

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

PROGRAMŲ INŽINERIJOS KATEDRA

Aistė Trepekaitė

Knygų leidyklos dokumentų valdymo sistema

Magistro darbas

Darbo vadovas

doc. E. Karčiauskas

Kaunas, 2007

KAUNO TECHNOLOGIJOS UNIVERSITETAS
INFORMATIKOS FAKULTETAS
PROGRAMŲ INŽINERIJOS KATEDRA

Aistė Trepekaitė

Knygų leidyklos dokumentų valdymo sistema

Magistro darbas

Recenzentas

2007-05-25

prof. dr. R. Butleris

Vadovas

2007-05-25

doc. E. Karčiauskas

Atliko

2007-05-25

IFM-1/2 gr. stud.
Aistė Trepekaitė

Kaunas, 2007

TURINYS

1. Įvadas	5
2. Technologijų ir programų sistemų apžvalga.....	6
2.1. Dokumentų valdymo apibrėžimai.....	6
2.1.1. Dokumentų valdymas	6
2.2.2. Dokumentų valdymo sistema.....	7
2.2. Dokumentų valdymo sistemų struktūra	8
2.2.1. Skirtumai tarp duomenų ir dokumento	8
2.2.2. Pagrindinės DVS funkcijos.....	9
2.3. DVS kitų sistemų kontekste.....	11
2.3.1. Standartiniai ir individualūs PĮ paketai.....	12
2.3.2. Interneto taikomosios programos ir portalai	12
2.3.3. Turinio valdymo sistemos.....	13
2.3.4. Darbų srautų sistemos	13
2.3.5. Grupinio darbo sistemos	14
2.3.6. Žinių valdymo sistemos	15
2.4. Dokumentų valdymo standartai	15
2.5. Dokumentų valdymas internete	16
2.5.1. Saugumas	16
2.5.2. Duomenų perdavimo optimizavimas	17
2.5.3. WebDAV	17
2.5.4. XML duomenų vaizdavimo standartas	18
2.5.3.1. XML korektiškumo tikrinimas	19
2.5.3.2. Standartinės sąsajos	20
2.5.3.3. XML transformavimas.....	21
2.5.3.4. XML dokumentų dalių išskyrimas ir apdorojimas	22
2.5.3.5. XML duomenų bazės.....	22
2.6. Dokumentų valdymo sistemų modeliai	23
2.6.1. Hierarchinis dokumentų valdymo modelis	23
2.6.1.1. Modelio trūkumai	24
2.6.1.2. Alternatyvūs sprendimai	24
2.6.2. Metaduomenimis grįsti modeliai	25
2.6.2.1. Onix – knygų metaduomenų standartas.....	26
2.6.3. Dokumento modeliu pagrįstas modelis.....	28
2.7. Egzistuojančių knygų leidyklų programų sistemų apžvalga.....	29
2.8. Knygų leidybos dokumentų valdymas.....	33

2.8.1. Knygų leidybos procesas	33
2.8.2. Knygų leidykloje saugomos informacijos kategorijos.....	34
2.8.3. Reikalavimai knygų leidyklos dokumentų valdymo sistemai	36
3. Projektinė dalis	39
3.1. Projekto paskirtis ir apribojimai	39
3.2. Veiklos sąveikų modelis	39
3.2.1. Veiklos sudėtis	39
3.2.3. Veiklos padalinimas.....	40
3.3. Sistemos panaudojimo atvejai	41
3.3.1. Sistemos aktorių aprašymas.....	42
3.3.2. Panaudojimo atvejų aprašymas.....	42
3.4. Sistemos architektūra	43
3.4.1. Sistemos komponentai	43
3.4.1.1. Paketas vartotojo sąsaja	43
3.4.1.2. Paketas Duomenų bazė	44
3.4.1.3. Paketas Administravimas.....	44
3.4.1.4. Paketas Leidybiniai darbai.....	45
3.4.1.5. Paketas Moduliai.....	45
3.4.2. Duomenų vaizdas.....	48
3.4.3. Išdėstymo vaizdas	49
4. Tyrimo dalis	50
4.1. Sukurtos sistemos kokybės tyrimas	50
4.1.1. Testavimas	50
4.1.2. Peržiūros	51
4.1.2.1. Kokybės kriterijai	51
4.1.2.2. Kokybės vertinimo rezultatai.....	51
4.2. Tolesnės vystymo galimybės	52
5. Išvados	55
6. Literatūra.....	56
7. Terminų ir santrumpų žodynas	58
1 Priedas. Perdavimo-priėmimo aktas.....	59

Lentelių sąrašas

1 lentelė. Darbų srautų elementai	14
2 lentelė. Grupinio darbo ir darbų srautų sistemų skirtumai.....	14
3 lentelė. Knygų leidykloje saugomos informacijos kategorijos.....	35
4 Lentelė. Veiklos padalijimas.....	40
5 lentelė. Esių aprašymas.....	48
6 lentelė. Knygų leidyklos dokumentų valdymo sistemos kokybės vertinimo parametrai	51
7 lentelė. Knygų leidyklos dokumentų valdymo sistemos atitikimas kokybės vertinimo kriterijams	52

Paveikslėlių sąrašas

1 pav. Apibendrintas dokumento gyvavimo ciklas dokumentų valdymo sistemoje.....	7
2 pav. Archyvas ir metaduomenų/indeksų duomenų bazė	9
3 pav. Darbų srautas.....	13
4 pav. XSL architektūra	21
5 pav. Apibendrintas metaduomenų modelis.....	26
6 pav. Metaduomenų aprašantysis informacinis modelis	26
7 pav. Onix įrašo dalies variantai, žymėms naudojant pilną tekstą ir trumpus kodus.	27
8 pav. Typefi Publishing System integravimas į TVS naudojantis XML	31
9 pav. Knygų leidyklos organizacinis modelis	34
10 pav. Supaprastintas darbų srautas	36
11 pav. Leidybinio projekto būsenų diagrama.....	37
12 pav. Dokumentų hierarchija knygos rengimo procese.....	38
13 Pav. Veiklos konteksto diagrama.....	40
14 pav. Sistemos panaudojimo atvejų vaizdas.....	41
15 pav. Sistemos išskaidymas į paketus	43
16 pav. Paketo Vartotojo sąsaja klasių diagrama	43
17 pav. Paketo Autorizacija klasių diagrama.....	44
18 pav. Paketo Duomenų bazė klasių diagrama	44
19 pav. Paketo Administratorius klasių diagrama	45
20 pav. Paketo Leidybiniai darbai klasių diagrama.....	45
21 pav. Paketas Moduliai.....	46
22 pav. Paketo Paieška klasių diagrama	46
23 pav. Paketo Leidybiniai projektai klasių diagrama.....	46
24 pav. Paketo Vartotojai klasių diagrama	47
25 pav. Duomenų vaizdo schema	48
26 pav. Išdėstymo vaizdas	49
27 pav. DVS išplėtimas	53
28 pav. Duomenų srautas XSL-FO.....	54

SUMMARY

Document management system for a book publishing office

Documents are very important to modern organizations. The software dealing with the management of documents and files is electronic document management. Cost reductions and quality improvements are immediate incentives. Many universal document management systems are developed for a wide use in various areas of activities, but none of them satisfied the requirements of the publishing process. Therefore the main objective of this paper is to identify the functionality of a document management system to handle the documents in a book publishing process.

The thesis contains basic concepts and technologies on a scale from file hierarchy based to document model based systems are classified and analysed from a document management perspective. Also a description of document management in book publishing process, including reviews of commercial applications, book industry standards and current best practice is given.

In the second half of this paper an implementation of a document management system in a book publishing process is illustrated. It covers system architecture, quality evaluation as well as suggestion for a future research.

1. ĮVADAS

Elektroniniai dokumentai – šiuolaikinės organizacijos veiklos pagrindas. Organizacijoms plečiantis, tuo pačiu ir gausėjant dokumentų kiekiui, stiprėja dokumentų valdymo poreikis. Su tuo pačiu dokumentu dirbant keliems žmonėms atsiranda keletas jo kopijų, – taip prarandama vis daugiau laiko reikiamos informacijos paieškai. Dėl organizacijos dokumentų valdymo strategijos nebuvimo žmogiškieji resursai panaudojami neoptimaliai. Informacinių technologijų taikymas leidžia efektyviau koordinuoti ir plėtoti veiklą.

Programinė įranga, skirta valdyti dokumentų srautus ir koordinuoti su jais susijusias vykdomas užduotis organizacijoje, vadinama dokumentų valdymo sistema. Jos dėka informacija yra valdoma ir tvarkoma taip, kad padėtų žmonėms vykdyti jų veiklą, sėkmingiau vykdyti procesams ir sąveikoms, o ne tik būtų duomenų pertvarkymas.

Šiuo metu tiek pasaulio tiek Lietuvos IT rinkoje sukurta daug dokumentų valdymo sistemų, kurios yra įdiegtos įvairiose srityse, knygų leidyba ne išimtis. Daugelis jų per daug apkrautos funkcionalumu ir neatitinka leidybos proceso valdymui užsakovo įmonėje keliamų reikalavimų. Tad šio darbo pagrindinis tikslas – realizuoti knygų leidybos proceso dokumentų valdymo sistemą.

Tyrimo objektas – dokumentų valdymo sistema knygų leidybos kontekste.

Darbo uždaviniai kyla tiesiogiai iš pagrindinio tikslo:

- Išanalizuoti elektroninių dokumentų valdymo sistemų savybes, kūrimo technologijas ir nustatyti kompiuterizuojamų leidybos proceso dokumentų valdymo funkcijų rinkinį.
- Tyrimu įvertinti sukurtos knygų dokumentų valdymo sistemos kokybę ir pateikti siūlymus tolesniam jos tobulinimui.

Darbo analitinėje dalyje analizuojami pagrindiniai dokumentų valdymo principai, atliekama egzistuojančių sprendimų apžvalga, išanalizuojamos bei įvardijamos problemos, kurias knygų leidyklos dokumentų valdymo sistema turėtų išspręsti. Projektinėje dalyje pateikiamas knygų leidyklos dokumentų valdymo sistemos realizacijos kelias, trumpai apibūdinamas technologijų pasirinkimas, architektūros specifiška ir glaustai pateikiamas sprendimo klasių diagramos bei duomenų vaizdas. Tyrimo dalyje atliekamas sistemos tyrimas, įvertinama sistemos kokybė, pateikiamos rekomendacijos tolesniam vystymui. Darbo gale pateikiamos išvados.

2. TECHNOLOGIJŲ IR PROGRAMŲ SISTEMŲ APŽVALGA

2.1. Dokumentų valdymo apibrėžimai

Elektroninių dokumentų valdymo sistemų tikslas – palaikyti informacijos procesus aprūpinant vartotojus paprastais, loginiais ir greitais dokumentų saugojimo, paieškos ir atrinkimo metodais.

Šiandien bendrinio termino *dokumentas* koncepcija informacinių sistemų standartizavime išplėsta. Siaurąja prasme tai fiziniai dokumentai, atspausdinti popieriuje arba laikomi fizinėse laikmenose (pvz., CD, DVD).

ISO/IEC 3613-1 standartas siūlo platesnį apibrėžimą: *struktūrizuota žmogui suprantamos informacijos visuma, kuria kaip vienetu gali keistis vartotojai bei sistemos* [1].

