autorius – dokumento autorius ar autorių sąrašas;
- tema – dokumento tema;
- aprašymas – platesnis dokumento aprašymas;
- sukūrimo / patalpinimo / modifikavimo datos – nurodo, kada dokumentas buvo sukurtas, kada patalpintas į sistemą, bei kada buvo atlikti vėliausi pakeitimai;
- dokumento tipas – pvz., ataskaita, priminimas, el. laiško kopija ir pan.;
- dokumento formatas – koku formatu saugoma dokumento informacija. Dažniausiai nurodomas dokumento failo MIME tipas (13);
- dokumento raktiniai žodžiai – raktinių žodžių aibė, apibendrinanti dokumento turinį. Metaduomenyse aprašomi raktiniai žodžiai naudojami dokumentų paieškos realizavimui.

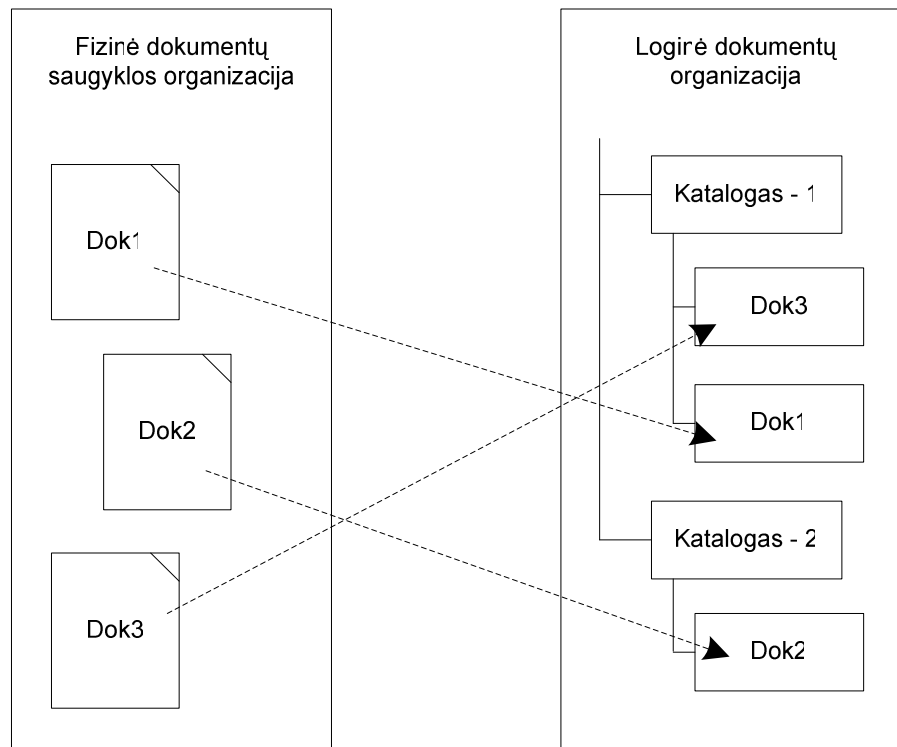
Šio tipo metaduomenis įveda sistemos vartotojas dokumento talpinimo į dokumentų valdymo sistemą metu. Dažnai sistemose yra realizuojamos papildomos priemonės automatiniam kai kurių elementų generavimui – pvz., dokumento patalpinimo bei paskutinio modifikavimo datos beveik visada generuojamos sistemos priemonėmis.

3.2.3. Loginės struktūros organizavimo metaduomenys

Kaip jau buvo minėta poskyryje 2.3, pagrindinė dokumentų valdymo sistemos funkcija yra centralizuotas dokumentų saugojimas. Šioje vietoje įvedamos dvi sąvokos susijusios su centralizuotos dokumentų saugyklos sąvoka – tai fizinė saugyklos struktūra bei loginė saugyklos struktūra.

Fizinė dokumentų struktūra – struktūra nusakanti kaip sistemoje yra realizuotas dokumentų saugojimas. Tai gali būti duomenų bazė, failų sistema ar kažkoks kitas realizacijos

būdas. Loginė dokumentų valdymo sistemos struktūra – tai yra loginė dokumentų organizavimo tvarka. Ji nusako sistemos vaizdą, kuris yra pateikiamas sistemos vartotojui. Labiausiai paplitęs yra hierarchinės katalogų struktūros – arba katalogų medžio – atvaizdavimo būdas.



9 pav. Fizinės struktūros atvaizdavimas į loginę struktūrą

Loginės ir fizinės struktūros atskyrimas turi didelį privalumą – realizuojant fizinę struktūrą, visas dėmesys yra skiriamas efektyviam dokumentų saugojimui bei patalpiniui ir išgavimui. Kadangi galima nekreipti dėmesio į fizinės struktūros pateikimą vartotojui, galima realizuoti efektyvesnę saugyklą. Savo ruožtu, kadangi loginis atvaizdavimas projektuojamas nepriklausomai nuo fizinės realizacijos, galima realizuoti vartotojui pateikti patogesnę loginę struktūrą.

Fizinės struktūros atvaizdavimui į vartotojui pateikiamą loginę struktūrą naudojami loginės struktūros organizavimo metaduomenys. Kiekvienam sistemoje saugomam dokumentui metaduomenyse nurodoma jo vieta loginėje dokumentų struktūroje.

Taip pat šie metaduomenys gali būti naudojami papildomam funkcionalumui realizuoti:

- sąryšių tarp atskirų dokumentų apibrėžimui. Tokiu atveju vieno dokumento metaduomenyse išsaugoma nuoroda į susijusius dokumentus;

- vieno fizinio dokumento priskyrimas keliems loginiams katalogams – tokiu atveju metaduomenyse nurodomas visų katalogų, kuriems dokumentas priklauso, sąrašas.

3.2.4. Versijų kontrolės metaduomenys

Versijų kontrolės metaduomenys yra skirti versijų kontrolės funkcijai realizuoti. Kiekvieną kartą kai dokumente, saugojamame sistemoje, atliekami pakeitimai, sistemoje išsaugojama nauja ir sena dokumento versijos. Tokiu būdu galima atsekti, kaip kito dokumentas savo gyvavimo ciklo metu.

Versijų kontrolei realizuoti egzistuoja keli būdai (3):

- Kopijų išsaugojimo – šiuo atveju, vartotojui padarius pakeitimus dokumente, išsaugoma ir sena, ir nauja dokumento kopijos. Naujausia dokumento versija yra saugoma pagrindinėje dokumentų saugykloje. Su ja susiejamos visos senesnės dokumentų versijos, kurios yra saugomos atskiroje versijų saugykloje. Vartotojui norint peržiūrėti kažkurią konkrečią dokumento versiją, ji yra pateikiama iš versijų saugyklos. Šiuo atveju dokumento versijų pagalba išsaugomi tik dokumento turinio pakitimai – metaduomenų pakitimai nėra fiksuojami.
- Dokumentų dubliavimo – šiuo atveju, išsaugant modifikuotą dokumento versiją, yra išsaugoma ir sena, bei dubliuojama visa su dokumentu susijusi informacija. Kartu su senesne versija yra išsaugojami visi dokumento metaduomenys – faktiškai į sistemą yra patalpintas naujas dokumentas, kuris nėra atvaizduojamas loginiame katalogų medyje, bet yra susiejamas su naujausia versija. Šis metodas leidžia sekti visus pakeitimus, atliekamus su dokumentu ar jo metaduomenimis, tačiau jis reikalauja daugiausiai resursų fizinėje dokumentų saugykloje – visos dokumento ir jo metaduomenų versijų kopijos reikalauja daug pastoviosios atminties saugojimui.
- Pokyčių registravimo – šiuo atveju, išsaugant modifikuotą dokumento versiją, yra išsaugomi tik naujai vartotojo padaryti pakeitimai. Pakeitimai yra aptinkami lyginant naują dokumento versiją su senąja. Kiekvienam dokumentui yra saugoma pradinė dokumento versija bei padarytų pakeitimų seka. Vartotojui norint peržiūrėti kažkurią konkrečią dokumento versiją, sistema pagal visus išsaugotus pakeitimų iš pradinės dokumento versijos sugeneruoja vartotojo nurodytą. Šis metodas yra naudingas, kuomet dokumentai yra labai dažnai redaguojami – tokiu atveju sutaupomi resursai fizinėje saugykloje, kurie būtų

naudojami dokumento kopijoms saugoti. Šio metodo esminis trūkumas – pateikiant dokumentą vartotojui, sistema turi apdoroti visą pokyčių informaciją, kas reikalauja papildomų skaičiavimo resursų.

Atskirų dokumento versijų susiejimas yra atliekamas panaudojant versijų kontrolės metaduomenis, kurie yra paprastai yra susiejami su pagrindine dokumento versija. Juose yra aprašoma, kokie pakeitimai buvo padaryti dokumente, pakeitimų autorius bei data, o taip pat nuoroda į atskiras dokumento versijas (arba į kiekvieną dokumento versiją atitinkančias pakeitimų aibes pakeitimų registravimo metodo atveju).

3.2.5. Prieigos teisių kontrolės metaduomenys

Prieigos teisių valdymas susijęs su autentifikavimo (angl. *authentication*) ir autorizacijos (angl. *authorisation*) sąvokomis. Vartotojo autentifikavimo veiksmas atliekamas prisijungimo prie sistemos metu. Vartotojas pateikia sistemai vartotojo vardą (angl. *login*) bei slaptažodį (angl. *password*). Sistema patikrina vartotojo vardo ir slaptažodžio porą, ir, jeigu pateikti duomenys teisingi, autentifikuoja vartotoją – jam leidžiama naudotis sistema.

Autorizacija – leidimas vartotojui atlikti konkretų veiksmą. Ji yra atliekama prieš atliekant bet kurį veiksmą su sistema. Dokumentų valdymo sistemose sutinkami trijų rūšių autorizacijos būdai:

- Paremto vartotojo vardu – paprasčiausias autorizacijos būdas. Kiekvienai funkcijai (dažniausiai visos sistemos lygyje) yra nurodomi vartotojai, kurie gali ta funkcija naudotis. Dažnai toks modelis naudojamas plataus vartojimo nesudėtingo funkcionalumo dokumentų valdymo sistemose, pavyzdžiui internete publikuojamuose dokumentų archyvuose: visi paprasti vartotojai, prisijungiantys prie sistemos gali atlikti dokumentų paiešką ir peržiūrą, tuo tarpu administratoriui leidžiama atlikti visus veiksmus. Šis modelis yra paprastas ir nereikalauja prieigos teisių kontrolės metaduomenų aprašymo kiekvienam dokumentui (15).
- Prieigos teisių sąrašais paremtas modelis. Šiuo atveju, su kiekvienu sistemoje saugomu dokumentu susiejamas prieigos teisių sąrašas (angl. *access control list*). Šiame sąraše nurodomi vartotojai ir jų prieigos prie aprašomo dokumento teisės. Šis metodas užtikrina lanksčiausią prieigos teisių kontrolę, tačiau turi trūkumą, kad prieigos teisės kiekvienam dokumentui reikia aprašyti kiekvienam vartotojui atskirai (16).

- Rolėmis paremtas modelis – kiekvienam vartotojui sistemoje priskiriama tam tikra rolė. Su dokumentu taip pat susiejamas prieigos teisių sąrašas, bet jame nurodomos prieigos teisės ne atskiriems vartotojams, bet rolėms. Tokiu būdu smarkai sumažinama prieigos teisių sąrašo apimtis, bet kartu sumažėja ir prieigos teisių kontrolės lankstumas (14).

Abiem paskutiniaisiais atvejais prieigos teisės turi būti aprašomos kiekvienam dokumentui, saugomam sistemoje. Tai yra atliekama prieigos teisių metaduomenų pagalba. Taip pat dauguma sistemų, realizuojančių katalogų medį loginiam dokumentų struktūros atvaizdavimui, leidžia aprašyti prieigos teises ne tik dokumentams, bet ir katalogams. Dažnai tai yra realizuojama tokiu būdu, kad kataloge esantys dokumentai ir katalogai „paveldi“ prieigos teises apibrėžtas „tėviniam“ katalogui.