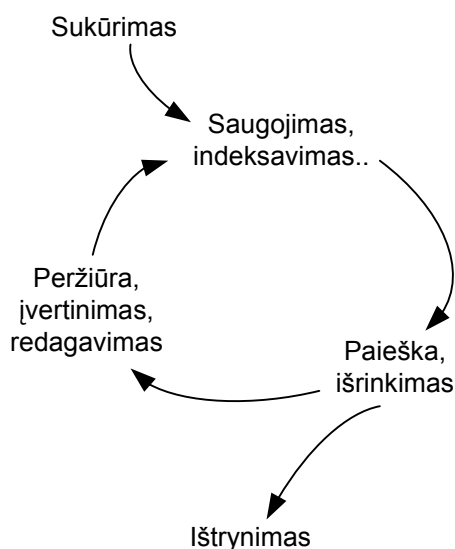
Taigi plačiąja prasme dokumento sąvoka išplečia siaurąją koncepciją: dokumentas struktūrizuotu būdu apjungia kontekstinę informaciją, skirtą publikuoti (pvz., interneto portaluose, kompiuteriu sugeneruotose ataskaitose ir kt.) bei keistis tarpusavyje, ne tik kaupti fizinėse laikmenose ar popieriuje. Elektroninis dokumentas gali būti sudarytas iš keleto skirtingų elementų (teksto, vaizdo ir kt.), išsaugomas skirtinguose skaitmeniniuose formatuose.

2.1.1. Dokumentų valdymas

Į dokumentų valdymo sampratą įeina dokumentų valdymas: čia labiau akcentuojami daugiau administraciniai aspektai nei dokumento turinio kūrimas ir plėtojimas, nes dauguma svarbių verslo procesų organizacijose remiasi dokumentų srautais. Dėl to dokumentams turi būti skiriamas ypatingas dėmesys ne tik operaciniame, bet ir strateginiame lygmenyje [2]. Dokumentų valdymas apjungia visus procesus, užduotis ir atsakomybes, kurios yra susijusios su dokumentų tvarkymu. Tai sudėtinis procesas apimantis visą dokumento gyvavimo ciklą: nuo sukūrimo, derinimo, naujų versijų kūrimo, iki saugojimo ir archyvavimo (žr. 1 pav.)

Kiekvienam dokumentui organizacijoje apibrėžiamas jo gyvavimo ciklas organizacijoje [3]:

- Pradinė informacija apie dokumentą ir kaip jis saugomas archyve.
- Skirtingi kriterijai dokumentams saugomiems archyve.
- Paieškos kriterijaus apibrėžimas, dokumentams archyve rasti ir dar kartą panaudoti.
- Dokumentai gali būti keičiami, publikuojami ir vėl talpinami archyve.
- Susitarimai dėl saugojimo periodo, nuo sukūrimo iki ištrynimo.



1 pav. Apibendrintas dokumento gyvavimo ciklas dokumentų valdymo sistemoje

2.2.2. Dokumentų valdymo sistema

Vienas iš dokumentų valdymo sistemos apibrėžimų: Tai automatizuotas elektroninių dokumentų valdymas per visą jų gyvavimo ciklą organizacijoje, nuo pradinio sukūrimo iki jų archyvavimo. Elektroniniai dokumentai gali būti bet kokio tipo skaitmeniniai objektai – HTML, XML, PDF, grafikos failai, elektroninės lentelės, tekstiniai bei sudėtiniai dokumentai [4].

Apibrėžti vienintelį programinės įrangos paketą, technologiją ar funkcijų rinkinį dokumentų valdymo sistemai praktiškai neįmanoma. Paprastai DVS naudoja duomenų bazę ir / arba failų sistemą informacijos laikymui ir valdymui.

Vienas iš painių aspektų yra tai, kad terminas *dokumentų valdymo sistema* neapibūdina vieningo tipo programinės įrangos paketo ar technologijos, kuriai būdingos specifinės, sutartos funkcijos. Terminas atspindi aibę tarpusavyje susijusių sistemų, kurios atlieka tam tikras funkcijas priklausomai nuo aplinkos ir užduoties, kurioms sistema yra suprojektuota.

Dėl vieningo ir bendro apibrėžimo trūkumo apie DVS galima kalbėti siaurąja ir plačiąja prasmėmis. Siaurąja prasme DVS apibrėžia dokumentų valdymo logiką, jų būseną, struktūrą, gyvavimo ciklą (žr. 1 pav.) ir turinį. Dokumentai yra aprašomi, kategorizuojami ir organizuojami specifinėje loginėje struktūroje, kad būtų įmanoma kuo lengviau juos ištrinti. Dokumentai yra sukuriami, tvirtinami, keičiami ir ištrinami arba įrašomi į archyvą.

DVS plačiąja prasme turi papildomas funkcijas tokias kaip optinį simbolių atpažinimą, automatinį indeksavimą, darbų srautų valdymą, skanavimą, publikavimą. Ribos tarp DVS ir

kitų sistemų, tokių kaip žinių, turinio valdymo arba įmonės resursų (angl. ERP – *Enterprise Resource Planning*) planavimo sistemų nėra tiksliai nubrėžtos. Funkcijos persipina.

DVS naudojimas teikia šiuos privalumus:

- Supaprastina ir pagreitina paiešką dokumento turinyje ir aprašyme.
- Suteikia nuotolinę prieigą prie dokumentų archyvo.
- Versijavimas ir pakeitimų stebėjimas tampa aiškus ir patikimesnis.
- Dokumentų perdavimas ir paskirstymas tampa greitesnis, atkartotinas ir patikimesnis.
- Dokumento saugojimo periodas ir automatinis ištrynimasis lengviau valdomi, taip sutaupant saugojimo kaštus.
- Dokumento turinio struktūra tampa labiau valdoma ir atkartojama.

2.2. Dokumentų valdymo sistemų struktūra

Dokumentas – DVS pagrindas, tad pirmiausia išskirsime jų tipus [5]:

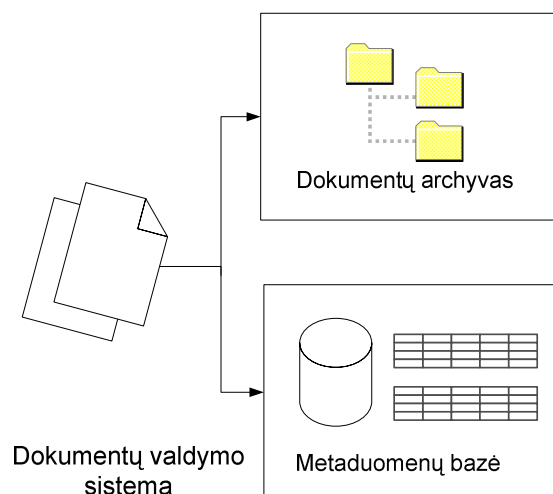
- Struktūrizuoti dokumentai – dokumentai, saugantys savyje koduotą informaciją, pagal kurią gali būti automatiškai apdorojami, pvz., tekstiniai, HTML, XML failai.
- Nestruktūrizuoti dokumentai – dokumentai, saugantys nestruktūrizuotą informaciją, pvz., grafinis failas, nuskanuotas dokumentas ir kt.

Paprastai dokumentai yra klasifikuojami kūrimo metu (skanuojant paveikslėlių, generuojant el. komercijos važtaraštį XML formatu), bet ši klasifikacija gali vėliau pasikeisti (pvz., nuskanavus dokumentą, iš jo informaciją galime išgauti optiniu simbolių atpažinimo metodu).

2.2.1. Skirtumai tarp duomenų ir dokumento

Dažnai sakoma, kad duomenys yra stipriai struktūrizuota informacija, tuo tarpu dokumentai yra neapibrėžtos struktūros. Techniškai nestruktūrizuota informacija saugoma dideliuose dvejetainiuose objektuose (neapibrėžtos struktūros dvejetainėse sekose) (angl. *blob* – *binary large object*). Struktūrizuota informacija (tekstiniai dokumentai, XML failai) yra saugomi dalinai arba pilnai reliacinėse duomenų bazėse [6].

Tipinės DVS atskiria dokumentų archyvą nuo metaduomenų/indeksų duomenų bazės (žr. 2 pav.)



2 pav. Archyvas ir metaduomenų/indeksų duomenų bazė

Dokumentą aprašanti informacija (indeksas, raktiniai žodžiai, autorius ir kt., taip vadinami metaduomenys) yra įrašoma duomenų bazėje su papildoma nuoroda į failą archyve.

Pasirinktinai archyvas gali būti perkeltas į tą pačią duomenų bazę, kad būtų išvengtos sinchronizacijos problemos, atsirandančios keitimų metu.

DVS saugoma informacija apie dokumentą:

- Originalus skaitmeninis dokumentas.
- Raktas. Kiekvienas dokumentas turi unikalų raktą DVS sistemoje.
- Metaduomenys. Dokumento atributai, palengvinantys paiešką ir išrinkimą.
- Struktūrinė informacija. Naudojant XML, EDIFACT ir kt. dokumentus DVS gali talpinti vidinę dokumento struktūrą duomenų bazėje. Šiuolaikinės duomenų bazės sistemos palaiko struktūrizuotą XML failų saugojimą, o turinio atributams išrinkti naudoja užklausų kalbas.
- Taisyklės. Dokumento gyvavimo ciklas, prieigos teisės bei kiti nustatymai, apibrėžiami DVS taisyklių rinkinyje.

Jei tik pirmieji du punktai realizuoti, DVS – tik paprasta archyvavimo sistema.

2.2.2. Pagrindinės DVS funkcijos

Pirmosios dokumento valdymo sistemų turėjo tik šias pagrindines savybes [4]:

- Dokumentų saugykla, kur elektroniniai dokumentai buvo archyvuojami.
- Skirtingos įvesties funkcijos dokumentams įtraukti į saugyklą.
- Užklausos funkcija, skirta archyve talpinamiems dokumentams lokalizuoti ir identifikuoti.
- Išvesties funkcijos dokumentams iš archyvo išrinkti.

Šiandien DVS sistemos palaiko daugybę funkcijų per visą dokumento gyvavimo ciklą organizacijoje, jas būtų galima sugrupuoti pagal pagrindinius dokumento gyvavimo ciklo etapus:

1. Funkcijos dokumento kūrimo etape (funkcijos, palengvinančios arba sutrumpinančios dokumento kūrimo laiką):
 - Elektroninis vaizdo išgavimas, – į DVS informacija įkeliama skenuojant dokumentus.
 - OCR, ICR atpažinimo technologijos į DVS nuskanuoto dokumento vaizdą konvertuoja į tekstinį formatą.
 - Importuotų dokumentų apdorojimo galimybė DVS sistemoje. Taip pat turi būti integruoti nuosavybinių (angl. *proprietary*) programų moduliai (Pvz.: *CorelDraw*, *Photoshop*). Dažnai skirtingų dokumentų peržiūros programos į DVS integruojamos per API sąsają.
 - Funkcijos, pagreitinančios dokumento kūrimą, jo turinio valdymu.
 - Bendradarbiavimo savybė leidžia vienu metu prie dokumento prieiti keliems vartotojams, susijusiems su to dokumento kūrimu.
 - Indeksavimas – iš dokumentų išrenkami svarbūs atributai, palengvinantys jų paiešką.
2. Funkcijos dokumento apdorojimo etape (palengvinančios arba sutrumpinančios dokumento kūrimo laiką):
 - Dokumentų versijų palaikymas, versijų istorija ir visų veiksmų atliktų sistemoje auditas. Egzistuoja dvi koncepcijos: saugant visą naują dokumentą arba tik besiskiriančią dokumento dalį.
 - Dokumento paėmimo/grąžinimo į sistemą fiksavimas ir vartotojų informavimas (angl. *check-in/check-out*), leidžiantis tik vienam vartotojui redaguoti dokumentą.
 - Bendra visos sistemos ir objektų lygio sauga, paremta rolėmis, grupėmis ir pozicijomis.
 - Pilnatekstė paieška tekstiniuose failuose.
 - Dokumento būsenos (saugomos metaduomenyse) valdymas (pvz.: jei jis baigtas, jis gali būti prieinamas kitiems vartotojams).
 - Dokumentų klasifikavimas į katalogus ir kolekcijas. Jei palaikomas proceso valdymas, kaip užduočių sąrašai skirtingiems darbuotojams, einama per šią darbų srautą (angl. *workflow*) sistemą.
 - Įvairios ataskaitos, susijusios su dokumentų klasifikavimu.
 - Metaduomenų indeksacija, leidžianti valdyti darbų srautą pagal metaduomenis (pvz.: pagal autorių, pavadinimą, sukūrimo datą).

Visos šios savybės pagerina dokumento panaudojimo produktyvumą ir efektyvumą, dokumento vartotojui sugaišťtant nedaug laiko.

3. Funkcijos dokumento užbaigimo etape. Šiame galutiniame etape organizacija kuria taisykles, kurios užtikrina dokumento saugojimą tuo laikotarpiu, kol jis yra naudingas. Šios taisyklės apibrėžia, kada dokumentai yra pašalinami iš archyvo. Atskaitos tašku dokumento naudingumo laikui įvertinti laikomas dokumento sukūrimo laikas, jo tipas ir kt.

- Dokumentų grupavimas pagal pasirinktą požymį.
- Dokumentų publikavimas internete. HTTP ir XML leidžia dokumentus publikuoti interneto svetainėse ir jiems būti prieinamiems el. komercijos aplinkoje.
- Atsarginių kopijų darymas (angl. *Backup*).
- Dokumento atkūrimas. Rastas dokumentas gali būti peržiūrimas ekrane, atspausdintas, persiunčiamas ir kt. Su struktūrizuotais dokumentais susijusi programa gali būti paleidžiama automatiškai.

Daugelis DVS turi papildomų funkcijų, kurios palaiko dokumentų naudojimą skirtingose srityse. Kai kurios jų taip pat palaiko administracinį darbą, kaip:

- Vartotojų ir paskyrų administravimą.
- Atsarginių kopijų darymą ir atstatymą. Svarbu, kad DVS palaikytų augančią atsarginių kopijų darymo strategiją.
- El. pašto integravimas, pasikeitimų pranešimams siuntinėti ir palaikantis dokumentų paskirstymą DVS sistemoje.
- Skaitmeninis parašas.
- Pranešimų sistema, informuojanti, kad tam tikras dokumentas yra dar tam tikroje būsenoje atitinkamą laiko tarpą.

Šis DVS funkcijų sąrašas nėra pilnas, bet ir nėra pilnai realizuojamas kiekvienoje DVS sistemoje. Priklausomai nuo organizacijos veiklos srities, nuo daugelio jau realizuotų savybių, kitos sąvokos, tokios kaip archyvavimo sistema, portalas, grupinio darbo ar darbų srautų sistemos yra naudojamos.

2.3. DVS kitų sistemų kontekste

Dokumentai ir dokumentų valdymas naudojami skirtingiems tikslams ir turi skirtingas savybes. Todėl dauguma sistemų vystėsi DV aplinkoje, skirtoje atitinkamam tikslui.

DVS sistemos šiame kontekste yra kaip pagrindinė paslauga. Dabar egzistuoja paslaugai orientuotos architektūros (angl. SOA – *service-oriented architecture*), kai taikomoji

programa veikia integruota į kitą sistemą. SOA – tarpusavyje susietų paslaugų rinkinys. Jos bendradarbiauja viena su kita, paprastai perduodant duomenis arba įtraukiant keletą papildomų paslaugų koordinuojant tam tikrą veiklą [7].

Ši strategiją verslo partneriams atveria aiškią bendradarbiavimo perspektyvą ir suteikia akivaizdžius abipusius privalumus, vieniems koncentruojantis programinės įrangos platformos projektavime, kitiems gilinant žinias konkrečių taikomųjų sprendimų kūrime.

Sistemos, naudojančios dokumentų valdymą kaip paslaugą užduočiai įvykdyti:

- Standartiniai ir individualūs PĮ paketai.
- Interneto taikomosios programos ir portalai.
- Turinio valdymo sistemos.
- Darbų srautų sistemos.
- Grupinio darbo sistemos.
- Žinių valdymo sistemos.

2.3.1. Standartiniai ir individualūs PĮ paketai

Beveik kiekviena PĮ realizacija susijusi su dokumentais. Nors dauguma duomenų esančių dokumentuose vaizduojami reliacinėse duomenų bazėse (pvz. Produkto aprašymas), būtina, kad kiekviena generuojama informacija būtų papildomai saugoma dokumentų formoje.

DVS integravimui į specifinį PĮ paketą egzistuoja įvairūs sprendimai ir reikalavimai:

- Dokumentams, katalogams abiejose sistemose raktai turi būti identiški: ryšys 1:1.
- PĮ sukurti dokumentai turi būti įtraukiami į DVS.
- Išoriniai dokumentai, susiję su verslu, turi būti nuskanuoti arba importuoti į DVS.
- Vykdam užklausas su verslu susijusioje duomenų bazėje, turi būti naudojami ryšiai su visais naudojamais dokumentais.

Tokiu būdu iš vartotojo pusės neįmanomas skirtumas tarp taikomosios programos ir DVS.

2.3.2. Interneto taikomosios programos ir portalai

Kad informacinė sistema būtų produktyvesnė ir naudingesnė vartotojui, kuriamos internetinės sąsajos. Visi reikalingi elementai (duomenų bazės, dokumentų valdymas, įmonės verslo valdymas ir kt.) turi būti integruojami portale. Portalo PĮ dažnai būna komercinio produkto dalis, bet gali egzistuoti ir kaip atskiras/savarankiškas produktas.

2.3.3. Turinio valdymo sistemos

Turinio valdymo sistema (angl. *Content Management System*) – įvairūs programiniai įrankiai, supaprastinantys informacinių sistemų turinio (tekstinį ir grafinį) valdymą taip, kad sukuriant keičiant turinį ar jo struktūrą nereikėtų jokių specialiųjų (programavimo) žinių.

Vienas pagrindinių turinio valdymo sistemos tikslų – užtikrinti sklandų daugelio vartotojų kooperatyvinį darbą. Taikoma internetinėms svetainėms, laikraščių leidyboje ir pan.

TVS sistemoje svarbi dokumente kaupiama informacija. Naudojamos taisyklės, kaip projektuoti informaciją, tačiau ji atskiriama nuo turinio, tai leidžia turėti daug privalumų, kai viskas susiejama į didesnę publikacijų rinkinį. Tokios sistemos turi tam tikrą duomenų srauto koncepciją, kuri nustato kur naujas turinys bus nukreiptas DVS sistemoje.

2.3.4. Darbų srautų sistemos

Darbų srautas (angl. *workflow*) – tai terminas, naudojamas užduotims aprašyti, procedūriniais žingsniais, į procesą įtrauktiems organizacijoms bei žmonėms, įeinančiai bei išeinančiai informacijai ir priemonėms, kurios yra reikalingos kiekviename verslo proceso etape.

Darbų srautų sistema padeda organizacijai specifikuoti, vykdyti, stebėti ir koordinuoti darbų srautus, paskirstytus organizacijos viduje. Tokia sistema turi du pagrindinius komponentus: darbų srautų modeliavimo komponentą, kuris leidžia administratoriui ir analitikui apibrėžti procesus ir veiklas, juos analizuoti ir imituoti, antra, – darbų srautų vykdymo komponentas. Jį sudaro vykdymo sąsaja ir vykdymo aplinka, kuri koordinuoja bei atlieka procesus ir veiklas.

Dokumentai organizacijos viduje paskirstomi skirtingiems darbuotojams pagal tam tikras taisykles ir atributus. Kiekvienas vykdytojas pradeda su tam tikru dokumentu susijusį procesą.

Paprasta DVS nepalaiko šios savybės, pagal kurią keli vartotojai gali bendradarbiauti ir keliais etapais apdoroti dokumentą. Tokio darbų srauto valdymui ir palengvinimui DVS sistemos naudojamos kartu su darbų srautų sistema.



3 pav. Darbų srautas

Elementas	Paaškinimas
Vartotojas (angl. <i>Customer</i>)	Asmuo ar padalinys, kuris pateikia užsakymą tam tikram darbui atlikti. Tai gali būti išorinis ar vidinis organizacijos atžvilgiu.
Atlikėjas, vykdytojas (angl. <i>Performer</i>)	Asmuo ar padalinys, kuris sutinka atlikti vartotojo užsakytą darbą.
Pasiruošimo etapas (angl. <i>Preparation</i>)	Vartotojas kreipiasi į vykdytoją dėl tam tikro darbo atlikimo arba atlikėjas pasiūlo vartotojui atlikti tam tikrą darbą.
Tarimosi etapas – derybos (angl. <i>Negotiation</i>)	Vartotojas ir vykdytojas suderina darbo pobūdį, atlikimo terminą ir vartotoją tenkinančias sąlygas.
Atlikimo etapas (<i>Performance</i>)	Vykdytojas atlieka darbą ir užpildo ataskaitą.
Priėmimo etapas (angl. <i>Acceptance</i>)	Vartotojas patvirtina, kad jis atliktu darbu patenkintas, arba atsisako priimti darbą.
Atlikimo laikas	Tai galutinis darbo atlikimo laikas, apskaičiuojamas sudėjus kiekvieno etapo atlikimo laikus.
Patinkančios sąlygos (angl. <i>Conditions of Satisfaction</i>)	Vartotoją tenkinantys darbo atlikimo terminai ir kriterijai.

2.3.5. Grupinio darbo sistemos

Grupinio darbo sistema integruoja darbą, kai su vienu dokumentu dirba keli vartotojai prie skirtingų kompiuterių. Kaip ir darbų srautų sistema ji palengvina bendradarbiavimą, tačiau jos skiriasi:

2 lentelė. Grupinio darbo ir darbų srautų sistemų skirtumai

	Darbų srautų sistema	Grupinio darbo sistema
Taikymo sritis	Sudėtingi verslo procesai	Grupių valdomos užduotys
Nukreiptos užduotys	Laiku ir įvykiais pagrįsti procesai	Sprendimai, projektai, komandinis darbas
Savybės	Vykdomas į automatizuotą procesą įjungiant skirtingas sistemas	Užduotis valdoma per DVS – infrastruktūros įgyvendinimą.
Proceso struktūra	Gerai struktūrizuota	Mažai struktūrizuota
Stebimas objektas	Procesas, užduotis	darbuotojas
Palaikomas	Organizacijos tikslų siekimas	Grupės ir darbo vietos
Aktyvus elementas	Sistema	Vartotojas
Organizacinis elementas	Sistema	Vartotojas
Kontroliuojamas elementas	Sistema	Vartotojas
Struktūra	Centralizuota	Lokali

Grupinio darbo sistemos veiklos – tai procesai, tačiau rezultatų struktūra negali būti iš anksto apibrėžiama. DVS naudojama šioje aplinkoje turi leisti laikinus pakeitimus katalogų struktūroje. Taip pat čia reikalingi lankstūs paieškos įrankiai.