Kaip ir dauguma dokumentų valdymo sistemos aspektų, aprašomų prieigos teisių aibė priklauso nuo sistemos realizacijos bei taikymo srities (pvz., nėra prasmės aprašinėti teisės modifikuoti dokumentą sistemoje, kuri nerealizuoja dokumentų redagavimo funkcijos). Žemiau pateikiama minimali būtina prieigos teisių aibė, kurios pagalba galima realizuoti prieigos teisių kontrolę universalioje dokumentų valdymo sistemoje:

- Teisė skaityti (*angl. read*) – jei leidžiama, vartotojas gali skaityti (peržiūrėti) aprašomą dokumentą. Jei aprašomas katalogas, tai vartotojui leidžiama peržiūrėti jo turinį.
- Teisė rašyti/modifikuoti (*angl. write*) – jei leidžiama, vartotojas gali rašyti į nurodytą dokumentą (t. y. jį modifikuoti ar pakeisti kitu). Jei aprašomas katalogas, tai vartotojui leidžiama tame kataloge kurti naujus objektus (katalogus arba dokumentus).
- Teisė trinti (*angl. delete*) – jei leidžiama, vartotojas gali panaikinti pasirinktą dokumentą ar katalogą.

Taikant sistemą prie konkrečios įmonės poreikių, apibrėžiamų teisių aibė gali išsiplėsti. Kaip pavyzdį būtų galima paminėti tokias vartotojo teises, kaip teisė peržiūrėti arba redaguoti metaduomenis, teisė peržiūrėti senesnes dokumento versijas ir kt. Pasirenkant prieigos teisių metaduomenų aprašymo bei apdorojimo modelį, yra labai svarbu, kad sistemai kintant, jį būtų galima papildyti naujomis prieigos teisėmis.

3.3. DVS metaduomenų apdorojimo realizacijos

3.3.1. Pasirinkimas tarp universalumo ir specializuotumo

Dokumentų valdymo sistema, kaip ir daugelis kitų programinės įrangos produktų, pasižymi ta savybe, kad įmonei jos duodama nauda priklauso nuo to, kiek tokia sistema yra specializuota ir kaip gerai atitinka įmonės poreikius. Tačiau specializuotos sistemos yra kuriamos kiekvienai įmonei atskirai pagal individualius poreikius. Kadangi tai vienetiniai produktai, tokių sistemų realizavimo kaina yra labai didelė, tokių sistemų naudojimą gali sau leisti tik didelės korporacijos.

Kita vertus, universalios dokumentų valdymo sistemos yra kuriamos masiniam vartojimui, tokių sistemų kainos yra mažesnės. Taigi, dauguma vidutinių ir mažų įmonių gali sau leisti universalias dokumentų valdymo sistemas. Tačiau kadangi tokios sistemos yra universalios – t.y. skirtos bendram naudojimui, dažnai yra sunku (ar net neįmanoma) jas modifikuoti ir pritaikyti prie įmonės veiklos srities bei vidinių procesų (18).

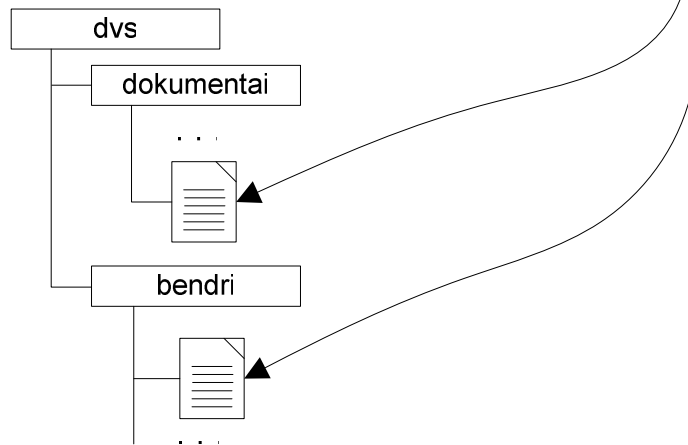
Literatūroje pateikiama daug įvairių specializuotų dokumentų valdymo sistemų metaduomenų aprašymo bei apdorojimo standartų (2, 7, 8). Taip pat aprašoma keletas universalių metaduomenų aprašymo standartų (10, 12), kurie yra skirti bendro pobūdžio dokumentų metaduomenis aprašyti bei apdoroti. Tačiau nėra apibrėžto standarto, kuris būtų ne tik universalus (t. y. būtų galima taikyti bet kokios veiklos srities dokumentams aprašyti), bet ir pakankamai lankstus, kad esant poreikiui, būtų galima aprašyti bei apdoroti ir specializuotus metaduomenis.

Skyreliuose 3.3.2 – 3.3.4 pateikiama keletas populiariausių dokumento metaduomenų aprašymo ir apdorojimo būdų. Realiose dokumentų valdymo sistemose naudojami daug sudėtingesni metaduomenų apdorojimo modeliai, dažnai apjungiantys kelių iš šių būdų savybes.

3.3.2. Nuorodų masyvo realizacija

Nuorodų masyvo (angl. *linkfarm*) realizacija – pats paprasčiausias dokumentų valdymo sistemos realizavimo būdas. Šiuo atveju naudojama metaduomenų duomenų bazė ir failų sistema, realizuojanti fizinę dokumentų saugyklos struktūrą. Duomenų bazėje išsaugomi kiekvieno dokumento metaduomenys, o taip nuoroda (angl. *hyperlink*) į dokumentą, saugomą failų sistemoje:

Autorius	Data	...	Dokumentas
Jonas	99.05.12	...	/dvs/dokumentai/dok1.doc
Petras	98.04.16	...	/dvs/bendri/ataskaita.doc
...



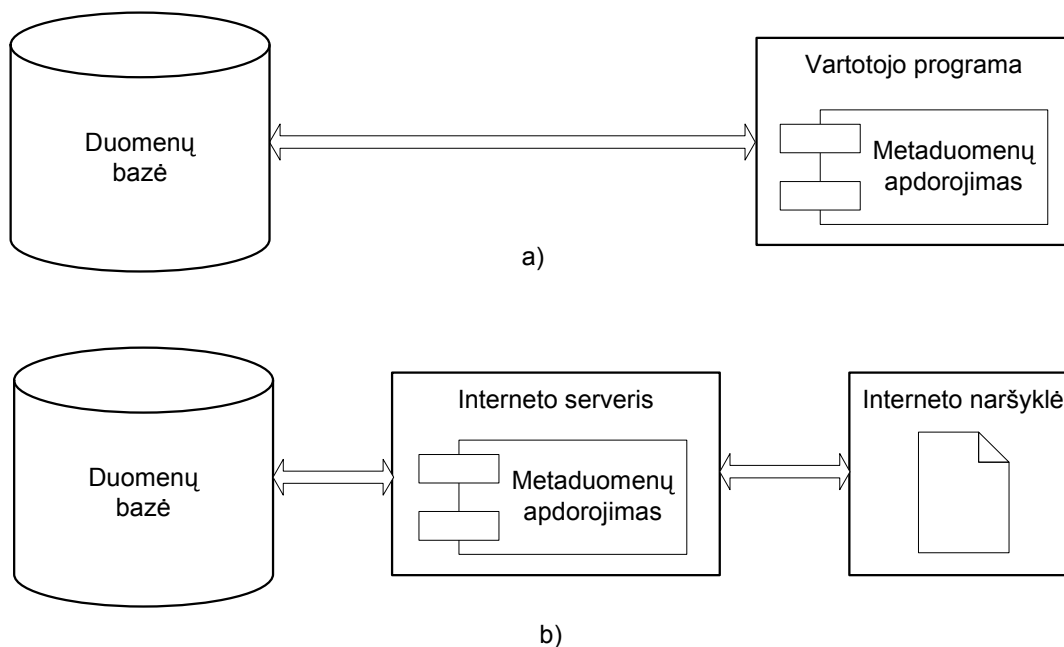
10 pav. Nuorodų masyvo realizacija

Tokio sprendimo privalumas yra tas, kad jo realizacija yra labai paprasta. Jis naudojamas dokumentų valdymo sistemose, kurios realizuoja minimalų funkcionalumą. Kaip pavyzdį būtų galima paminėti internetinį dokumentų archyvą, pasiekiamą per interneto naršyklę. Tokiu atveju, dokumentai saugomi interneto serveryje, o duomenų bazės lentelėje, kuri aprašo nuorodų masyvą, kartu su metaduomenimis išsaugoma nuoroda į dokumentą, kuri naudojama dokumento peržiūrai su interneto naršykle. Tokiose sistemose dokumentų organizacijos fizinė ir loginė struktūros sutampa, o dokumentų išgavimui naudojamas paieškos modulis, kuris atlieka dokumentų paiešką pagal jų metaduomenis. Vartotojų autentifikavimo bei autorizacijos funkcijos dažnai realizuojamos vidinių interneto serverio priemonių pagalba. Dokumentų patalpinimui ir redagavimui naudojamas atskiras programinis modulis, dažnai vadinamu administraciniu modeliu (17). Šio būdo trūkumas yra tas, kad taip realizuotos sistemos yra labai riboto funkcionalumo.

3.3.3. Realizacija panaudojant reliacinę duomenų bazę

Tai pats populiariausias metaduomenų aprašymo ir apdorojimo būdas, naudojamas daugelyje universalių vidutinio lygio (angl. *mid-level*) dokumentų valdymo sistemų. Tiek pačių dokumentų, tiek metaduomenų saugojimas vyksta duomenų bazės lentelėse. Kiekvienam metaduomenų tipui skiriama atskira duomenų bazės lentelė (ar kelios lentelės) kuri yra susiejama su dokumentu pagal pirminį raktą.

Dokumento apdorojimas realizuojamas programiškai – dokumentas ir metaduomenų informacija yra skaitoma iš duomenų bazės panaudojant SQL užklausas ir rezultatai yra pateikiami vartotojui. Tokio tipo dokumentų valdymo sistemos dažniausiai realizuojamos naudojant dviejų ar trijų lygių architektūrą:



11 pav. DVS realizacija panaudojant reliacinę duomenų bazę : a) 2-jų lygių architektūra; b) - 3-jų lygių architektūra

Dviejų lygių (angl. *two-tier*) architektūros atveju, realizuojama klientinė programa, instaliuojama vartotojo kompiuteryje. Kreipiniai į duomenų bazę bei metaduomenų apdorojimas realizuojamas būtent šioje programoje. Tokios realizacijos privalumas yra tas, kad galima realizuoti didelio funkcionalumo vartotojo sąsają – tai smarkiai palengvina vartotojo darbą su sistema. Trūkumai – įmonėje naudojama daug programinės įrangos kopijų (t. y. kiekvieno sistemos vartotojo kompiuteryje yra atskira programos kopija), kas labai apsunkina sistemos palaikymą. Taipogi lygiagrečios darbo sąlygas vartotojams reikia užtikrinti duomenų bazės valdymo sistemos priemonėmis, kas dažnai apsunkina sistemos realizaciją.

Trijų lygių architektūra realizuojama panaudojant interneto puslapius. Šiuo atveju, programiniai komponentai, atliekantys metaduomenų apdorojimą bei užtikrinantys duomenų valdymo sistemos funkcionalumą, yra realizuojami kaip interneto serverio modulis. Sąsaja su vartotoju yra užtikrinama internetinių puslapių pagalba, kuriuos vartotojas peržiūri su interneto naršykle. Taip realizuota sistema yra daug lengviau palaikoma, mažesni realizacijos kaštai, sumažinamas įmonės tinklo bei duomenų bazės serverio apkrovimas(3).

Nors dokumentų valdymo sistemos realizacija, panaudojant duomenų reliacines bazes yra labai populiarus sprendimas, ypač vidutinio lygio sistemose, jis turi keletą trūkumų, ypač kuomet kalbama apie universalias sistemas. Kadangi dokumento metaduomenys saugomi duomenų bazės lentelėse, jų struktūra (elementų aibė bei tipai) turi būti apibrėžiami sistemos realizavimo metu. Duomenų bazės schema yra projektuojama pagal apdorojamų metaduomenų struktūrą. Taip pat fiksuota metaduomenų struktūra naudojama realizuojant programinį kodą, užtikrinantį komunikavimą su duomenų baze bei dokumentų apdorojimą. Keičiant metaduomenų formatą (pvz., papildant naujais elementais ar panašiai), reikės keisti ir duomenų bazės struktūrą, bei perrašyti dalį programinio kodo. Yra kuriama daug įvairiais metodais paremtų technologijų, kurios turėtų sumažinti šio pobūdžio problemų kiekį, tačiau nei viena iš jų nėra iki galo išbaigta, kad visiškai jas panaikintų.