2.3.6. Žinių valdymo sistemos

Žinių valdymas taiko kolektyvines žinias ir darbuotojų sugebėjimus organizacijos tikslams siekti. Organizacijos žinios neapsiriboja, vien tomis žiniomis kurias turi jos darbuotojai, bet ir tomis, kurios yra organizacijos procesuose ir jos kultūroje. Strategiškai suplanuotas žinių valdymas leidžia organizacijai įgauti didesnio lankstumo rinkoje ir tapti konkurencingesne.

2.4. Dokumentų valdymo standartai

Kaip ir kitos informacijos valdymo sistemos, DVS turi nemažai vienas kitą papildančių ir tarpusavyje konkuruojančių standartų, apibūdinančių DVS architektūrą. Keletas žinomų standartų [4]:

- ODMA (angl. *Open Document Management API*) – tai bandymas palengvinti daugiaplatforminį (angl. *cross-platform*) ir daugiaprograminį (angl. *cross-application*) bendravimą, standartizuojant prieigą prie DVS sistemų iš klientų taikomųjų programų ir naudojant API – taikomųjų programų sąsają (angl. *Application Programming Interface*). Ši sąsaja leidžia programoms lanksčiai bendrauti su DVS be jokių sunkiai koduojamų sąryšių tarp taikomosios programos ir DVS. [9] Nors standartas orientuotas į daugiaplatformiškumą, buvo įrodyta, kad jis per daug priklausomas nuo *Microsoft* technologijų. Kita iškilusi problema – web klientų palaikomumas. Apibrėžtas transporto sluoksnis tokiose aplinkose nebuvo pritaikomas, todėl tik kelios DVS palaiko šį standartą. ODMA standartą palaiko *Borland*, *Documentum*, *Interleaf*, *Novell*, *Oracle*, *PC DOCS*, *SoftSolutions*, *Sybase*, *WordPerfect*, and *XSoft*.
- Shamrock (angl. *Shamrock Document Management Coalition*) – standartas, kitaip nei ODMA Shamrock standartas specifikavo vidurinių (angl. *middleware*) programinės įrangos sluoksnį, kuris yra įterpiamas tarp kliento taikomosios programos ir DVS serverio bei teikiantis įrankų rinkinį dokumentų saugumui, administravimui ir prieigai. Priešingai nei ODMA, kur skirtingos programos turi priėjimą prie tos pačios DVS, Shamrock standarte programai turėti prieigą prie daugelio DVS sistemų. Pagrindiniai Shamrock partneriai yra *Saros* ir *IBM*, kurių DVS technologija paremta daugiausia šiuo standartu. Kiti partneriai: *Adobe*, *EDS*, *Hewlett-Packard*, *Microsoft*,

Verity, ViewStar, Wang, taip pat ir ODMA nariai Documentum, Interleaf, PC DOCS, Sybase and XSoft.

- DEN (angl. *Document-Enabled Network*) – rungtyniaujantis su shamrock, inicijuotas partnerystės tarp Novell ir Xerox, kuris taip pat kūrė vidurinę sluoksnį palengvinti taikomųjų programų vystymą paskirstytam dokumentų valdymui.
- OPENDOC – PĮ komponentinė architektūra, kuri apibrėžia sąsają nepriklausomai programinei įrangai dirbti kartu viename dokumente. OpenDoc – atviras standartas sudėtiniais dokumentams, sudarytiems iš daug skirtingų tipo elementų, patalpintų viename faile (pvz., lentelės, grafikai, tekstas, video, garsas, 3D grafika). Šiame standarte apibrėžiamos taisyklės palengvina sąveikaujančių PĮ komponentų, susijusių su tuo pačiu dokumentu, kūrimą.
- OLE (angl. Object Linking and Embedding): tai Microsoft standartas, leidžiantis objektams iš vienos taikomosios programos būti įterptiems į kitą programą, išsaugant originalų įterpiamo failo formatą ir nuorodą į programą, kuria jis buvo sukurtas.
- OGM (angl. Object Management Group) – pramonės korporacijoms, sukūrus karkasą OMA (angl. Object Management Architecture) objektiškai orientuotoms aplinkoms specifiškai. Jis iš dalies specifiškai modelį, klasifikuojantį objektiškai orientuotų aplinkų komponentus, sąsajas bei protokolus.
- WfMC (angl. Workflow Management Coalition) – nekomercinė darbų srautų vartotojų, mokslinių grupių organizacija, kuri skatina ir vysto darbų srautų PĮ standartus.

2.5. Dokumentų valdymas internete

Galimybė jungtis prie DVS už organizacijos ribų per interneto naršyklę šiandien tapo aktuali. WWW naudoti kaip transporto sluoksnį dokumentų apsiuntimui buvo logiškas žingsnis. Šalia pagrindinio duomenų persiuntimo HTTP protokolo realizuota aibė technologijų, pritaikytų dokumentų valdyme.

2.5.1. Saugumas

Kiekvienas vartotojas sistemoje yra identifikuojamas ir autentifikuojamas. Visi dokumentai ir vartotojai sistemoje turi tam tikras saugumo žymes, kurios nurodo, kokias teises turi kiekvienas vartotojas ir kokias saugumo žymes turi kiekvienas dokumentas. Kadangi visi vartotojų veiksmai sistemoje yra registruojami, sistemos administratorius gali peržiūrėti visus sistemos užfiksuotus įvykius. Egzistuoja bendri dokumentai, kurie yra prieinami visiems

darbuotojams, privatūs, kurie naudojami tik kurios nors vienos darbo grupės viduje ir dokumentai

Be vartotojų autentifikacijos saugumui užtikrinti gali būti naudojami šie protokolai: SSL (angl. *Secure Sockets Layer*) ir TSL (angl. *Transport Layer Security*) – saugumo protokolai užkoduojantys pateiktus asmens duomenis, todėl informacija esanti sistemos duomenų bazėje yra saugi ir nepasiekama pašaliniams asmenims.

2.5.2. Duomenų perdavimo optimizavimas

Kadangi internetinio ryšio greitis ne visada didelis palyginus su vietiniu organizacijos tinklu, keletas technologijų naudojama prieigai prie nutolusių dokumentų pagreitinti:

- Duomenų srautų suspaudimas naudojant suspaudimo bibliotekas.
- Laukimo trukmės minimizavimas naudojant pastovų ryšį, srautinį darbą (angl. *pipelining*) – keleto užklausų siuntimas nelaukiant atsako, arba keletą atsakų sujungiant į vieną.

2.5.3. WebDAV

WebDAV [18] (angl. *Web-based Distributed Authoring and Versioning*) – HTTP protokolo papildymas, leidžiantis nutolusiems vartotojams kartu taisyti bei tvarkyti failus serveryje. WebDAV išplečia HTTP, standartinei sinchroninio bendradarbiavimo internete infrastuktūrai sukurti. WebDAV išplėtimai per bendras skirtingų saugyklų tipų sąsajas palaiko įvairaus turinio publikavimą ir internetą paverčia panašiu į tinklu pasiekiamą failų sistemą. WebDAV ypatybės yra *pavadinimo* ir *reikšmės* poros, – pavadinimas – URL eilutė, o reikšmė – gerai suformuotos išplėstinės žymėjimo kalbos (angl. XML) seka. URL pavadinimas leidžia naujų ypatybių pridėjimui, be jokio registravimosi serveryje, tuo tarpu XML reikšmės leidžia įvesti duomenis per struktūrizuotą žymėjimą ir išplėsti pridėdant naujas žymes. WebDAV palaikomos papildomos HTTP savybės:

- Įvairių resursų redagavimas per WWW.
- Vartotojų prieigos kontrolė.
- Dokumentų versijavimas.
- Failų sistemos operacijos: kopijuoti, perkelti, ištrinti ir užrakinti.
- Išplėstas metaduomenų modelis.

Tarp kliento ir serverio perduodamų duomenų kodavimui naudojamas XML formatas.

Specifikacija nusako tam tikrą objektinį modelį, kurį sudaro:

2.5.4. XML duomenų vaizdavimo standartas

XML (angl. *eXtensible Markup Language*) – formatas skirtas struktūrinių dokumentų aprašymui, naudojant žymes. Tai taisyklių, nurodymų ir susitarimų rinkinys, apibrėžiantis, kaip tekstiniu formatu aprašyti struktūrizuotą informaciją. XML kūrimas buvo pradėtas 1996 metais, o 1998 m. vasarį World Wide Web Konsorciumas (W3C) išleido pirmąją rekomendaciją. XML naudoja standartinius segmentus failo struktūrai ir turiniui nustatyti. Turint tuos pačius XML segmentus visuose failuose galima efektyviai indeksuoti, ieškoti, derinti ir pakartotinai naudoti tekstinius duomenis. Tai, kad XML tekstai nėra ribojami tam tikrų kompiuterinių kalbų, leidžia keistis informacija tarp įprastai nesuderinamų sistemų [10].

XML dokumentai sudaryti iš žymėjimų ir laisvo teksto. Žymėjimą sudaro žymės ir jų atributai. XML standartas nustato bendras jų sudarymo taisykles, bet neapibrėžia konkrečių žymių rinkinio ar jų reikšmės. XML naudoja žymes tik informacijos atskyrimui, interpretavimui ir suprasminimui palikdama programai, kuri naudoja dokumentą. XML tai tik būdas struktūrizuoti aprašyti informaciją [10]. XML neturi iš anksto apibrėžtų žymių, jas aprašo dokumento autorius arba naudojami tam sukurti specifiniai standartai. XML yra metakalba, naudojama apibrėžti įvairaus taikymo žymėjimo kalboms. HTML standartas apibrėžia, ką reiškia kiekviena žymė bei atributas ir kaip juos atvaizduoti naršyklės lange, tuo tarpu XML aprašo tik informacijos turinį. Duomenų interpretacija ir atvaizdavimo būdai nuo turinio atskirti. XML dokumentai gali būti naudojami pačiose įvairiausiose srityse, tokiose kaip el. komercija, bendravimas tarp verslo partnerių ar organizacijos viduje.

Naudojant XML galima sukurti duomenų struktūras, kurios aprašo jose esančių duomenų turinį, nepriklausomai nuo to, kaip turinys bus atvaizduotas. Nors XML griežtai specifikuoja sintaksę, tačiau leidžia laisvai apibrėžti XML dokumentų prasmę. XML leidžia susikurti savą gramatiką arba naudotis jau sukurta gramatika, pavyzdžiui, tokia, kuri naudojama konkrečioje probleminėje srityje (pvz. MathML (angl. *Mathematical Markup Language*) skirtas aprašyti matematiniams tekstams, ebXML (angl. *Electronic Business XML*) yra standartas verslo informacijos mainams ir kt.).

XML tapo vienu populiariausiu duomenų aprašymo formatu, tiek duomenų mainuose, tiek į dokumentus orientuotose sistemose. Štai pagrindiniai bruožai, kurie lėmė tokį populiarumą:

- Galimybė aprašyti bet kokio sudėtingumo informaciją.
- Prireikus išplėsti aprašomos informacijos tipus, galima pridėti naujų žymių.
- Galima patikrinti ar duomenys struktūriškai teisingi.
- Duomenis galima transformuoti į įvairius formatus.

- Nepriklauso nuo platformos ar gamintojo.