3.3.4. Sudėtingų aukšto lygio sistemų realizacijos specifiška

Didelėse įmonėse – korporacijose – yra naudojamos sudėtingos dokumentų valdymo sistemos. Jos pasižymi sudėtinga architektūra bei integracija su kitomis korporacijos informacinėmis sistemomis, dažnai yra paskirstytos, realizuojamos panaudojant sudėtingas komponentines programinės įrangos kūrimo technologijas, tokias kaip EJB (angl. *Enterprise Java Beans*). Dažnai tokios sistemos kūrimo metu dalis komponentų yra pakartotinai panaudojama iš jau egzistuojančių informacinių sistemų, o dalis įsigyjama iš atitinkamą funkcionalumą turinčių komponentų kūrimu užsiimančių informacinių technologijų įmonių. Tokios sistemos visada yra kuriamos pagal konkrečios įmonės reikalavimus ir veiklos srities specifišką, todėl didesniąją sistemos kūrimo laiko dalį užima taikymo srities specifikos bei reikalavimų, keliamų sistemai, analizė.

Kadangi tokios dokumentų valdymo sistemos pasižymi aukštu specializacijos lygiu, tai naudojamų metaduomenų formatai priklauso nuo taikymo srities. Dažniausiai yra naudojami specialiai tai sričiai sukurti metaduomenų apdorojimo komponentai, įsigijami iš toje taikymo srityje besispecializuojančių komponentinės programinės įrangos gamintojų. Tokie komponentai dažnai turi metaduomenų pritaikymo (angl. *customizing*) prie konkrečios sistemos poreikių galimybę.

4. METADUOMENŲ APDOROJIMAS XML TECHNOLOGIJŲ PAGALBA

4.1. Naudojamų XML technologijų aprašymas

4.1.1. XML duomenų aprašymo formatas

XML (angl. *eXtensible Markup Language*) yra universalus duomenų aprašymo formatų, skirtų aprašyti tekstinio-hierarchinio pobūdžio duomenų struktūras, aprašymo standartas. Pagrindinė XML standarto (19) savybė yra tai, kad juo yra apibrėžiama tik formato aprašymo sintaksė ir taisyklės, todėl panaudojant XML galima apibrėžti duomenų aprašymo formatą praktiškai bet kuriai taikymo sričiai.

Duomenų aprašymas XML formatu suteikia šiuos privalumus:

- Aukštas suderinamumo lygis – XML formatu aprašytus duomenis galima apdoroti bet kokioje sistemoje, palaikančioje XML standartą.
- Programinės apdorojimo priemonės – standartinės programinės bibliotekos skaitymui, rašymui, validacijai, transformavimui iš vieno formato į kitą.
- Galimybė automatiškai validuoti aprašomų duomenų sintaksės ir semantikos korektiškumą.
- *De facto* standartas apsikeitimui informacija tarp skirtingų sistemų.
- XML tipo duomenys lengvai skaitomi ir suprantami.

Aprašant XML standartu paremtą formatą, galima pasinaudoti technologija, vadinama XML Schema (28). Jos pagalba galima formaliai aprašyti formatą: nurodyti galimus XML dokumento elementus bei jų reikšmes, hierarchijos tvarką ir kt.

4.1.2. XPath ir XSLT technologijos

XPath (angl. *XML Path Language*) – tai kalba, skirta identifikuoti ir adresuoti atskiras XML dokumento dalis. Tai atliekama pagal XPath standartą (20) aprašoma išraiška. Šios išraiškos rezultatas gali būti tuščia aibė, vienas ar daugiau XML dokumento elementų arba elemento atributų. XPath technologija naudojama XSLT bei XML duomenų bazėse.

XSLT (angl. *eXtensible Stylesheet Language Transformations*) – tai XML formatu aprašoma kalba, skirta transformuoti XML aprašytus duomenis iš vieno formato į kitą. Transformacijos metu gali būti sukuriami nauji arba pašalinami egzistuojantys dokumento fragmentai, „iškerpama“ tam tikra dokumento dalis ir kitos operacijos (21). Transformavimo taisyklės yra aprašomos deklaratyviai, panaudojant XSLT šablonus (angl. *XSLT stylesheets*) ir yra taikomos transformuojamo XML dokumento turiniui panaudojant XPath išraiškas.

4.1.3. XML apdorojimo konvejeriai

XML apdorojimo konvejeris (angl. *pipeline*) – tai toks duomenų, aprašytų XML formatu, apdorojimo būdas, kuomet iš eilės atliekama keletas apdorojimo žingsnių (XSLT transformacijų arba modifikavimo programinėmis priemonėmis) ir vieno apdorojimo žingsnio rezultatas yra perduodamas sekančiam žingsniui kaip pradiniai duomenys (23). Kiekvienas programinis modulis, atskirame žingsnyje apdorojantis XML duomenis, vadinamas XML procesoriumi (angl. *processor*). Konvejerinio XML apdorojimo pagrindinis pranašumas yra tas, kad vieną kartą sudarytą konvejerį galima naudoti daugelio dokumentų apdorojimui. Taip pat, kadangi apibrėžiamos XML procesoriaus įėjimo ir išėjimo sąsajos, nusakančios kokia tvarka turi būti perduodami įėjimo duomenys, ir kokia tvarka nuskaitomi išėjimo duomenys (22), konvejerio konfigūraciją galima lengvai keisti, pridėdant arba pašalinant iš jo skirtingą funkcionalumą realizuojančius XML procesorius.

4.1.4. XML duomenų bazės

Literatūroje (26) pateikiamos trys XML duomenų bazių sistemų klasės:

- Grynos (angl. *native*) XML duomenų bazių sistemos – suprojektuotos ir naudojamos saugoti ir apdoroti XML formato duomenis.
- XML-palaikančios duomenų bazių sistemos – duomenų bazės, saugančios informaciją ne XML formatu, tačiau suteikiančios sąsajas prieigai prie tų duomenų kaip prie XML duomenų.
- Hibridinės duomenų bazių sistemos – prieiga suteikiama tiek per XML, tiek per kitokio formato sąsajas.

Pirmojo tipo (grynos) XML duomenų bazių sistemos yra labai naudingos, kuomet reikia saugoti ir apdoroti didelį kiekį XML formato duomenų. Naudojant XML duomenų bazę vietoje reliacinės, sutaupoma resursų, kurie būtų naudojami kiekvienos kreipties metu informacijos tarp XML ir reliacinės schemos konvertavimui. Skirtingai nuo reliacinių duomenų bazių, kur informacija saugoma lentelėse, XML duomenų bazėje informacija saugoma kaip XML dokumentų aibė, todėl tokios duomenų aibės dar vadinamos kolekcijomis.

XML duomenų bazių sistemos suteikia galimybę atlikti paiešką vienos duomenų bazės ribose, naudojant XPath išraiškas ieškomo XML dokumento aprašymui (25). Sparčiam XML dokumento išgavimui iš duomenų bazės gali būti naudojamas raktas, kurį sugeneruoja ir grąžina XML duomenų bazių sistema, patalpinant į ją dokumentą pirmą kartą. Talpinant dokumentą į sistemą, galima nurodyti ir savo raktą, tik yra reikalaujama, kad toje pačioje

kolekcijoje nebūtų dviejų dokumentų su tokiais pačiais raktais. XML dokumentų talpinimui į duomenų bazę naudojama XUpdate kalba (27). Kai kuriose sistemose yra galimybė kurti hierarchines duomenų bazines (XML duomenų bazė kitoje, „tėvinėje“ XML duomenų bazėje).

4.2. Dokumento metaduomenų aprašymas XML formatu

4.2.1. Metaduomenų aprašymas

Siekiant išvengti galimo terminų „dokumentas“ (dokumentų valdymo sistemoje saugomas dokumentas) bei „XML dokumentas“ (XML duomenų bazėje saugomas metaduomenų įrašas), tolimesnėje darbo dalyje vietoj pastarojo vartosime terminą „XML įrašas“.

Kiekvieno metaduomenų, susietų su sistemoje apdorojamu dokumentu, tipo (žr. skyrelį 3.2.1) metaduomenims aprašyti bus skiriama po atskirą XML įrašą. Taigi, reikia apibrėžti šiuos XML formatus:

- pirminių dokumento metaduomenų aprašas XML formatu;
- prieigos teisių metaduomenų aprašas XML formatu;
- versijų kontrolės metaduomenų aprašas XML formatu;
- loginės struktūros organizavimo metaduomenų aprašas XML formatu.

Kiekvieno tipo metaduomenys saugomi atskirame XML įrašė, jie turi būti kažkaip susiejami tarpusavyje bei su aprašomu dokumentu. Tam įvedama unikalios identifikatoriaus (angl. *unique identifier*) sąvoka. Kiekvienam į sistemą talpinamam dokumentui yra sugeneruojamas naujas unikalios identifikatorius. Šis identifikatorius įrašomas į atitinkamą kiekvieno XML įrašo šakninio elemento atributą. Tokiu būdu, skirtingų tipų metaduomenų įrašai, turintys tą pačią identifikatoriaus reikšmę, aprašo tą patį objektą.

4.2.2. Pirminių dokumento metaduomenų aprašymas

Pirminių dokumento metaduomenų aprašymui XML formatu galimos dvi alternatyvos:

- galima panaudoti kažkurį iš jau egzistuojančių standartų, pavyzdžiui Dublin Core Metadata (12);
- galima suprojektuoti ir naudoti naują formatą pirminių metaduomenų apdorojimui.

Pirmas variantas yra paprastesnis, kadangi nereikia gaišti laiko metaduomenų aprašymo standartui kurti. Tačiau toks sprendimas gali sukelti dviejų rūšių problemas:

- Sistema perkraunama bereikalinga informacija. Jeigu naudojamas standartas yra per platus ir apibrėžia daug privalomų metaduomenų elementų (pvz., Dublin

Core standartas numato 13 metaduomenų elementų), o dalis jų nėra aktualūs realizuojamai sistemai, tokiu atveju vartotojas bus priverstas įvedinėti bereikalingus duomenis.

- Standarto uždarumas. Kai kurie metaduomenų standartai yra uždari – t. y. elementų aibė yra griežtai specifikuota, ir nėra galimybės jos papildyti naujo tipo elementais. Šiuo atveju gaunama priešinga situacija – apribojamas sistemos lankstumas bei išplėtimo galimybės.

Todėl vidiniam sistemos naudojimui, geriau apibrėžti naują pirminių metaduomenų aprašymo XML formatu standartą. Norint pasiekti didžiausią lankstumą, geriausia apibrėžti minimalią aibę privalomų metaduomenų elementų, bei suprojektuoti metaduomenų aprašo išplėtimo mechanizmą, kuris būtų naudojamas sistemos pritaikymo vienoje ar kitoje srityje metu. Aprašysime pirminių metaduomenų XML įrašą pagal 3.2.2 skyrelyje pateiktą elementų aibę:

Lentelė 1

Pavyzdinis metaduomenų XML elementų aprašas

Metaduomenų elementas	XML elemento pavadinimas	Galimos reikšmės
Pavadinimas	title	Tekstas
Autorius	author	Tekstas
Tema	subject	Tekstas
Aprašymas	description	Tekstas
Sukūrimo data	date-created	Data
Patalpinimo data	date-uploaded	Data
Modifikavimo data	date-modified	Data
Dokumento tipas	type	Reikšmė iš galimų reikšmių aibės
Dokumento formatas	format	Reikšmė iš galimų reikšmių aibės
Raktiniai žodžiai	keywords, keyword	Kiekvienam raktiniam žodžiui po elementą
Specifiniai taikymo sričiai	extend	-

XML elementas „extend“ – naudojamas metaduomenų aprašymo schemos praplėtimui. Jo atributas „name“ nusako norimo į metaduomenis įtraukti elemento pavadinimą, o XML elemento reikšmė – metaduomenų elemento reikšmę. Pateiksime pagal apibrėžtą schemą aprašytą pavyzdinį pirminių metaduomenų įrašą XML formatu:


```

<?xml version="1.0" encoding="UTF-8"?>
<metadata id="200404151114235000">
  <!-- dokumento autorius -->
  <title>Pavyzdinis dokumentas</title>
  <!-- dokumento autorius -->
  <author>Rimantas Žukaitis</author>
  <!-- dokumento tema -->
  <subject>Dokumentų valdymas</subject>
  <!-- dokumento aprašymas -->
  <description>Pavyzdinis dokumentas ir jo metaduomenys</description>
  <!-- dokumento sukūrimo data -->
  <date-created>2004-04-15 11:10:15</date-created>
  <!-- dokumento patalpinimo į DVS data -->
  <date-uploaded>2004-04-15 11:14:23</date-uploaded>
  <!-- dokumento paskutinio modifikavimo data -->
  <date-modified>2004-05-10 23:45:33</date-modified>
  <!-- dokumento tipas -->
  <type>report</type>
  <!-- dokumento failo formatas -->
  <format>text/xml</format>
  <!-- dokumento raktiniai žodžiai -->
  <keywords>
    <keyword>dokumentas</keyword>
    <keyword>pavyzdys</keyword>
  </keywords>
  <!-- dokumento galiojimo pabaigos data -->
  <extend name="expires">2004-06-01 00:00:00</extend>
</metadata>

```

Pavyzdyje parodyta, kaip pasinaudojant metaduomenų aprašo išplėtimo mechanizmu, prie bazinių pirminių metaduomenų elementų (aprašytų skyrelyje 3.2.2) buvo pridėtas dar vienas, nurodantis dokumento galiojimo trukmės pabaigą.

4.2.3. Prieigos teisių metaduomenų aprašymas

Prieigos teisės nusakomos dvejomis duomenų grupėmis – duomenys apie vartotojus, naudojami vartotojų autentifikavimui bei dokumento prieigos teisių metaduomenys, naudojami autorizacijai.

Vartotojų teisių valdymo supaprastinimui įvesime vartotojų grupės sąvoką – vartotojas gali priklausyti vienai ar kelioms vartotojų grupėms. Nurodant prieigos prie dokumento teises, galima jas apibrėžti grupei – taip jos suteikiamos visiems vartotojams, esantiems grupėje. Toks sprendimas smarkiai sumažina prieigos teisių kontrolės sąrašo apimtį – tuo pačiu ir metaduomenų, saugomų kartu su kiekvienu dokumentu kiekį.

Sistemos vartotojai ir vartotojų grupės irgi gali būti aprašomi XML įrašų XML duomenų bazėje pagalba. XML įrašo, aprašančio sistemos vartotoją, pavyzdys:

```

<?xml version="1.0"?>
<principal type="user">
  <!-- vartotojo vardas ir pavardė -->
  <name>Rimantas Žukaitis</name>

```

```

<!-- vartotojo prisijungimo vardas -->
<login>rimaz</login>
<!-- vartotojo slaptažodis -->
<password>drowssap</password>
<!-- varotojų grupės, kurioms priklauso varotojas -->
<groups>
  <group id="users"/>
</groups>
</principal>

```

XML įrašo, aprašančio sistemos vartotojų grupę, pavyzdys:

```

<?xml version="1.0"?>
<principal type="group">
  <!-- vartotojų grupės pavadinimas -->
  <name>users</name>
  <!-- vartotojų grupės aprašymas -->
  <description>Sistemos vartotojai</description>
</principal>

```

Vartotojų prieigos teisių sąrašas formuojamas „draudimo pagal nutylėjimą“ principu. Jeigu konkretus vartotojas (o taip pat nei viena iš grupių, kurioms vartotojas priklauso) iš viso nėra paminėtas prieigos teisių sąrašė, tai jam draudžiamos visos operacijos su sąrašo aprašomu dokumentu. Jei konkreti teisė vartotojui nėra paminėta, ji pagal nutylėjimą taip pat draudžiama. Tokia saugumo politika sumažina neautorizuotos prieigos tikimybę, kuomet vartotojas „per klaidą“ pamiršta apriboti prieigos teises naujai sukurtam dokumentui. Taip pat tokia saugumo politika padidina prieigos teisių modelio lankstumą – įvedus į sistemą naujų prieigos teisių tipų (pvz., „redaguoti metaduomenis“ ar pan.) prieigos teisių modelis funkcionuos korektiškai, nes neatsiras „neuždraustų prieigos teisių“.

Kiekvienam vartotojui (ar vartotojų grupei), kuriam yra nurodomos prieigos teisės, skiriama po vieną XML elementą prieigos teisių kontrolės sąrašė. Jame nurodomas vartotojo vardas, ir tam vartotojui suteikiamos teisės.

Jeigu vartotojas priklauso kelioms grupėms, kurioms yra apibrėžiamos prieigos teisės, tuomet tam vartotojui naudojama visų grupių teisių sąjunga. Tačiau jeigu nurodomos teisės ir vartotojui, ir grupei, kuriai jis priklauso, tam vartotojui autorizuoti yra naudojamos jam nurodytos prieigos teisės, ignoruojant visas vartotojų grupes. Pateikiamas minimalaus prieigos teisių metaduomenų įrašo pavyzdys:

```

<?xml version="1.0"?>
<security>
  <acl>
    <!-- vartotojų grupei "users" leidžiama tik skaityti -->
    <ace principal="users" read="true"/>
    <!-- vartotojui "rimaz" leidžiama ir skaityti, ir rašyti -->
    <ace principal="rimaz" read="true" write="true"/>
  </acl>
</security>

```

4.2.4. Loginės organizacijos metaduomenų aprašymas

Loginė struktūra organizuojama dviem aspektais – įvedama katalogo sąvoka bei nurodomi sąryšiai tarp katalogų ir dokumentų.

Katalogo aprašymas metaduomenimis yra beveik identiškas dokumento aprašymui. Katalogui taip pat aprašomas prieigos teisių sąrašas, gali būti priskiriami pirminiai metaduomenys. Tačiau katalogui nėra atitiktens dokumentų saugykloje, taip pat nesaugomi versijų kontrolės metaduomenys.

Sąryšiai tarp katalogų ir dokumentų yra aprašomi dvigubai – katalogo metaduomenyse nurodoma, kokie dokumentai (ir katalogai) yra jo viduje, o dokumento metaduomenyse taip pat nurodoma, kokiam katalogui (ar keliems katalogams) jis priklauso. Dvigubo sąryšio sudarymas labai paspartina metaduomenų apdorojimą.

Taip pat dokumento loginės organizacijos metaduomenyse gali būti nurodomi, su juo susiję dokumentai.

Sąsajos tiek tarp katalogų ir dokumentų, tiek tarp dokumentų yra realizuojamos nurodant siejamo objekto unikalų identifikatorių. Žemiau pateikiamas XML įrašo, aprašančio katalogą, pavyzdys:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- katalogas su id: -->
<folder id="20020502012341100">
  <nodes>
    <!-- katalogo viduje saugomų objektų id -->
    <node id="20020506142000000"/>
    <node id="20020506142010000"/>
    <node id="20020615111015000"/>
    <node id="20020615111025000"/>
  </nodes>
  <parents>
  </parents>
</folder>
```

Dokumento loginės struktūros organizacijos metaduomenis aprašantis XML įrašas:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- dokumento id: -->
<doc id="20020506142010000">
  <parents>
    <!-- tevinio katalogo id: -->
    <parent id="20020502012341100"/>
  </parents>
  <links>
    <!-- nuoroda į kitą dokumentą -->
    <link id="200404151114235000"/>
  </links>
</doc>
```

4.2.5. Versijų kontrolės metaduomenų aprašymas

Versijų kontrolė atliekama kopijų išsaugojimo būdu. Kiekvieną kartą kai vartotojas po dokumento redagavimo išsaugo pakeitimus, sena dokumento versija yra nukopijuojama į atskirą saugyklą. Jai yra suteikiamas naujas identifikatorius, unikalus senų versijų saugyklos ribose. Tokiu būdu, pagrindinėje dokumentų saugykloje visa laiką yra saugoma naujausia dokumento versija, o visos senesnės versijos, pradedant nuo pat pirminės dokumento versijos, patalpintos į sistemą, yra saugomos senų versijų saugykloje.

Dokumento versijų kontrolės metaduomenyse yra saugomi įrašai apie kiekvieną versijos pasikeitimą – koks vartotojas atliko pakeitimus, kada jie buvo atlikti, taip pat papildomas vartotojo pastabas apie padarytus pakeitimus. Kartu su įrašu apie pakeitimus, saugoma nuoroda į senų versijų saugyklą. Nuorodos sukuriamos panaudojant kiekvienai versijai suteiktą identifikatorių.

Dokumento versijų kontrolės metaduomenyse taip pat saugoma papildomi duomenys apie dokumentą – einamosios versijos numeris, taip pat elementas aprašantis, ar dokumentas konkrečiu momentu yra redaguojamas. Pateikiamas XML įrašo, aprašančio versijų kontrolės metaduomenis, pavyzdys:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- documento id -->
<versions id="200404151114235000">