2.5.3.1. XML korektiškumo tikrinimas

Atitinkantis XML sintaksę dokumentas vadinamas taisyklingu. Susidūrę su netaisyklingais dokumentais, įrankiai turėtų pranešti apie klaidas, nes tai jau nebėra XML dokumentas. Tačiau norint žinoti, ar dokumentas yra prasmingas, neužtenka vien taisyklingumo tikrinimo. Teisingi duomenys turi atitikti tam tikrą loginę struktūrą, kurią aprašo schemų kalbos. Būtent schemų kalbomis aprašomos naujos XML kalbos. Bet kuris dokumento korektiškumą tikrinantis analizatorius gali užtikrinti, kad nagrinėjamas dokumentas atitinka nustatytas kalbos taisykles. Kadangi, kartu su XML standartu pateikta schemų kalba DTD turėjo mažai galimybių, įvairios organizacijos sukūrė daug įvairių schemų kalbų. Gramatikai apibrėžti naudojami du formatai:

- DTD (angl. *Document Type Definition*) – dokumento tipo apibrėžimas, sudarytas iš taisyklių apie XML dokumento struktūrą ir turinį. DTD išvardina visus galimus elementus, jų pasirodymo tvarką ir atributus. Taigi, DTD nustato hierarchinę dokumento struktūrą, – tai svarbiausia, tikrinant XML dokumento korektiškumą. Nors palyginus su kitomis schemų kalbomis DTD turi mažai galimybių, bet jis yra populiarus ir dėl savo paprastumo. Nors dokumento tipo apibrėžimo metodas puikiai tiko pirmiesiems bandymams, bet kuriant sudėtingas sistemas iš karto paaiškėjo jo trūkumai. Viena pagrindinių DTD problemų – tai nėra korektiškas XML dokumentas. Todėl darbui su juo nebetinka standartiniai XML analizatoriai. Be to, vienam elementui aprašytos taisyklės negali būti panaudotos kito elemento aprašymui.
- XML schema – W3C standartas, patvirtintas 2001 metais ir išsprendžiantis DTD schemas problemas. XML Schema dokumentai yra taisyklingi XML dokumentai, ji palaiko vardų sritis bei turi priemonės elementams priskirti tipus, taip leisdamas tikrinti dokumentus ne tik pagal elementų struktūrą, bet ir pagal jų turinį. Vienas labiausiai išskiriančių iš kitų schemų kalbų XML Schema savybė yra tipų paveldimumas [13].

Taigi XML dokumentų struktūrai tikrinti dažniausiai naudojamas XML schemas formatas, kadangi jis yra pranašesnis ir patogesnis nei DTD. XML schemas privalumai:

- schemas naudoja XML sintaksę;
- schemoms apdoroti gali būti naudojamos įprastinės XML programos;
- schemas palaiko standartinius duomenų tipus ir galima susikurti savus;
- schemose gali būti išoriniai elementai, kurie naudojami visame XML dokumente ir vietiniai, įgyjantys specifinę reikšmę pagal situaciją.

XML schema aprašo XML dokumentų struktūrą. Ji nusako elementus, kurie gali būti naudojami XML dokumentuose. Jei XML dokumentas naudoja schemą, tai neaprašytų elementų negalima vartoti. XML dokumentų tikrinimas reiškia, kad išoriniai duomenys turi laikytis taisyklių (gramatikos), kurios yra aprašytos schemoje. Schema užtikrina, kad XML dokumentas yra tokio formato, kokio tikimasi. XML schema yra galinga ir sudėtinga XML dokumentų struktūros kūrimo ir tos struktūros gramatikos patvirtinimo priemonė [13].

2.5.3.2. Standartinės sąsajos

Norint patikrinti XML dokumento korektiškumą ar jį panaudoti, dokumentas pirmiausia turi būti išanalizuotas. Tai atlieka programos vadinamos XML analizatoriais (angl. XML parser), jos sugeba nuskaityti ir (dažniausiai) patikrinti XML dokumento duomenų korektiškumą. Analizatorius įgalina programą naudotis XML dokumente esančiais duomenimis. Pagrindiniai du analizatorių standartai [14]:

- SAX (Simple API for XML) – įvykiais pagrįsti analizatoriai, kurie skaitydami XML dokumentą generuoja įvykius ir praneša apie tai programai. Pvz., SAX įvykis yra iškviečiamas kai aptinkamas XML elemento pabaiga. Įvykyje yra aprašomas pasibaigusio elemento pavadinimas. Pirmiausia realizuotas java programavimo kalboje, o dėl savo paprastumo vėliau ir kitose. Norint naudotis SAX metodu, reikia įgyvendinti įvykių apdorotoją, kuri naudos nagrinėtojas skaitydamas dokumentą. SAX yra greitas XML dokumentų nagrinėjimo metodas, ypač tinkantis dideliems dokumentams, kurie negali tilpti atmintyje.
- DOM (Document Object Model) – dokumento objektinis modelis. Tai oficialus W3C standartas, palaikomas daugelio programavimo kalbų bei įgyvendintas interneto naršyklėse. Naudojant DOM, nagrinėtojas nuskaityto visą XML dokumentą ir sukuria atmintyje medžio pavidalo struktūrą, kurioje yra dokumento struktūra ir turinys. Iškviestais analizatorių programai gražinama rodyklė į šią struktūrą. Programa gali manipuluoti dokumentu, pertvarkydama, pridėdama ar ištrindama atšakas, bei vėl užrašyti objektinį modelį korektišku XML failu. Nors DOM analizatorių dažniausiai paprasčiau išsikviesti, tačiau dirbti su DOM medžiais yra gana sudėtinga. Be to, jis žymiai lėtesnis ir naudoja daugiau resursų už SAX analizatorius. Tačiau DOM gali būti efektyviai panaudotas su mažesniais XML dokumentais, situacijose, kai greitis nėra ypač svarbus programos reikalavimas. Dažnai SAX nagrinėtojas būna sudedamoji DOM nagrinėtojo dalis.

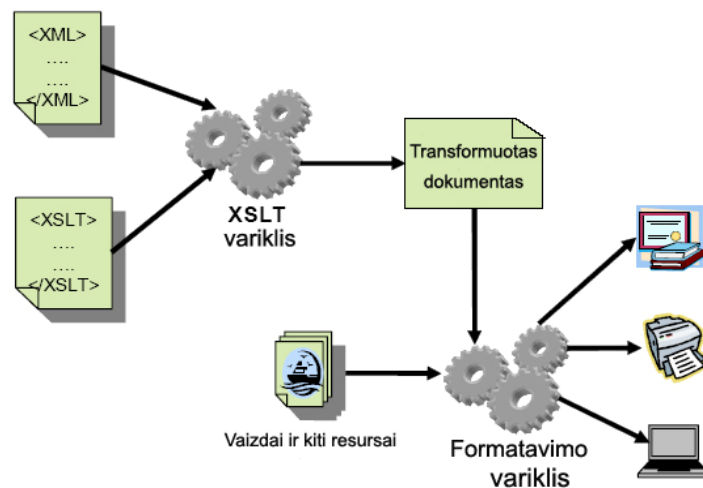
2.5.3.3. XML transformavimas

Vienas pagrindinių XML technologijos privalumų – tai galimybė konvertuoti XML duomenis iš vienos formos į kitą, naudojantis bendrinėmis priemonėmis.

XSL (angl. *eXtensible Stylesheet Language*) yra W3C pasiūlytas standartas, valdantis XML dokumento atvaizdą naršyklėje. XSL sukurtas remiantis CSS2 (angl. *Cascading Style Sheets*), bei DSSSL (angl. *Document Style Semantics and Specification Language*) XSL – kalba skirta stiliams (stylesheet) aprašyti. XSL susideda iš:

- XSLT (angl. *eXtensible Stylesheet Language Transformations*) – išplečiama stilių kalba transformacijoms, skirta transformuoti XML dokumentą į kitokio formato dokumentą. XSLT gali būti naudojama XML dokumentą transformuoti į kitą XML dokumentą.
- XSL *Formatting Objects* – kalba, kuri formuoja XML dokumento vaizdą po XSLT transformacijos (pvz., pdf dokumento generavimui iš XML dokumento). Nors XML-FO dokumentus galima kurti ir rankomis, bet dažniausiai tam yra rašoma XSL transformacija ir formatavimo objektai generuojami iš duomenų failų (žr. 4 pav).

XSLT gali transformuoti XML dokumentą į dokumentą, kuris gali būti atvaizduotas naršyklėje. Pavyzdžiui XML dokumentą galima transformuoti į HTML, kurį be problemų galima atvaizduoti naršyklėje. Ši transformacija kiekvieną XML elementą atvaizduoja į HTML elementą. XSLT taip pat gali pridėti ar išmesti, perrūšiuoti elementus transformacijos rezultato dokumente. XSLT naudoja XPath surasti elementus XML pradiniam dokumente, kurie atitinka tam tikras sąlygas ir po to atlieka elementų transformaciją. XSL naudoja vieną ar kelis šablonus, kurie apibrėžia, kaip atvaizduoti XML elementus. XSL šablonai aprašomi naudojantis šablono (*template*) elementu [15].



4 pav. XSL architektūra

2.5.3.4. XML dokumentų dalių išskyrimas ir apdorojimas

Atskirų XML dokumento dalių išskyrimas ir manipuliavimas jomis yra svarbios veiklos, todėl sukurta daug susijusių specifikacijų, kurios apibrėžia, kaip nurodyti tam tikras XML dokumento vietas ar atlikti informacijos paiešką su parametrais. Dvi svarbiausias šios srities kalbos [17]:

- XPath yra kalba, skirta XML dokumento dalių adresavimui, kurios sintaksė primena kelius failų sistemoje ar URL. XPath palaiko funkcijas darbui su atrinktais duomenimis. Funkcijos skirtos gauti informaciją apie dokumento atšakas bei darbui su eilutėmis, skaičiais ir loginiais kintamaisiais. Programuotojai gali plėsti prienamų funkcijų biblioteką. XPath naudoja kompaktišką, ne XML sintaksę, kad galėtų būti naudojama XML atributuose arba nuorodose. XPath dirba su abstrakčia, logine XML dokumento struktūra, o ne tekstine sintakse. Ji skirta darbui su vienu XML dokumentu, į kurį žvelgia kaip atšakų medį. XPath užklaustos rezultato reikšmės konceptualiai irgi yra laikomos atšakomis.
- XQuery – tai bandymas pateikti užklausų kalbą, kuri būtų panašaus funkcionalumo ir turėtų stiprų formalizmą, kaip SQL reliacinėms duomenų bazėms. XQuery yra funkcinė kalba, kurioje kiekviena užklausa yra išraiška. Skirtingų išraiškų sintaksė ir semantika labai skiriasi, todėl, kad XQuery kūrimui turėjo įtakos daug skirtingų sistemų. XQuery turi sudėtingą tipų sistemą, kuri paremta XML Schema duomenų tipais ir, kitaip nei XPath, palaiko manipuliavimą dokumento atšakomis. Taip pat XQuery duomenų modelis sukurtas darbui ne tik su pavieniu XML dokumentu, bet ir taisyklingu jo fragmentu, dokumentų seka ar dokumentų fragmentų seka.