  <!-- dabartinė dokumento versija -->
  <current version="31">
    <!-- dokumentas yra užregistruotas redagavimui -->
    <editing value="true">
      <!-- redaguojančio vartotojo vardas -->
      <user>rimaz</user>
      <!-- redagavimo pradžios laikas -->
      <since>2004-04-18 16:30:13</since>
    </editing>
  </current>
  <!-- įrašai apie senesnes versijas -->
  <history>
    <!-- versijos numeris -->
    <version value="30">
      <!-- versijos autorius -->
      <author>rimaz</author>
      <!-- kada padaryti pakeitimai -->
      <date>2004-04-18 09:20:18</date>
      <!-- aprašymas -->
      <notes>smulkūs pakeitimai</notes>
      <!-- nuoroda į senų versijų db -->
      <link id="20040415111423500030"/>
    </version>
    <version> <!-- .... --> </version>
    <!-- ..... -->
  </history>
</versions>
```

4.3. Dokumentų ir metaduomenų saugojimas XML formatu

4.3.1. Dokumentų saugykla

Kaip buvo minėta poskyryje 2.4, įmonėje apdorojamus dokumentus galima suskirstyti į dvi kategorijas: tekstinės informacijos dokumentus bei skaitmeninius dokumentus. Šių dviejų kategorijų dokumentai yra apdorojami skirtingai:

- Struktūrizuota tekstinė informacija yra aprašoma XML formatu. Dauguma šiuolaikinių plataus vartojimo programų (pvz., MS Office 2003 ar StarOffice biuro programų paketai) turi galimybę išsaugoti duomenis XML formatu. XML formato dokumentai yra saugojami atskiroje dokumentams skirtoje XML duomenų bazėje (kolekcijoje). Saugant dokumentą į XML duomenų bazę, kaip raktas nurodomas jo unikalus identifikatorius.
- Dvejetainiai dokumentai, kurių neįmanoma perversi į XML formatą, saugomi atskiroje saugykloje. Dvejetainių dokumentų saugykla realizuojama failų sistemos katalogų medžio principu. Į šį katalogų medį dvejetainiai dokumentai taip pat patalpinami naudojant sistemos sugeneruotą identifikatorių. Dviejų hierarchijos lygių medis formuojamas maišos (angl. *hashing*) metodu, vietoj rakto naudojant identifikatorių. Taip pasiekiamas beveik tolygus dokumentų pasiskirstymas po katalogus, taip paspartinant jų paieškos ir nuskaitymo laiką.

4.3.2. Metaduomenų saugykla

Metaduomenų saugykla realizuojama panaudojant penkias skirtingas XML duomenų bases. Kiekvienam skirtingam metaduomenų tipui panaudojama atskira XML duomenų bazė:

- Dokumento pirminių metaduomenų XML duomenų bazė.
- Loginės struktūros organizavimo metaduomenų XML duomenų bazė.
- Versijų kontrolės metaduomenų XML duomenų bazė.
- Prieigos teisių kontrolės metaduomenų XML duomenų bazė.
- Vartotojus ir vartotojų grupes aprašanti XML duomenų bazė.

Toks metaduomenų išskirstymas į atskiras kolekcijas paspartina jų nuskaitymą ir apdorojimą. Tačiau didžiausias tokio sprendimo privalumas yra tas, kad kiekvieno tipo metaduomenų XML įrašus galima saugoti į atitinkamą XML duomenų bazę panaudojant tą patį raktą – dokumento, kuri aprašo metaduomenys, unikalų identifikatorių. Taip visi metaduomenys susiejami su dokumentu, kurį jie aprašo, nenaudojant daugiau jokių programinių priemonių.

Tiesa, vartotojų ir vartotojų grupių įrašus aprašanti kolekcija realizuojama šiek tiek kitaip – čia vietoj rakto naudojamas sistemos vartotojo vardas ir vartotojų grupės vardas vartotojams ir vartotojų grupėms atitinkamai.

Taip pat naudojama dar viena XML duomenų bazė XML formato dokumentų senoms versijoms saugoti, bei atskiras failų sistemos katalogų medis dvejetainių dokumentų senoms versijoms saugoti. Šios senų versijų saugyklos realizuojamos analogiškai pagrindinėms, tik vietoj unikalios dokumento identifikatoriaus naudojamas versijos identifikatorius, suteikiamas kiekvienai versijai versijų valdymo metaduomenų apdorojimo metu.

4.4. Metaduomenų apdorojimo modelis

4.4.1. XML apdorojimo konvejerio komponentai

Siūlomas konceptualus metaduomenų apdorojimo dokumentų valdymo sistemoje modelis remiasi konvejerinio XML formatu aprašytų duomenų (XML srautų) apdorojimo principu.

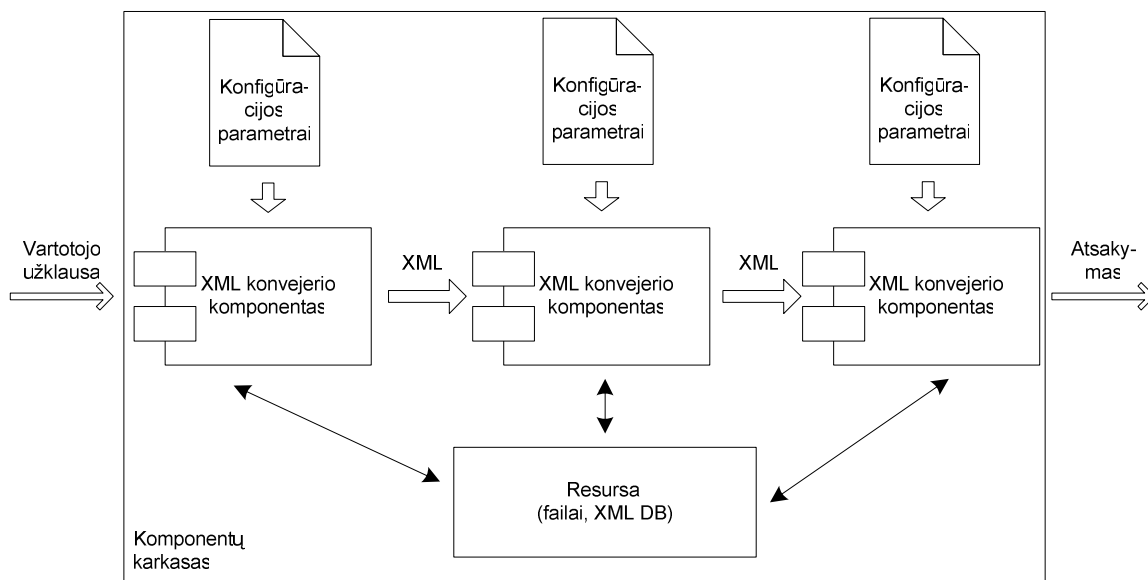
Kaip minėta skyrelyje 4.1.3 XML apdorojimo konvejeris susideda iš XML procesorių – programinių modulių, gebančių apdoroti XML duomenų srautą. XML procesoriai realizuojami kaip programinio komponentų karkaso (angl. *component framework*) komponentai. XML apdorojimo konvejeris funkcionuoja šio karkaso kontekste.

Siūlomo metaduomenų apdorojimo modeliui realizuoti reikia tokio funkcionalumo komponentų:

- Užklauso XML duomenų bazėje atlikimo komponentas. Kaip parametras šiam komponentui paduodama XPath išraiška, nusakanti užklausa. Komponentas gražina visų XML įrašų, tenkinančių duotą išraišką, sąrašą.
- XML įrašo skaitymo iš XML duomenų bazės komponentas. Kaip parametras šiam komponentui perduodama rakto reikšmė.
- XML įrašo įrašymo į XML duomenų bazę komponentas – įrašo apdorojamą XML srautą į XML duomenų bazę, panaudodamas parametru perduotą rakto reikšmę.
- XML apdorojimo XSLT transformacijos pagalba komponentas.
- Rezultatų persiuntimo vartotojui komponentas – nurodo XML konvejerio pabaigą. Konvejerio apdorotas XML srautas konvertuojamas į nurodytą formatą (HTML, PDF ar kitą) ir persiunčiamas vartotojui.
- Kito XML konvejerio kvietimo komponentas – aktyvuoja kitą XML apdorojimo konvejerį.

- Sąlygos tikrinimo komponentas – naudojamas konvejerio išsišakojimui į du, priklausomai nuo tikrinamos sąlygos. Jei sąlyga tenkinama, vykdoma pagrindinė šaka, priešingu atveju – papildoma.
- Dvejetainių resursų apdorojimo komponentas – skirtas persiųsti vartotojui informaciją, kuri nėra aprašoma XML formatu.
- Vartotojo užklauso duomenų apdorojimo komponentas – generuoja XML srautą iš vartotojo užklauso parametrų (persiunčiamų HTML formų pagalba).

Komponentų karkasas realizuoja papildomą sistemos funkcionalumą – atlieka vartotojų užklauso apdorojimą, aktyvuodamas atitinkamą XML konvejerį, taip pat persiunčia vartotojui (kaip atsakymą į užklauso) konvejerio sugeneruotą rezultatą. Taip pat užtikrina komponentų funkcionavimą, suteikdamas jiems būtinus resursus – prieigą prie failų, duomenų bazių ir kt. Pavyzdinio konvejerio veikimo schema pavaizduota paveiksle 12.



12 pav. XML konvejerio funkcionavimas

Svarbiausia komponentų karkaso funkcija – jame aprašomi XML konvejeriai. Nurodoma, kokie ir kokia tvarka aktyvuojami komponentai sudaro konvejerį, kiekvienam komponentui nurodomi reikiami parametrai. Tai daroma konfigūracinių failų pagalba ir nereikalauja programinio kodo perrašymo ar papildymo.

4.4.2. Metaduomenų apdorojimas

Dokumento metaduomenų apdorojimas atliekamas XML konvejerių pagalba. Kadangi tam tikro tipo metaduomenų apdorojimas dažnai būna tik vartotojo užklauso apdorojimo

proceso dalis (pvz., vartotojo prieigos teisių tikrinimas užklauskos peržiūrėti kažkurį dokumentą apdorojimo metu), metaduomenų apdorojimas realizuojamas tik tam tikroje konvejerio dalyje arba panaudojant papildomą konvejerį, kuris yra aktyvuojamas iš pagrindinio.

Su metaduomenimis atliekamos šios operacijos:

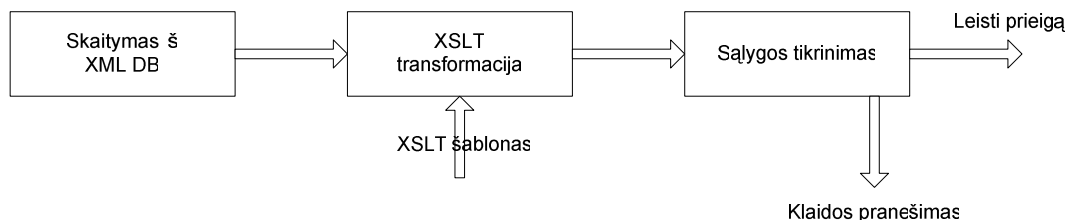
- Skaitymo – šios operacijos metu metaduomenys nuskaitymi iš XML duomenų bazės ir perduodami į tolimesnius konvejerio komponentus apdorojimui. Nuskaitymui naudojamas XML įrašo skaitymo iš duomenų bazės komponentas, kuriam parametru pagalba perduodamas atitinkamo įrašo raktas bei kolekcijos pavadinimas.
- Tam tikros reikšmės filtravimo ir tikrinimo – iš viso XML įrašo, aprašančio metaduomenis, išrenkama tik konkretaus elemento reikšmė. Tai atliekama naudojant XSLT transformacijos komponentą ir specialiai tam tikslui parašytą XSLT šablona, kuris iš visos XML dokumento struktūros „iškerpa“ tiktai reikiamą elementą. Pavyzdžiui, iš skyrelyje 4.2.2 pateikto pirminių dokumento metaduomenų autoriaus elementą „iškerpantis“ šablonas atrodytų taip:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version=""
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/metadata/author">
    <xsl:copy-of select="."/>
  </xsl:template>
</xsl:stylesheet>
```

- Įrašymo – į XML duomenų bazę įrašomas naujas XML įrašas. Tai atliekama XML įrašo įrašymo į XML duomenų bazę komponento pagalba. Parametru pagalba komponentui perduodama rakto reikšmė bei XML duomenų bazės, į kurią bus įrašyti metaduomenys, pavadinimas.
- Atnaujinimo – jau egzistuojantis metaduomenų įrašas XML duomenų bazėje atnaujinamas, įrašant naujas reikšmes. Operacija atliekama taip pat kaip ir įrašymo operacija, tik įrašymo į XML duomenų bazę komponentui perduodamas papildomas parametras, nurodantis, kad esamą įrašą reikia pakeisti nauju.
- Pašalinimo – metaduomenų XML įrašas ištrinamas iš XML duomenų bazės. Tam atlikti taip pat naudojamas įrašymo į XML duomenų bazę komponentas, kuriam parametru pagalba nurodomas raktas bei nurodoma, kad reikia atlikti trynimo veiksmą.

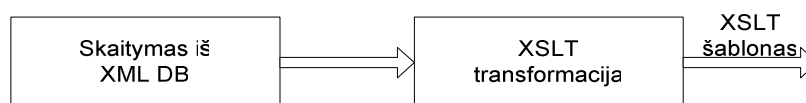
4.4.3. Prieigos teisių apdorojimas

Konvejerio, apdorojančio prieigos teisių informaciją, schema:



13 pav. Prieigos teisių informacijos apdorojimas

Visam konvejeriui, prieš jį aktyvuojant, karkasas nustato 3 parametrus – dokumento, kuriam tikrinamos prieigos teisės, identifikatorius; vartotojo, kuriam tikrinamos teisės, vardas bei prieigos teisės, kuri yra tikrinama, pavadinimas. Skaitymo iš XML DB metu iš dokumento prieigos teisių sąrašą aprašančių metaduomenų sugeneruojamas XML srautas, kuris perduodamas XSLT transformaciją atliekančiam komponentui. XSLT komponentas naudoja XSLT šabloną sugeneruojamą iš vartotojo metaduomenų atskiru XML konvejeriu:

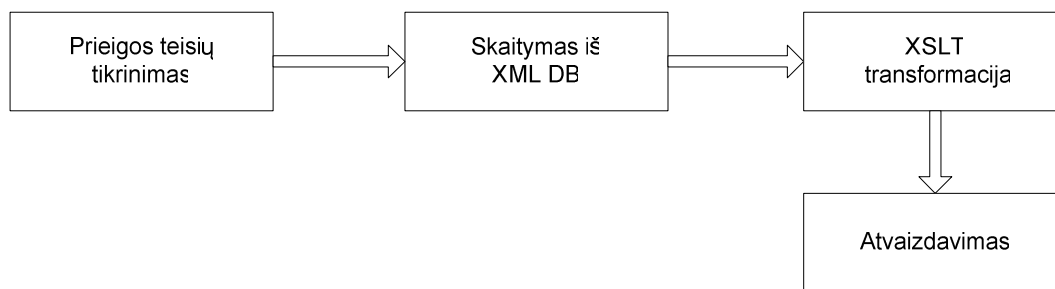


14 pav. Vartotojo grupių sąrašo apdorojimo XML konvejeris

Šis pagalbinis konvejeris apdoroja vartotojo gupių sąrašą, nuskaitytą iš XML DB, ir pagal tai, kokioms vartotojų grupėms vartotojas priklauso, sugeneruoja XSLT šabloną. Šis šablonas naudojamas transformuoti prieigos teisių sąrašo informacijai tokiu būdu, kad rezultatų XML sraute lieka prieigos teises aprašančių elementų tik tuo atveju, jeigu jos yra leidžiamos vartotojui, kurio vardas buvo perduotas į konvejerį. Sąlygos tikrinimo komponentas patikrina, ar liko nors vienas prieigos teisių sąrašo elementas. Jei taip – konvejeris grąžina netuščią sąrašą, tuo pranešdamas, kad vartotojas buvo autorizotas sėkmingai. Priešingu atveju konvejerio vykdymas stabdomas ir pranešama apie autorizacijos klaidą.