2.5.3.5. XML duomenų bazės

XML dokumentai yra duomenys, todėl nenuostabu, kad kuriama daugybė technologijų patogiam tų duomenų išsaugojimui. Dar vis populiariu laikyti XML dokumentus tekstiniu pavidalu failų sistemoje, bet tai neefektyvu ir riboja duomenų panaudojamumą. Alternatyvus metodas yra PDOM (angl. *Persistent Document Object Model*), kuris dokumentą paverčia DOM struktūra ir išsaugo jį failų sistemoje dvejetainiu pavidalu. Taip išvengiama daugiausiai resursų naudojančio etapo – analizavimo.

Taip pat kuriamos duomenų bazės, kurių gimtasis duomenų formatas yra XML. Didesni duomenų bazių kūrėjai, kaip Oracle, IBM, kuria XML išplėtimus savo duomenų bazėms, nes įprastos reliacinės duomenų bazės sunkiai susidoroja su mažai struktūrizuotais XML

duomenimis, o specialios XML užklausų kalbos smarkiai lenkia SQL, dirbant su hierarchiniais duomenimis.

Kaip teigiama [16, 17], XML duomenų bazės turi šiuos privalumus prieš reliacines ar objektines duomenų bazines:

- XML duomenys į duomenų bazę yra pakraunami tiesiogiai, jais nereikia niekaip manipuluoti ar ištraukti iš dokumentų prieš saugojant.
- Užklausos gražina XML dokumentus arba jų fragmentus, todėl išlaikoma hierarchinė XML dokumento struktūra.

Programuotojams XML duomenų bazės sprendžia šias dvi problemas:

- Reikia pernelyg daug resursų, norint sukurti atminties struktūras didelių dokumentų medžiams ir vykdyti jiems užklausas. Dirbant su XSLT ar DOM, apdorojant 100 megabaitų failą, gali prireikti net 1.2 gigabaito atminties. XML duomenų bazės gali laikyti gigabaitus XML duomenų, laisvai adresuojamų XPath užklausomis.
- Norint pasiekti tam tikrą dokumento fragmentą, paprastai reikia išnagrinėti visą XML failą. Mažiams dokumentams tai nėra problema, tačiau dideliems dokumentams, ypač siunčiamiems per tinklą, tai ryšio ir procesoriaus švaistymas. XML duomenų bazė gali rasti reikalingus dokumento fragmentus be XML dokumento nagrinėjimo.

2.6. Dokumentų valdymo sistemų modeliai

Analizuojant daugelį dokumentų valdymo sistemų, galima išskirti du pagrindinius modelius, vieno autoriaus [19] apibrėžtus kaip:

- Integruotas dokumentų valdymas arba dar vadinamas hierarchiniu modeliu [22]. Dokumentai čia taikliai pavadinti „juodomis dėžėmis“, nes jų struktūra nežinoma. Tokio modelio tikslas – įgalinti lengvą dokumentų atrinkimą, naudojant nuorodų informaciją.
- Dokumento modeliu pagrįstas modelis. Visa informacija, kuri tradiciškai buvo saugoma grafiniuose ir tekstiniuose dokumentuose yra įrašoma į vieną integruotą duomenų bazę, iš kurios informacija gali būti išgaunama, naudojant užklausas, o dokumentai gali būti generuojami automatiškai, sujungiant kelių dokumentų dalis.

2.6.1. Hierarchinis dokumentų valdymo modelis

Hierarchija – tai organizacijos struktūra su tam tikra, vienintele esybe hierarchijos viršuje. Kiekviena esybė turinti savyje tam tikrą struktūrą, – kitų esybių rinkinį [22]. Hierarchiniame

dokumentų valdymo modelyje, šios esybės vadinamos katalogais, savo viduje kaupiančiais dokumentus bei kitus dokumentus (pakatalogius). Taigi tai bet kuri tipinė failų sistema, kuria palaiko visos šiuolaikinės operacinės sistemos.

Šio modelio realizavimas yra gana nesudėtingas, štai kodėl šios dokumentų valdymo sistemos šiuo metu yra labiausiai paplitusios.

2.6.1.1. Modelio trūkumai

DVS sistemos, realizuotos hierarchiniu failų sistemos principu turi keletą trūkumų, susijusių su dokumento erdvės, – dokumentų kolekcijos, valdymu [20, 21, 22]:

- Modelis teigia, kad vienas dokumentas gali būti laikomas tik vienoje vietoje. Tačiau akivaizdu, kad dokumentas logiškai gali priklausyti kelioms dokumentų hierarchinės struktūros šakoms (pvz., dokumentas gali būti susijęs su keliomis organizacijos veiklomis), hierarchinėje sistemoje jį įmanoma laikyti tik vienoje vietoje ir tai labai trukdo dokumentams sisteminti. Vienas iš sprendimų [21] būtų sukurti reikalingo failo kopiją ir patalpinti kitame kataloge, bet tai iššaukia informacijos dubliavimą.
- Reikalaujama dokumentų išsidėstymo struktūra kinta laike, hierarchiniame modelyje didelių failų rinkinių pertvarkymas vargina.
- Ribota paieška ir komplikotas dalinimasis. Dažniausiai dokumentai į sistemą įkeliami naudojantis vienais sistemavimo kriterijais, o ieškomi naudojantis kitais.
- Negalima manipuluoti dokumentuose esančia informacija, tad ir iš kelių dokumentų sudaryti vieno naujo neįmanoma.

2.6.1.2. Alternatyvūs sprendimai

Dėl anksčiau minėtų hierarchinio modelio apribojimų, dokumentų valdymo sistemoms literatūroje siūloma keletas alternatyvų. Šie modeliai yra orientuoti į dokumento turinį ir didžiausią dėmesį skiria dokumento struktūrai.

Norint pašalinti prieš tai minėtus trūkumus, siūloma tobulinti vartotojo sąsają, pvz., išnaudoti erdvę, sukuriant nuorodas į tą patį dokumentą iš skirtingų sistemos vietų, tobulinti vartotojo sąsają [21], naudoti automatizuotus komponentus, pvz., kuriant užklausas, leidžiančias atrinkti dokumentus pagal tam tikrą kriterijų, arba naudoti lankstesnes informacines struktūras – metaduomenis. Daugelis autorių [21, 22, 23] metaduomenis laiko vienintele aprašymo priemone dokumentams, kurie neturi struktūros arba kurių struktūra yra nežinoma.

2.6.2. Metaduomenimis grįsti modeliai

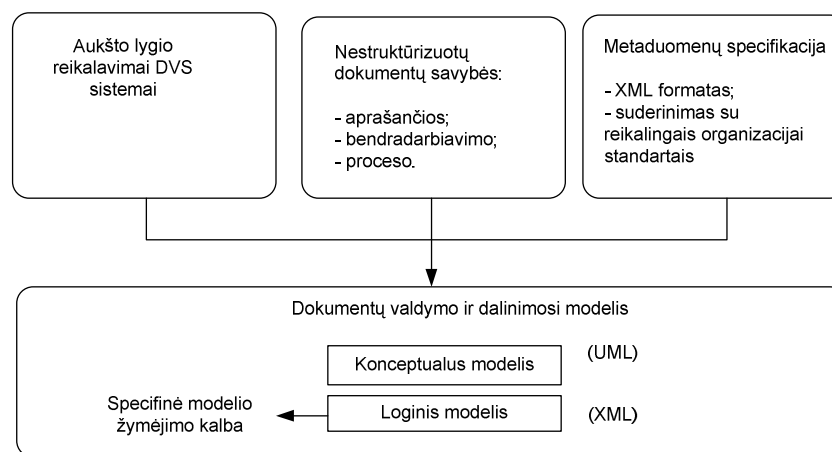
Metaduomenys – tai struktūrizuota informacija, aprašanti, aiškinanti, lokalizuojanti arba kitokiais būdais palengvinanti informacinio resurso išrinkimą, panaudojimą ar valdymą. Dažniausiai metaduomenys vadinami duomenimis apie duomenis [11]. Dokumentų metaduomenų naudojimas yra pagrindinė alternatyva daugeliui nestructūrizuotų dokumentų valdymo sistemų.

Viename darbe [24] siūlomas informacinis dokumentų valdymo ir dalinimosi modelis (angl. *Document Management and Sharing Model*), apibrėžiantis efektyvios metaduomenų formos savybes. Šio modelio branduolio savybių apibrėžimui modelio autorius kaip objektus analizuoja skaitmeninius organizacijos dokumentus, kuriuos reikia identifikuoti ir atlikti jų paiešką, jais dalinasi darbuotojai tų pačių tikslų siekimui, yra charakterizuojami skirtingomis būsenomis (pvz., juodraštis, atiduotas peržiūrai ir kt.) jų gyvavimo cikle. Dėl šių aspektų dokumentų valdymo ir dalinimosi modelis suskaidomas į tris papildomas:

- Aprašantysis informacinis modelis. Pateikia savybių rinkinį, kurios aprašo ir identifikuoja dokumentą, pvz., pavadinimas, kūrėjas, data, apibūdinimas, dokumento tipas, kontaktai. Šie metaduomenys yra bendri, naudojami paieškos ir indeksavimo tikslams.
- Bendradarbiavimo modelis. Formalizuoja žmogiškųjų resursų struktūrą (organizacinė schema) ir prieigos prie informacijos resursų reguliavimą, pagrįstą organizacijos rolėmis ir atsakomybėmis. Šis modelis suteikia prieigos prie dokumentų teisių valdymą, orientuotą į vieną vartotoją bei į rolę ar grupę.
- Proceso modelis. Apibūdina dokumento gyvavimo ciklą, paprastai sudarytą iš šių pagrindinių etapų: sukūrimo, peržiūros, publikavimo, prieigos, archyvavimo ir trynimo. Gyvavimo ciklas nebūtinai turi turėti visus šiuos etapus, – tai procesas, specifikuotas organizacinio modelio viena po kitos sekančių užduočių, atliekamų atitinkamų aktorių, terminais.

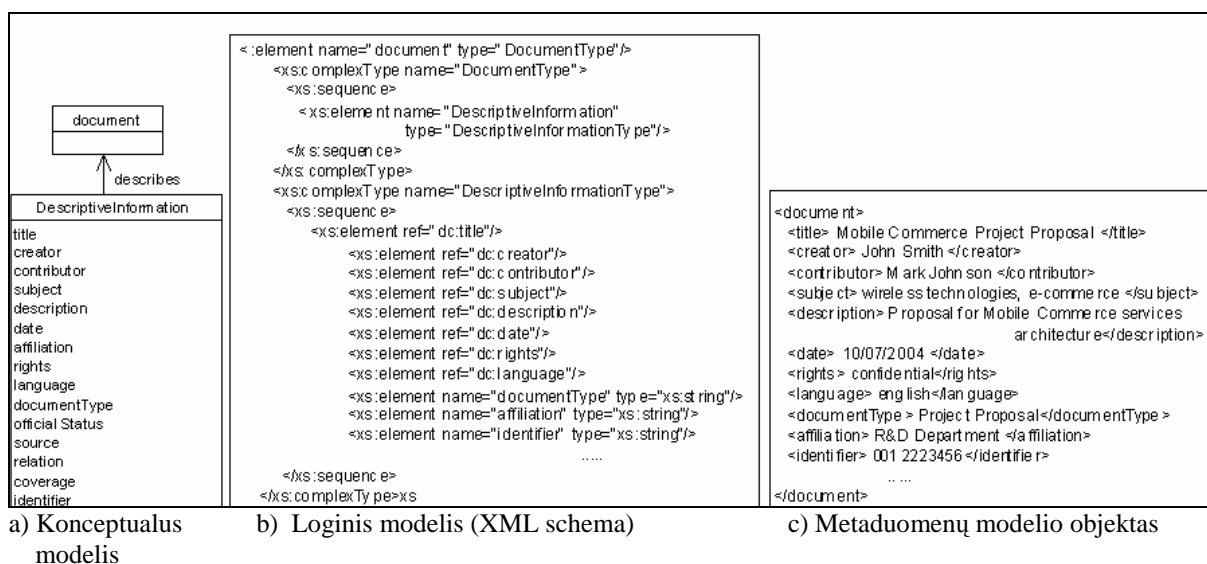
Šiame modelyje metaduomenys specifikuojami dviem abstrakcijų projektavimo lygiais (žr. 5 pav.):

- Konceptualus projektavimas, paremtas UML notacija, pateikianti abstrakčią nuo platformos nepriklausomą konceptų bei sąryšių tarp jų atvaizdavimą.
- Loginis projektavimas. Šis lygis su tam tikra sritimi susietus konceptus verčia į atitinkamą XML struktūrą.



5 pav. Apibendrintas metaduomenų modelis

6 pav. pateiktas dokumentų valdymo ir dalinimosi modelio dalis, vaizduojanti aprašantįjį informacinį modelį (a) ir jo loginį atvaizdavimą XML schemeje. Modelio objektas – dokumentas (c), pavaizduotas XML formatu.



6 pav. Metaduomenų aprašantysis informacinis modelis

Efektyvių metaduomenų išskyrimui dokumentų valdymo ir dalinimosi modelis projektuojamas pirmiausia organizacijoje naudojamų dokumentų klasifikacija: informacija kiekvienai dokumentų klasei, surenkama klausimyno principu, apibrėžia modelio specifikacijas, – leidžia išskirti bendrinius, bendradarbiavimo bei proceso metaduomenis išvengiant jų pertekliško.

2.6.2.1. Onix – knygų metaduomenų standartas

Pasaulinė knygų pramonė duomenų apie knygas perdavimui el. komercijos sistemoms, vykdančioms knygų prekybą, naudoja Onix standartą.

Onix (angl. *Online Information Exchange*), – tarptautinis knygų prekybos metaduomenų standartas, inicijuotas 1999 m. Amerikos leidėjų asociacijos (angl. AAP – *American Association of Publishers*) su pradiniu tikslu: suteikti galimybę leidėjams tiekti „turtingą“

informaciją apie produktą internetinei knygų prekybai. Onix, kurio pirmoji versija pasirodžiusi 2000 m., šiuo metu leidžiamas ir vystomas organizacijos EDItEUR bendradarbiaujančios su BISG (angl. *Book Industry Study Group*) JAV ir BIC (angl. *Book Industry Communication*) Anglijoje. Kitose šalyse (Australijoje, Kanadoje, Vokietijoje ir Korėjoje) įkurtos grupės kurios Onix DTD schemą išplėtė ir pritaikė vietinėms el. knygų prekybos rinkoms.

Onix sudarytas iš turinio specifikacijos ir XML DTD schemas. Turinio specifikaciją sudaro apibrėžtų duomenų elementų rinkinys, kodų sąrašai ir XML žymės, kurios gali būti ir trumpi kodai (pvz., <b012>) arba tekstinės etiketės (pvz., <ProductForm>). Į Onix duomenų elementus šalia išsamios bibliografinės informacijos, įeina turinių lentelės, tekstiniai elementai (pvz., anotacijos, apžvalgos, ištraukos, autorių biografijos ir kt.), paveikslėliai (pvz., knygos viršeliai, autorių nuotraukos), nuorodos į video, audio failus, tinklalapius, informacija apie knygos reklamą.

7 pav. pateikta pavyzdinė Onix įrašo dalis dviem variantais: etiketėse naudojant pilną tekstą ir trumpas žymes.

<pre> <ProductIdentifier> <ProductIDType>02</ProductIDType> <IDValue>0816016356</IDValue> </ProductIdentifier> <ProductForm>BB</ProductForm> <Title> <TitleType>01</TitleType> <TitleText textcase = "02">British English, A to Zed</TitleText> </Title> <Contributor> <SequenceNumber>1</SequenceNumber> <ContributorRole>A01</ContributorRole> <PersonNameInverted>Schur, Norman W</PersonNameInverted> <BiographicalNote>A Harvard graduate in Latin and Italian literature, Norman Schur attended the University of Rome and the Sorbonne before returning to the United States to study law at Harvard and Columbia Law Schools. Now retired from legal practise, Mr. Schur is a fluent speaker and writer of both British and American English</BiographicalNote> </Contributor> </pre>	<pre> <productidentifier> <b221>02</b221> <b244>0816016356</b244> </productidentifier> <b012>BB</b012> <title> <b202>01</b202> <b203 textcase = "02">British English, A to Zed</b203> </title> <contributor> <b035>A01</b035> <b037>Schur, Norman W</b037> <b044>A Harvard graduate in Latin and Italian literature, Norman Schur attended the University of Rome and the Sorbonne before returning to the United States to study law at Harvard and Columbia Law Schools. Now retired from legal practise, Mr. Schur is a fluent speaker and writer of both British and American English </b044> </contributor> </pre>
---	---

7 pav. Onix įrašo dalies variantai, žymėms naudojant pilną tekstą ir trumpus kodus.

Nuo kitų metaduomenų standartų Onix skiriasi savo informacijos gausumu. Onix įrašas turi daugiau nei 200 duomenų elementų, vieni jų privalomi, kiti pasirenkami. Pvz., *DublinCore* turi tik 15 pasirenkamų elementų. Pilnas Onix įrašas įkeltas į tinklalapį paieškai suteikia fizinės knygos naršymo iliuziją [12].

Onix standartas suteikia galimybę sudaryti metaduomenų įrašus pagal du lygius. Pirmojo lygio duomenų elementai atitinka pagrindinius leidėjų reikalavimus, ypač mažesnių. Antrasis apima pirmojo lygio duomenis bei papildomą sudėtingesnę informaciją.

2.6.3. Dokumento modelių pagrįstas modelis

Dokumento modelių pagrįstos DVS, kuriose naujas dokumentas gali būti pusiau automatiškai sugeneruotas iš jau turimų dokumentų [19]. Tiesa, reikia pabrėžti, kad tokių sistemų potencialas atsiskleidžia tik naudojantis dokumentais, kurių struktūra atskirta nuo turinio, pavyzdžiui, SGML (angl. *Standard Generalized Markup Language*) arba vartojant kurį nors kitą griežtą šiuo metu paplitusį jos dialektą, pvz., XML.

Dokumento modelių pagrįstos DVS dažnai skirstomos į [24]:

- Pagrįstas hipertekstiniu dokumentų modelių;
- paieška pagrįstas sistemas.

Hipertekstinių dokumentų modelių pagrįstose sistemose loginiai ryšiai tarp dokumentų ir dokumentų kolekcijų sukuriami naudojant nuorodas. Taip atsiranda vienalytė sistema, kurioje galima naršyti intuityviai. Deja, tokia sistema turi trūkumų: keliaujant nuorodomis, didelės apimties sistemoje labai lengva pasimesti, sunku sukurti apsaugą nuo nekorektiškų ryšių.

Didelėse, nepažįstamose, heterogeninėse dokumentų kolekcijose hipertekstinio modelio paprastai neužtenka. Tokiose sistemose lengva patekti į aklavietes – dokumentus, kuriuose nėra reikšmingų nuorodų. Paprastai tokioms situacijoms išspręsti pasitelkiama aukšto lygio paieška, kuri padeda arba surasti reikalingus dokumentus, arba patekti į dokumentus su reikiamomis nuorodomis. Įvairių paieškos metodų tyrimais užsiima daug komercinių ir akademinų įstaigų. Pačius metodus galima suskirstyti į [24]:

- Sintaksinės paieškos metodus. Tai paprasčiausi iš paieškos metodų. Paieška atliekama įrašant žodį arba frazę. Šie metodai taikomi daugelyje interneto paieškos sistemų.
- Paieškos metodus metaduomenyse. Metaduomenyse paprastai ieškoma binarinių arba uždaro formato dokumentų.
- Semantinės paieškos metodus. Sudėtingiausi iš metodų. Paieškai naudojami sinonimai ir kitos semantinės koncepcijos. Semantinių paieškų rezultatas – dažnai randami dokumentai, neturintys paieškos frazės, bet atitinkantys paieškos lūkesčius pagal turinį.

Plačiausiai paplitęs indeksavimo metodas – indeksuoti kiekvieną žodį. Šiuo atveju suindeksuotos informacijos dydis siekia 180 – 350% pradinio dokumento. Šio metodo

trūkumas yra tai, kad įvedus klaidingą užklausą (klaidingai įvedus bent vieną simbolį) sistema neras vartotojo ieškomos informacijos arba ras ne tai ko ieško. Indeksuojant kiekvieną žodį atsiranda dar viena problema. Skanuojant dokumentą ir po to naudojant optines teksto atpažinimo programas (OCR) gali dokumente įsivelti klaida. Jei ši klaida bus tame žodyje, kurio ieškos vartotojas, paieškos sistema šiuo atveju irgi neras ieškomos informacijos.

Literatūroje daugiausia mokslinių darbų šioje srityje nuveikta kuriant informacijos išrinkimo iš XML dokumento metodus bei algoritmus, pagreitinančius patį išrinkimo procesą.

2.7. Egzistuojančių knygų leidyklų programų sistemų apžvalga

Yra sukurta daugybė kitų dokumentų valdymo sistemų, realizuojančių panašius sprendimus ir metodus, kuriuos galima pritaikyti skirtingo pobūdžio ir dydžio organizacijose. Mano nagrinėjama sritis specifinė – knygų leidyba. Reikia automatizuoti leidykloje atliekamą darbą, – knygos leidybos procesą.

Pasaulinėje IT rinkoje sukurta dokumentų ir galingų turinio valdymo sistemų orientuotų į knygos leidėjus.

Šiuo metu pasaulyje knygų leidyba yra daugiau verslas nei kultūrinė veikla, – aktualu kuo daugiau parduoti išleistų knygų. Tad akivaizdu, kodėl dauguma sukurtų leidybinių IT sprendimų savo funkcionalumu (CRM, ERM, palaikomi duomenų mainų tarp e-komercijos subjektų standartai ir kt.) yra skirti daugiau leidžiamų knygų pardavimams padidinti, nei pačiam knygos rengimo procesui koordinuoti, nors pastarasis yra integruotas į daugelį knygų informacinių sistemų.

AVATAR (D. Britanija) – tai jau kitokio pobūdžio, verslo valdymo sistema knygų bei žurnalų leidėjams ir platintojams. Į šią sistemą įeina tokie moduliai kaip apskaita, sutartys, projekto kainos skaičiavimai, honorarai, knygos rengimo valdymas (tvarkaraščiai, personalo aktyvumo analizė, darbų srauto valdymas, informacija apie autorines teises). Duomenų mainams naudojamas EDI formatas [25].

Klopotek (Vokietija) – leidybinių IT sprendimų lyderio Europoje sistema *PPM (Product Planing and Management)* - tai knygų ir žurnalų leidybos sprendimai, planavimas, pardavimai, ryšių su vartotojais valdymas ir kt.