4.4.4. Dokumento ir jo metaduomenų peržiūra

Dokumento peržiūros funkciją realizuoja toks XML apdorojimo konvejeris:



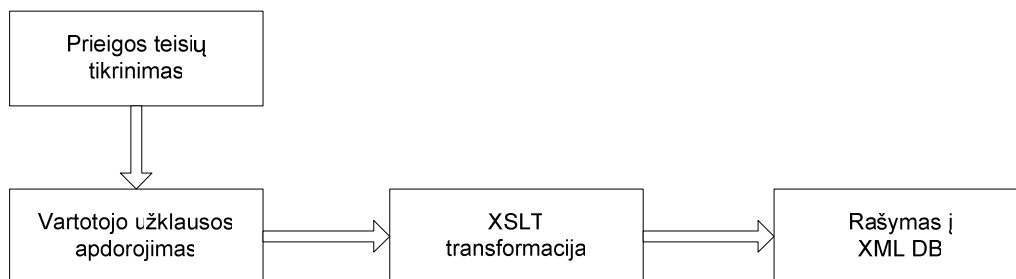
15 pav. Dokumento peržiūros XML apdorojimo konvejeris

Prieš vykdant konvejerį, karkasas nustato vieną parametą – dokumento identifikatorių. Pirmu žingsniu vykdomas prieigos teisių patikrinimas, jau aprašytas skyrelyje 4.4.3. Jei vartotojui peržiūros veiksmas buvo autorizuotas, iš XML DB, kurioje saugomi dokumentai, pagal dokumento identifikatorių skaitomas dokumento turinys. Trečiu žingsniu atliekama XSL transformacija – dokumentas paruošiamas atvaizdavimui. Paskutiniame žingsnyje dokumentas yra konvertuojamas į atvaizdavimo formatą (pvz., į HTML iš XHTML) ir persiunčiamas vartotojui.

Analogiški konvejeriai realizuoja ir metaduomenų peržiūrą – tik tuomet antrame žingsnyje skaitoma ne iš dokumentų XML DB, bet iš atitinkamo tipo metaduomenų XML DB. Taip pat, tokio paties pobūdžio konvejeriu yra realizuojama dokumento versijų peržiūra – tokiu atveju konvejerio parametras yra ne unikalus dokumento identifikatorius, o versijų saugyklos identifikatorius. Dokumento versija yra nuskaitoma iš versijų saugyklos XML DB.

4.4.5. Dokumento ir jo metaduomenų patalpinimas

Kaip ir skaitymo atveju, ta pati schema galioja tiek dokumento, tiek jo metaduomenų patalpinimui. Nors vartotojui dokumento ir jo metaduomenų patalpinimas atrodo kaip vienas veiksmas, jis realizuojamas keliomis atskiromis užklausomis: pirma patalpinamas dokumentas, tuomet įvedami jo metaduomenys, nustatomos prieigos teisės ir t. t. XML apdorojimo konvejerio principinė schema visais šiais atvejais ta pati, skiriasi tik XML DB į kurias įrašinėjama informacija:



16 pav. Dokumento ar jo metaduomenų patalpinimo XML apdorojimo konvejeris

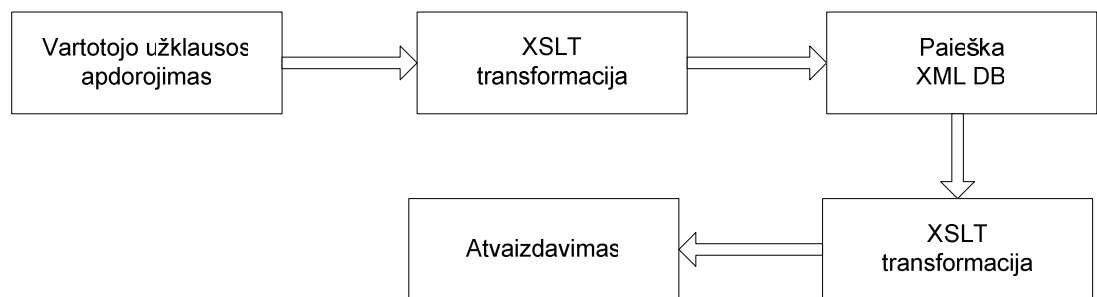
Pirmiausia apdorojami katalogo, į kurį talpinamas dokumentas, prieigos teisių metaduomenys. Jei vartotojas autorizuojamas, vartotojo užklauso apdorojimo komponentas apdoroja vartotojo užklauso, perduodamą HTTP (angl. *hyper text transfer protocol*) protokolu. Sugeneruojamas XML srautas, kuriame XML formatu aprašomi visi užklauso parametrai. Šis srautas apdorojamas trečiame konvejerio komponente, pritaikant XSLT transformaciją. Jos pagalba sudaromas naujo formato XML srautas, kuriame informacija aprašoma tokiu formatu, kokiu bus įrašoma į XML DB, bei aprašoma XUpdate formato užklausa. Šios užklauso pagalba paskutinis konvejerio komponentas patalpina informaciją į atitinkamą XML DB, kuri nurodoma šio komponento parametru.

4.4.6. Pirminių metaduomenų apdorojimas dokumento paieškos metu

Dokumento paieška pagal jo pirminius metaduomenis organizuojama, panaudojant XML duomenų bazių sistemos funkciją atlikti saugomų XML įrašų paiešką naudojant ieškomą dokumentą nusakančias XPath užklauso. Pavyzdžiui, jeigu pirminiai dokumento metaduomenys aprašomi 4.2.2 skyrelyje pateikto pavyzdžio formatu, ir norima rasti visus dokumentus su raktiniu žodžiu „valdymas“, XPath užklausa atrodytų taip:

```
//metadata[keywords/keyword = "valdymas"]
```

Pavyzdinis XML konvejeris, realizuojant paiešką, atrodo taip:

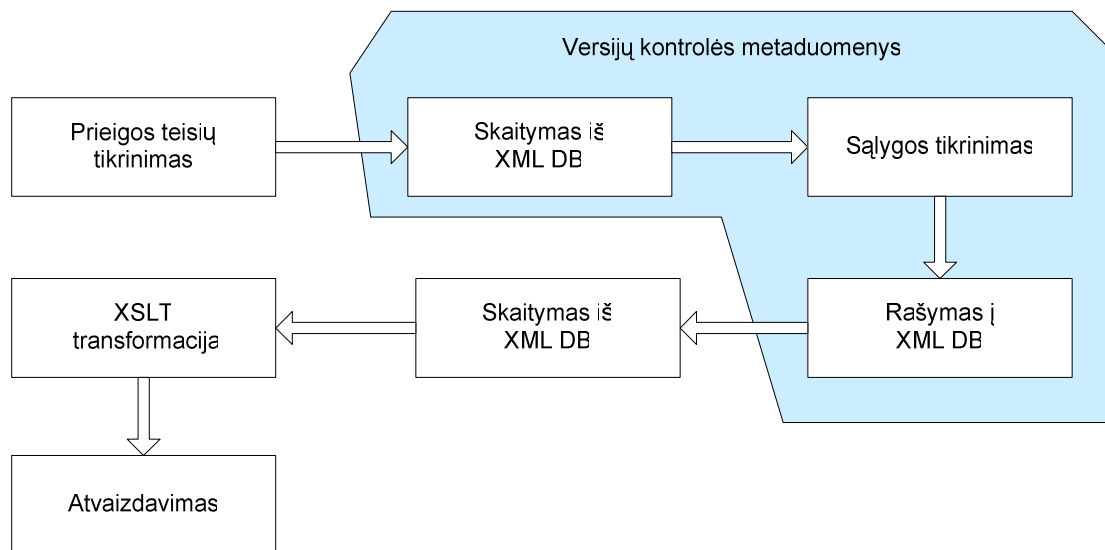


17 pav. Paieškos XML apdorojimo konvejeris

Vartotojo užklausa apdorojama, iš jos sugeneruojamas XML srautas, aprašantis paieškos parametrus. XML transformacijos metu sudaroma XPath užklausa, aprašanti šiuos parametrus. XPath užklausa įvykdoma XML DB, paieškos rezultatai pateikiami kaip XML srautas, kuriame kiekvienam rastam dokumentui skiriamas vienas įrašas. Šis srautas transformuojamas XSLT transformacija į vartotojui suprantamą formatą (pvz., HTML). Transformacijos metu rezultatų puslapyje sukuriamos nuorodos į kiekvieną rastą paieškos parametrus tenkinantį dokumentą. Rezultatų puslapis konvertuojamas ir persiunčiamas vartotojui.

4.4.7. Metaduomenų apdorojimas dokumento redagavimo metu

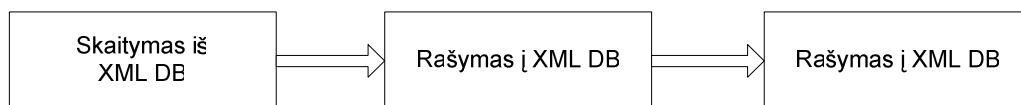
Kaip aprašyta poskyryje 2.3, dokumento redagavimas dokumentų valdymo sistemoje vyksta dviem etapais: vartotojas pirma „užsiregistruoja“ dokumento redagavimui. Sistemoje yra pažymima, kad dokumentas yra redaguojamas, ir kitiems vartotojams prieigos teisės prie šio dokumento yra apribojamos (pvz., uždraudžiama dokumentą redaguoti). Baigęs redaguoti dokumentą, vartotojas išsaugo pakeitimus dokumentų valdymo sistemoje. Sistema apdoroja pakeitimus ir „atlaisvina“ dokumentą, atstatydama buvusias prieigos teises.



18 pav. Registracijos redagavimui XML apdorojimo konvejeris

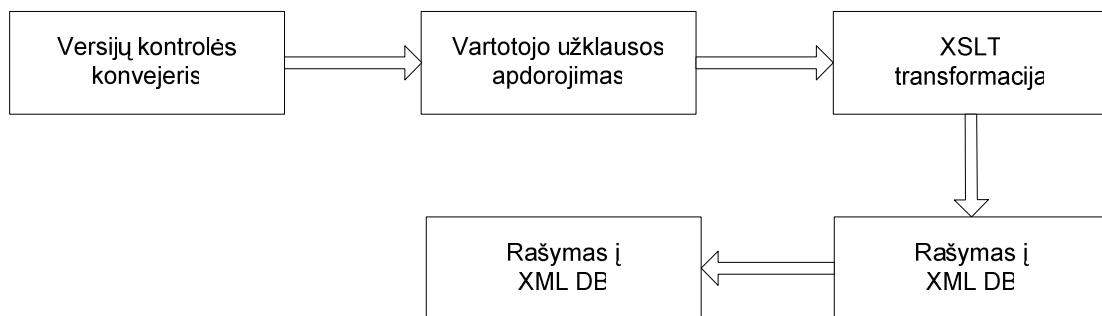
Visame konvejeriye naudojamas bendras parametras – dokumento unikalus identifikatorius. Pirmiausia patikrinamos vartotojo prieigos teisės, ar jį turi teisę redaguoti dokumentą. Tai daroma panaudojant prieigos teisių apdorojimo konvejerį, aprašytą skyrelyje 4.4.3. Jei vartotojas autorizuojamas, sekančiu žingsniu iš versijų kontrolės metaduomenų XML DB nuskaitomas versijų kontrolės įrašas. Trečiasis konvejerio komponentas patikrina ar dokumentas dar nėra redaguojamas. Jeigu dokumentas „laisvas“ versijų kontrolės metaduomenų įrašė pažymima, kad dokumentas redaguojamas. Penktas, šeštas ir septintas žingsniai atitinka funkcionalumą, realizuojamą dokumento peržiūrai (žr. sk. 4.4.4). Šiais žingsniais vartotojui yra pateikiama dokumento kopija redagavimui.

Pakeitimų išsaugojimo funkcionalumas realizuojamas atskiru XML apdorojimo konvejeriu. Pirmu žingsniu yra aktyvuojamas pagalbinis versijų kontrolės konvejeris. Jo funkcija yra nukopijuoti seną dokumento versiją į senų versijų saugyklą, bei padaryti atitinkamus pakeitimus dokumento versijų kontrolės metaduomenyse.



19 pav. Versijų kontrolės pagalbinis konvejeris

Dokumento sena versija yra skaitoma iš dokumentų saugyklos ir iš jos generuojamas XML srautas. Sekantis komponentas šį srautą vėl paverčia dokumentu ir išsaugo į senų versijų duomenų bazę. Paskutinis konvejerio komponentas padaro pakeitimus dokumento versijų metaduomenyse.



20 pav. Pakeitimų dokumente išsaugojimo konvejeris

Versijų kontrolės konvejeriui baigus savo darbą, apdorojama vartotojo užklausa. Sugeneruojamas XML srautas, kuriame XML formatu aprašomi visi užklauso parametrai. Šis srautas apdorojamas trečiame konvejerio komponente, pritaikant XSLT transformaciją, kuri informaciją iš užklauso parametru pavidalo pertvarko į dokumentą, kuris yra įrašomas į dokumentų XML DB. Paskutinis komponentas atlieka dokumento versijų kontrolės metaduomenų redagavimą, „atlaisvindamas“ dokumentą – t. y. pažymėdamas, kad dokumentas jau nebėra redaguojamas.

4.5. Apibendrinimas

Šiame skyriuje buvo pasiūlytas konceptualus modelis dokumentų valdymo sistemos metaduomenų aprašymui bei apdorojimui, panaudojant XML formatą metaduomenų aprašymui, bei XML srautų apdorojimo konvejerius metaduomenų apdorojimui. Buvo pateikti metaduomenų aprašymo formato pavyzdžiai, bei supaprastinti metaduomenų apdorojimo konvejeriai. Pagrindinė siūlomo modelio ypatybė yra tai, kad programiniai komponentai, skirti konvejerių sudarymui, visiškai nepriklauso nuo apdorojamų metaduomenų formatų bei organizacijos. Visa taikymo sričiai specifinė informacija – metaduomenų aprašymo schema, nuorodos į XML duomenų bazes metaduomenims saugoti, metaduomenų transformacijos – visa tai iškelta iš programinio kodo į konfigūracinius failus.

Šiame modelyje visos operacijos, priklausomos nuo metaduomenų formato, realizuojamos XSLT transformacijų pagrindu. Toks sprendimas smarkiai padidina sistemos lankstumą ir galimybes pritaikyti prie konkrečios taikymo srities specifikos – net ir iš pagrindų pakeitus metaduomenų aprašymo modelį, užtektų tik perrašyti XSLT transformacijų šablonus, ir sistema vėl korektiškai funkcionuotų.

5. TESTINIS PROGRAMINIS MODULIS

5.1. Realizacija

5.1.1. Java programavimo kalba

Testinis programinis modulis buvo realizuotas Java (29) programavimo kalba. Tai objektinė kalba, kuri yra labai populiari, nuolat vystoma ir tobulinama. Java panaudojimas užtikrina programinės įrangos pernešamumą – vieną kartą sukompiliuota, su Java parašyta programinė įranga veiks visose platformose, turinčiose Java virtualią mašiną.

Java programavimo kalba testinio programinio modulio realizacijai buvo pasirinkta dėl dviejų priežasčių:

- Java turi daugiausia XML formato duomenims apdoroti skirtų programinių įrankių ir bibliotekų. Naudingiausia iš jų metaduomenų apdorojimui yra SAX (angl. *Simple API for XML processing*) – nuoseklaus XML dokumentų apdorojimo biblioteka, įgalinanti daug našesnę XML formato dokumentų apdorojimą nei kitų programavimo kalbų siūlomos technologijos(31).
- 2003 m. pavasario semestrą, studijų Alborgo (Danija) universitete metu, buvo atliekamas Java ir C programavimo kalbų tyrimas ir našumo palyginimas taikant jas skaitmeninio signalų apdorojimo algoritmų realizavimui. Tyrimas parodė, kad Java sparta yra nedaug mažesnė, o kartais net ir didesnė negu C. Tyrimo rezultatai (žr. 1 priedą) yra įtraukti į tarptautinės konferencijos, skirtos įterptinėms sistemoms, pranešimų medžiagą (32).

5.1.2. Apache Cocoon XML konvejerių karkasas

Kadangi XML apdorojimui konvejeriais reikalingas atitinkamas programinis karkasas (angl. *framework*), testinio modulio realizacijai buvo naudojamas Apache Cocoon XML duomenų apdorojimo karkasas (24). Tai yra specialus programinės įrangos karkasas XML duomenų apdorojimui ir atvaizdavimui interneto puslapiuose skirtingais formatais. Karkasas yra realizuotas Java programavimo kalba, laikantis komponentinio programavimo principų ir yra paremtas XML apdorojimo konvejerių principu. Jame yra realizuota dauguma konvejerių funkcionavimui reikalingų priemonių, o taip pat kai kurie standartiniai XML apdorojimo komponentai. Taip pat šis karkasas pasižymi griežtai apibrėžta ir tvarkinga architektūra, leidžiančia karkasą plėtoti ir taikyti prie konkrečios veiklos srities.

5.1.3. Apache Xindice XML duomenų bazė

Apache Xindice – tai XML:DB standarto testinė realizacija (angl. *reference implementation*), skirta programinės įrangos kūrėjams, norintiems susipažinti su XML duomenų bazių valdymo sistemomis prieš investuojant į brangias komercines sistemas.

Xindice priklauso „grynųjų“ XML duomenų bazių kategorijai (žr. sk. 4.1.4) ir realizuoja visas standartines funkcijas: hierarchines XML duomenų bases, dokumento patalpinimą ir išgavimą pagal raktą, XPath užklausų bei XUpdate informacijos įrašymo kalbas. Dėl to Xindice yra ideali platforma testinei siūlomo metaduomenų modelio realizacijai.

5.1.4. Realizuota programinė įranga

Sukurtas prototipinis dokumentų valdymo sistemos branduolys metaduomenų apdorojimui pagal siūlomą modelį. Realizuotame testiniame modulyje aprašyti visi metaduomenų tipai bei realizuoti metaduomenų apdorojimo konvejeriai apibrėžti siūlomame modelyje.

Pagrindinis testinio modulio tikslas – palyginti, kiek padidėja užklausos aptarnavimo laikas, kuomet yra apdorojami dokumento metaduomenys, ir nustatyti, kokie veiksniai įtakoja apdorojimo trukmę (dokumentų kiekis saugykloje, apdorojamo dokumento dydis ir kt.).

5.2. Programinė įranga testavimui

5.2.1. XML duomenų bazių generavimas

Programinio modulio testavimui reikėjo įvairios apimties XML duomenų bazių su dokumentų ir metaduomenų aprašais. Tam buvo realizuotas programinis įrankis reikiamo dydžio XML duomenų bazėms generuoti.

Dokumentų XML duomenų bazei užpildyti buvo generuojami XHTML tipo dokumentai. Kiekvienas dokumentas buvo generuojamas su atsitiktiniu pastraipų skaičiumi (intervale nuo N iki M), kiekvienoje pastraipoje atsitiktinis žodžių skaičius (nuo K iki L). Kiekvienam generuojamam dokumentui buvo numatomas jo ilgis baitais (neskaitant XHTML žymių – matuojama vien tik dokumento teksto apimtis). Žodžiai dokumentų generavimui buvo imami atsitiktinai iš žodyno, sudaryto iš tekstinio failo.

Pirminiai dokumento metaduomenys buvo generuojami panašiu principu. Tik kiekvieno pirminių metaduomenų elemento generavimui naudojama atskira pradinių žodžių aibė, sugeneruojama iš to paties teksto, kuris naudojamas dokumentų generavimui.

Prieigos teisių metaduomenys visiems dokumentams generuojami tie patys – kadangi sistemoje yra tik vienas vartotojas, jis prie visų saugojamų dokumentų turi visas prieigos teises.

Loginės struktūros generavimas – sistemoje sugeneruojama fiksuotas skaičius katalogų, ir dokumentai atsitiktinai priskiriami vienam iš jų.

Versijų kontrolės metaduomenys taip pat visiems dokumentams generuojami vienodi – visi sugeneruoti dokumentai nėra redaguojami ir neturi senesnių versijų.

5.2.2. Užklausų generatorius

Realizuotas testavimo įrankis, kuris generuoja HTTP užklausas ir pateikia jas serveryje veikiančiam testiniam programiniam moduliui. Kiekvienai sugeneruotai užklausa yra išmatuojamas laikas nuo užklausos išsiuntimo iki atsakymo gavimo. Kad rezultatai būtų tikslesni, užklausa kartojama kelis (konkrečiu atveju buvo pasirinkta 10) kartų, ir paskaičiuojama vidutinė trukmė.

5.3. Rezultatai

5.3.1. Dokumento skaitymo testavimas

Šiame eksperimente buvo atlikti tokie testai:

SK – tiesioginis dokumento skaitymas iš XML DB, konvejerį sudaro tik vienas komponentas.

SK + PT – dokumento skaitymas iš XML DB, patikrinus vartotojo prieigos teises.

SK + XSL – tiesioginis dokumento skaitymas iš XML DB, prieš persiunčiant vartotojui, pritaikoma atvaizdavimo transformacija.

SK + PT + XSL – vartotojo prieigos teisių tikrinimas, dokumento skaitymas iš XML DB, XSLT atvaizdavimo transformacija prieš persiunčiant vartotojui.

2 lentelėje pateikiama užklausos apdorojimo laikų priklausomybė nuo dokumentų, esančių sistemoje, kiekio. Trukmės reikšmės pateikiamos normalizuotos mažiausios reikšmės atžvilgiu.

Lentelė 2

Dokumentų skaitymo testo rezultatai

Testo pavadinimas	Dokumentų skaičius sistemoje				
	100	500	1500	5000	10000
SK	1,00	1,08	1,16	1,32	1,37
SK + PT	1,62	1,70	1,81	2,01	2,07
SK + XSL	1,42	1,48	1,57	1,63	1,69
SK + PT + XSL	2,01	2,07	2,15	2,27	2,34

Paanalizavus 2 lentelės rezultatus, galima padaryti tokias išvadas:

- Vartotojo užklauskos apdorojimo trukmė priklauso nuo dokumentų kiekio sistemoje, tačiau ši įtaka nėra tokia didelė kaip buvo tikėtasi – padidinus saugomų dokumentų kiekį nuo 100 iki 10000, dokumento išgavimo laikas padidėjo apie 37%.
- Operacijos su metaduomenimis (šiuo atveju prieigos teisių tikrinimas) padidina apdorojimo laiką 50 – 60%. Taip yra dėl to, kad sistema turi atlikti papildomą kreipinį į XML duomenų bazę prieigos teisių informacijos nuskaitymui.
- XSLT transformacija dokumento atvaizdavimui padidina užklauskos apdorojimo trukmę 20 – 40%.

3 lentelėje pateikiama užklauskos apdorojimo laikų priklausomybė nuo apdorojamų dokumentų dydžio (kilobaitais). Trukmės reikšmės pateikiamos normalizuotos mažiausios reikšmės atžvilgiu.

Lentelė 3

Apdorojimo trukmės priklausomybė nuo dokumentų dydžio

Testo pavadinimas	Saugomų dokumentų dydis, KB				
	10 - 40	100 - 150	200 - 300	400 - 600	800-1000
SK	1,00	2,51	3,13	6,18	11,02
SK + PT	1,51	3,10	3,59	7,02	11,48
SK + XSL	1,38	3,07	3,69	7,51	12,42
SK + PT + XSL	2,07	3,65	4,21	8,28	13,03

Šiuo atveju užklauskos apdorojimo laikui didžiausią įtaką daro apdorojamų dokumentų dydis. Didėjant sistemoje saugomų dokumentų apimčiai, kartu auga ir nuskaitymo iš XML DB bei perdavimo per tinklą trukmė. Taip pat pastebimai padidėja XSLT transformacijos trukmė.

5.3.2. Skirtingų dokumentų valdymo operacijų palyginimas

Šiame eksperimente buvo atlikti tokie testai:

SK – dokumento peržiūros funkcija;

PAT – dokumento patalpinimo funkcija;

REG – registracijos dokumento redagavimui funkcija;

IŠS1 – pakeitimų dokumente išsaugojimas (be senų versijų išsaugojimo);

IŠS2 – pakeitimų dokumente išsaugojimas (su senų versijų išsaugojimu).

DVS funkcijų užklausos apdorojimo trukmių palyginimas

Testo pavadinimas	Dokumentų skaičius sistemoje				
	100	500	1500	5000	10000
SK	1,00	1,09	1,17	1,31	1,38
PAT	1,62	1,78	1,87	2,01	2,40
REG	2,01	2,24	2,35	2,40	2,63
IŠS1	2,14	2,41	2,56	2,96	3,15
IŠS2	2,56	2,86	3,01	3,24	3,52

Paanalizavus 4 lentelės rezultatus, galima padaryti tokias išvadas:

- Dokumento skaitymo iš saugyklos operacija atliekama sparčiausiai, kadangi ji nereikalauja rašymo į XML duomenų bazę.
- Panašiai kaip ir lentelės 2 rezultatuose, operacijos trukmė pailgėja iki 38% sistemoje saugomų dokumentų kiekiui išaugus nuo 100 iki 10000.
- Palyginti su skaitymo operacija, kitų operacijų trukmė išauga, priklausomai nuo jas realizuojančio XML konvejerio sudėtingumo. Sudėtingiausios operacijos – dokumento pakeitimų išsaugojimas su senos versijos išsaugojimu trukmė yra apie 2,5 karto ilgesnė nei dokumento skaitymo operacija.

5.3.3. Apibendrinimas

Apibendrinant testavimo rezultatus, galima padaryti tokias išvadas:

- Panaudojus XML konvejerius panaudojimas dokumentų metaduomenų apdorojimui, užklausos apdorojimo laikas išauga iki dviejų kartų skaitymo operacijos metu ir iki penkių kartų dokumento pakeitimo išsaugojimo metu (su senos versijos išsaugojimu).
- Užklausos apdorojimo laikas auga nežymiai, kuomet didėja sistemoje saugomų dokumentų skaičius.
- Didėjant apdorojamų dokumentų apimčiai (kilobaitais), auga ir užklausos apdorojimo trukmė. Tai vyksta dėl to, kad didelio dokumento persiuntimas per tinklą užtrunka ilgiau, o taip pat XSLT transformacijos užtrunka ilgiau, kuomet apdorojami dideli XML dokumentai.

6. IŠVADOS

Šio darbo metu buvo atlikta įvairių šiuolaikinių dokumentų valdymo sistemų apžvalga, aptartas dažniausiai sutinkamas funkcionalumas bei pasitaikančios problemos. Norint įmonėje efektyviai panaudoti dokumentų valdymo sistemą, reikia ją pritaikyti prie įmonės specifikos bei darbo srities. Tačiau tokių specializuotų sistemų kūrimas yra labai brangus procesas, ir jas gali įsigyti tik didelės kompanijos. Kita vertus, egzistuoja palyginti pigios, universalios dokumentų valdymo sistemos, kurios kuriamos plačiam naudojimui ir nėra orientuotos į jokią veiklos sritį. Tačiau tokias sistemas labai dažnai sunku (o kartais net visai neįmanoma) pritaikyti prie įmonės poreikių.

Analizės metu pastebėta, kad dažnai būtent uždaras ir nelankstus dokumentų valdymo sistemos metaduomenų modelis yra pagrindinė kliūtis pritaikyti sistemą prie įmonės veiklos specifikos.

Šiame darbe pasiūlytas naujas dokumentų valdymo sistemos metaduomenų aprašymo bei apdorojimo koncepcinis modelis, paremtas XML technologijomis. Metaduomenys saugomi XML formatu ir apdorojami taikant XML apdorojimo konvejerius. Pritaikius šį modelį, tiek metaduomenys, tiek jas apdorojimo valdymo logika nėra susijusi su sistemos programiniu kodu – kitaip tariant sistema nėra „pririšta“ prie vieno konkretaus metaduomenų apdorojimo formato. Tokiu būdu realizuotą sistemą galima lengvai pritaikyti prie įmonės veiklos srities – apdorojamų dokumentų metaduomenų formatas lengvai gali būti modifikuojamas tam, kad geriau atspindėtų įmonėje vykstančius procesus.

2003 m. pavasario semestrą, studijų Alborgo (Danija) universitete metu, buvo atliekamas Java ir C programavimo kalbų tyrimas ir našumo palyginimas. Tyrimų rezultatai (žr. 1 priedą) yra įtraukti į tarptautinės konferencijos, pranešimų medžiagą (32). Tyrimai parodė, kad Java kalbos greitimeika yra nedaug atsilieka nuo C, todėl testinio programinio modulio realizacijai buvo pasirinkta Java programavimo kalba.

Darbo metu taip pat buvo sukurtas aprašomą metaduomenų modelį realizuojantis programinis testinis modulis. Jis buvo naudojamas iširti, kaip padidėja dokumento apdorojimo sistemoje laikas, realizavus metaduomenų apdorojimą pagal aprašytą modelį. Pastebėta, kad sistemos reakcijos laikas išauga nuo dviejų iki penkių kartų (priklausomai nuo atliekamos operacijos), lyginant su tiesioginio dokumento skaitymo iš XML duomenų bazės trukme. Jeigu sistema taikoma nedidelėje ar vidutinėje įmonėje, toks reakcijos laiko padidėjimas yra visai priimtinas.

Numatyti tolimesni darbai:

- Įtraukti į modelį ir sudėtingesnio dokumentų valdymo sistemų funkcionalumo aprašymą: darbo procesų (angl. *workflow*) realizavimą, integraciją su kitomis įmonės valdymo sistemomis (turinio, žinių valdymo, grupinio darbo palaikymo ir pan.).
- Išplėsti testinio programinio modulio funkcionalumą. Apibrėžus griežtas sąsajas ir realizavus grafinę sąsają sukurtas programinis modulis galėtų būti naudojamas kaip pilnavertės dokumentų valdymo sistemos branduolys.
- Atlikti papildomų metaduomenų apdorojimo trukmės tyrimų, naudojat pažangesnę programinę įrangą – pvz., komercinę XML duomenų bazę.
- Optimizuoti programinį kodą, realizuojantį XML konvejerių komponentus, siekiant sumažinti apdorojimo trukmes.

7. LITERATŪRA

1. Mezaine, F., Regzui Y. A document management methodology based on similarity contents/ F. Mezaine, Y. Regzui // Information Sciences, 2004., Nr. 158, p. 15 – 36.
2. Yao, Y. H. XML-based ISO9000 electronic document management system / Y. H. Yao, A. Trappey, P. S. Ho// Robotics and Computer Integrated Manufacturing, 2003, Nr. 19, p. 355 – 370.
3. Sutton, M. Document Management for the Enterprise: Principles, Techniques, and Applications, 1996. 400 p. ISBN 0471147192.
4. Dieng, R. Methods and Tools for Corporate Knowledge Management/ Dieng, R., Corby, O., Giboin, A., & Ribière, M.// 11th Banff Workshop on Knowledge Acquisition, Modelling and Management, KAW'98: tarptautinės konferencijos medžiaga [Banff, Alberta, Canada, 1998 m. balandžio 18 – 23 d.].
5. A Basic Introduction to Electronic Document (Content) Management Systems [online]. Chedworth, UK: G G Baker & Associates. Prieiga per internetą: <<http://www.ggbaker.com/GGB&A/EDMS%20Tutorial/IntroEDMS.htm>>.
6. Banayay, C. Web Content Management. The BizTech Network [online]. 2000, gruodis [žiūrėta 2004-04-19]. Prieiga per internetą: <<http://www.brint.com/members/20120418/content/>>
7. Jeffery, K. Metadata: The Future of Information Systems. World Meteorological Organization [online]. 2000. [žiūrėta 2004-04-21]. Prieiga per internetą: <<http://www.wmo.ch/web/www/WDM/ET-IDM/Doc-2-3.pdf>>
8. Content Standard for Digital Geospatial Metadata. Federal Geographic Data Committee [online]. 1998. [žiūrėta 2004-04-21]. Prieiga per internetą: <<http://www.fgdc.gov/metadata/constan.html>>
9. Draft WMO Core Metadata Standard – XML representation. World Meteorological Organization [online]. 2002, gegužė. [žiūrėta 2004-04-21]. Prieiga per internetą: <<http://www.wmo.ch/web/www/metadata/WMO-metadata-XML.html>>
10. Resource Description Framework (RDF). World Wide Web Consortium [online]. 2004, vasaris. Prieiga per internetą: <<http://www.w3.org/RDF/#specs>>
11. Jeffery K. Metadata: An Overview and some Issues. 11th ERCIM Database Research Group Workshop on Metadata for Web Databases: tarptautinės konferencijos medžiaga [Sankt Augustin, Germany, 1998 m. gegužės 25 – 26 d.].

12. Dublin Core Metadata Element Set, Version 1.1: Reference Description. Dublin Core Metadata Initiative [online]. 2003, birželio 2 d. [žiūrėta 2004-04-25]. Prieiga per internetą: <<http://dublincore.org/documents/dces/>>
13. Freed, N. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types/N. Freed, N. Borestein. RFC 2046 [online]. , 1996, lapkritis. [žiūrėta 2004-03-15]. Prieiga per internetą: <<http://www.rfc-editor.org/rfc/rfc2046.txt>>
14. Role-Based Access Control Models/ R. Sandhu, E. Coyne, H. Feinstein, C. Youman // IEEE Computer. 1996, Nr. 29(2), p. 38-47.
15. McLean, J. Security models // Encyclopedia of Software Engineering. Wiley Press, 1994.
16. Nagaratnam, N. Resource Access Control for an Internet User Agent/ N. Nagaratnam, S. Byrne // Proceedings of the USENIX Conference on Object-Oriented Technology and Systems '97: konferencijų medžiaga. Portland, Oregon, USA. [1997 m. birželio 16 – 20 d.]
17. Song, Y. Anticipating reuse: documenting buildings. for operations using web technology/ Y. Song, M. Clayton., R. Johnson // Automation in Construction, 2002, Nr 11, p. 185 – 197.
18. Edwards, K. Balancing Generality and Specificity in Document Management Systems/ K. Edwards, A. Lamarca // Palo Alto Research Center [online]. 2001 [žiūrėta 2004-04-20]. Prieiga per internetą: <<http://www2.parc.com/csl/members/kedwards/pubs/interact-docurama.pdf>>
19. World Wide Web Consortium. Extensible Markup Language (XML) 1.0 (Third Edition) [interaktyvus]. 2004 vasario 4 d., [žiūrėta 2004-05-03]. Prieiga per internetą: <<http://www.w3.org/TR/REC-xml/>>.
20. World Wide Web Consortium. XML Path Language (XPath) 1.0 [online]. 1999 lapkričio 16 d., [žiūrėta 2004-05-04]. Prieiga per internetą: <<http://www.w3.org/TR/xpath>>.
21. World Wide Web Consortium. XSL Transformations (XSLT) 1.0 [online]. 1999 lapkričio 16 d., [žiūrėta 2004-05-04]. Prieiga per internetą: <<http://www.w3.org/TR/xslt>>.
22. Nottingham, M. Akamai's position paper for the W3C XML Processing Model Workshop // Distributed XML Processing Models [online]. 2001, [žiūrėta 2004-05-06]. Prieiga per internetą: <<http://www.mnot.net/papers/XMLProcessingWS.html>>
23. Irwin, W. XML in the visualisation pipeline/ W. Irwin, N. Churcher // Proc. Selected papers from 2001 Pan-Sydney Workshop on Visual Information Processing, Sydney, Australia. Conferences in Research and Practice in Information Technology, 11. 2001, p. 59 – 67.
24. The Apache Software Foundation. Apache Cocoon 2.1 [online]. [žiūrėta 2004-05-10]. Prieiga per internetą: <<http://cocoon.apache.org/2.1/>>

25. Salminen, A. Requirements for XML Document Database Systems/ A. Salminen, F.Tompa // ACM Symposium on Document Engineering, November 2001, 85-94.
26. Salminen, A. System Desiderata for XML Databases/ A. Salminen, F.Tompa // Proceedings of the 27th VLDB Conference, Roma, Italy, 2001.
27. XML:DB Initiative. XML:DB Initiative: XUpdate - XML Update Language [online]. 2000 rugsėjo 14d. [žiūrėta 2004-04-24]. Prieiga per internetą: <<http://www.xmldb.org/xupdate/xupdate-wd.html>>
28. World Wide Web Consortium. XML Schema Part 0: Primer [online]. 2001 gegužės 2 d., [žiūrėta 2004-05-04]. Prieiga per internetą: <<http://www.w3.org/TR/xmlschema-0/>>
29. Eckel, B. Thinking in Java, 2nd Edition. Prentice Hall, 2000. 1200 p. ISBN 0 13-027363-5.
30. The Apache Software Foundation. Apache Xindice 1.0 [online]. [žiūrėta 2004-05-12]. Prieiga per internetą: <<http://xml.apache.org/xindice/>>
31. The Official Website For SAX [online]. [žiūrėta 2004-05-12]. Prieiga per internetą: <http://www.saxproject.org/>
32. Java and C Performance Evaluation for DSP Applications / E. Kazanavičius, V. Dusevičius, A. Liutkevičius, R. Žukaitis // Conference at GSPx 2004, Embedded Applications: Software & Hardware. Santa Clara, CA USA, Sept. 27-30, 2004. Prieiga per internetą: http://www.gspix.com/GSPX/2004_submissions.php