QSI's Publishing Solutions (JAV) – galingas įrankis įgyvendinantis specifinius stambių tarptautinių leidėjų taip pat ir mažesnių poreikius. Tai valdymo sistema suteikia galimybę

valdyti resursus ir verslo informaciją, sparčiai augančioje leidybinėse organizacijose. QSI's Publishing Solutions savybės ir pagrindiniai moduliai [26]:

- Informacijos apie knygą valdymas
- Sutarčių valdymas
- Onix+ standartas
- Katalogų valdymas
- Knygos planavimo valdymas
- Pardavimų valdymas
- Internetinės svetainės valdymas
- Ataskaitų valdymas
- Klientų sistemų ir integravimo paslaugos.

Papildomos QSI's Publishing Solutions galimybės:

- Žinių valdymas.
- Darbų srauto (angl. *workflow*) integravimas.
- ERP (angl. *Enterprise Resource Planning*) - įmonės resursų planavimo integravimas.
- CRM Integration (QSI point solutions) – ryšių su klientais valdymas.
- SCM (Supply Chain Management) – pristatymų grandinės valdymas.
- ONIX (angl. *ONline Information eXchange*) informacijos apie leidinį perdavimo internetu standarto realizavimas.
- Realizuotas produkto meta duomenų generavimas ir transformavimas į ONIX+/XML/EDI standartus, bei persiuntimas leidybiniais partneriams.

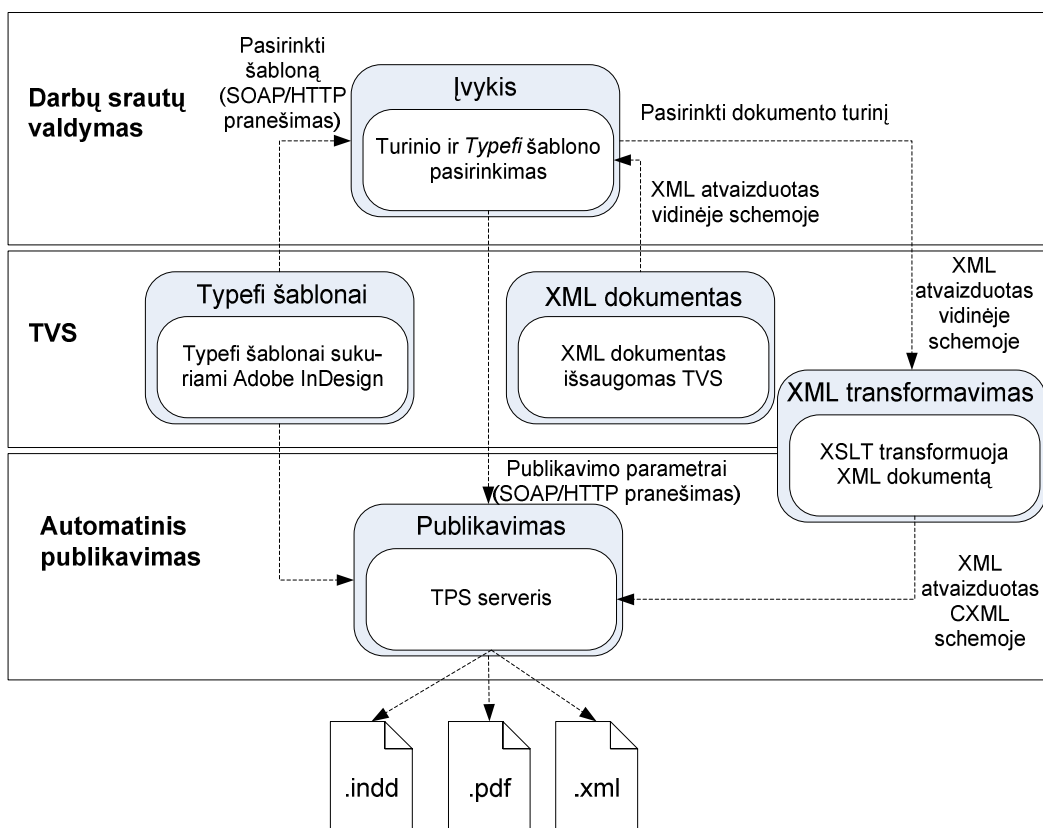
QSI's Publishing Solutions palaiko Macintosh, Windows XP/2000/NT, Windows 95/98/ME, realizuotos web technologijos ir daugiaplatformiškumas (PC/Mac). Naudoja tokias technologijas kaip MS-SQL/Oracle duomenų bazes, kliento/serverio architektūrą, intraneto/interneto portalus (Cognos, XML, or .NET), palaikomos programavimo kalbos – XML, JavaScript, Perl Script, Java, VBScript, HTML, C/C++, COBOL, 4GL.

Knygų leidybos pramonėje siekis didinti pelnus nulėmė pastangas efektyvinti knygos rengimo spaudai procesą nuo rankraščio/vertimo pradžios iki galutinio produkto gavimo. Nors kompiuterizuotų darbų srautų atsiradimas žada naujus pagerintus būdus efektyviam tikslų siekimui, įrankiai skirti leidybos proceso automatizavimui vis dar fragmentuoti. Šias problemas išsprendė *Typefi*.

Typefi Publishing System (JAV, Australija) - leidybinė sistema, pilnai automatizuojanti knygos kūrimo procesą. Realizuotas darbų srautas, turinio valdymas, lankstus knygos

leidybos sprendimas [27]. Darbų srautų valdymas suteikia integruotą aplinką tiek kiekvienam leidybinės komandos nariui, tiek išoriniam resursui ar paslaugos tiekėjui, įtrauktam į leidybos procesą, leidžiančią bendradarbiauti taip pagreitinant patį procesą ir sumažinant kaštus. Turinio valdymo galimybė leidžia valdyti knygos turinį, – tekstą, iliustracijas, nuotraukas, lenteles ir kt., – vientisoje XML standartu grįstoje struktūroje, kuri užtikrina pakartotiną turinio panaudojimą, keleto dokumentų sujungimą į vieną. Turinio redagavimas sinchronizuotas. Versijavimo ir aprobavimo valdymas, integracija su MS Word ir Adobe InDesign, šioje leidybinėje sistemoje neleidžia prarasti skaitmeninio turinio valdymo.

Vienoje techninėje ataskaitoje [27] *Typefi* pateikia minėtos leidybinės sistemos veikimo metodologiją (žr. 8 pav.).



8 pav. Typefi Publishing System integravimas į TVS naudojantis XML

Integravimo metodas ir strategijos:

- Dokumento turinys yra išsaugomas XML formate, naudojant specifinę Typefi schemą vadinamą CXML (Content XML), tam, kad būtų galima schemas konvertuoti į ir iš CXML naudojant XSLT, kuri automatiškai paleidžiama iš *Typefi Publishing System* (toliau TPS) serverio.

- Dizaineris sukuria Typefi dizaino šabloną Adobe InDesign programoje. Šablonas yra automatizuojamas su *Typefi* dizaineriu, įrankis, įterpiamas į Adobe InDesign ir įgalina automatizaciją bei struktūrinių savybių pridėjimą prie Indesign dokumento.
- TPS kompozicijos variklis – galingas įrankis, veikiantis už *Typefi* puslapio išdėstymo sistemos ribų. Variklis veikia kartu su Adobe InDesign, CXML turiniui pritaikydamas projektavimo taisykles ir elementus specifikuotus Typefi dizaino šablone. Komunikacija su TPS kompozicijos varikliu valdoma per TPS serverį.

TPS projektuota atvirais standartais, naudojant SOAP ir XML kaip pagrindines integravimo ir komunikacines technologijas. TPS siūlo daug HTTP iškvietimų, SOAP API skriptų ir atvirą XML schemą, integravimui su kitomis sistemomis palengvinti.

Tai XML turinio, laikomo kitose turinio valdymo sistemose, publikavimas per TPS. Kadangi TPS turi savo turinio valdymo sistemą, *Typefi* dizaino šablonų (InDesign failai) turinys gali būti saugomas tiek jos viduje, tiek išorėje.

Integravimo į TVS per XML algoritmo etapai:

- Vartotojas arba automatizuotas procesas pasirenka turinį publikavimui iš kitos TVS.
- Vartotojas arba automatizuotas procesas pasirenka išvesties šabloną, sukurtą anksčiau grafinio dizainerio. Jei *Typefi* šablonai yra laikomi TPS, juos galima išrinkti HTTP užklausa.
- Turinys yra išrenkamas iš TVS XML formatu naudojant vidinę organizacijos schemą.
- XML transformuojamas į CXML naudojant XSLT skriptą. CXML – viešai prieinama schema.
- Vykdomas HTTP iškvietimas į TPS, į kurį įtraukiamas XML turinys, *Typefi* dizaino šablono pavadinimas ir reikalingi kintamieji (tokie kaip PDF išvesties nustatymai).
- TPS apdoroja XML, formatuodama jį, kad pastarasis atitiktų dizaino šablono reikalavimus, gražintų PDF, log failą ir pasirinktinai InDesign dokumentą, atitinkantį galutinį failą.
- Gražinti failai gali būti išsaugomi TVS ir/arba pateikiami vartotojui.

Svarbu pastebėti, kad kai kuriais atvejais XSLT naudojama turinio į CXML formatą transformavimui yra specifinė pagal šabloną. Kai du šablonai turi tą pačią turinio struktūrą, bet skirtingą formatavimą, dizainą ir vaizdo charakteristikas, ta pati XSLT gali būti naudojamas XML konvertavimui į CXML. Kai du šablonai turi skirtingas turinio struktūras, pvz., vienas dokumentas yra sudarytas iš skyrių, o kitas iš duomenų bazių lentelių, reikės skirtingų XSLT skriptų.

Gerai suprojektuotoje realizacijoje vienintelis XSLT skriptų rinkinys gali būti naudojamas visų tipų konvertavimams, nes pastraipų ir dizaino elementų pavadinimai yra unikalūs šablonų rinkinyje.

Daugelyje atvejų pakeitimai InDesign dokumente gali būti grąžinami atgal į CXML, naudojant dar vieną HTTP iškvietaimą į TPS. CXML transformavimas į pradinį XML priklausys nuo keletu faktorių, tokių kaip konvertavimo suskaidymo lygis ir poreikis išsaugoti metaduomenis.

Kai kurios organizacijos nekaupia XML savo TVS sistemose. Vietoj to, jos išsaugo MS Word ar kitame formate. Dokumentus. TPS gali būti panaudotas tokioje aplinkoje. Šis integravimas yra žymiai paprastesnis nei į XML pagrįstą TVS.

2.8. Knygų leidybos dokumentų valdymas

2.8.1. Knygų leidybos procesas

Projektuojant knygų leidyklos dokumentų valdymo sistemą aktualus dokumento kontekstas, jį supančios aplinkos informacija. Pirmiausia reikėtų apibrėžti knygų leidybos procesą. Knygų leidybos procesą galima išskaidyti į šiuos esminius etapus, kuriuos sudaro atitinkamos veiklos:

- Knygos parinkimas.
- Autorinės sutarties sudarymas. Išsirinkus knygą pasirašoma autorinė sutartis su užsienio leidyklos autorinių teisių agentūra, arba su lietuvių autoriumi.
- Knygos rengimo koordinavimas. Knygos leidybinės grupės sudarymas (vertėjai, redaktoriai, stilistai, korektoriai, maketuotojas, dizaineris).
- Knygos ISBN registravimas. Registruojamas ISBN agentūroje.
- Popieriaus užsakymas. Siunčiamas užsakymas popieriaus tiekėjui.
- Užsakymas spaustuvei. Siunčiamas knygos spausdinimo užsakymas spaustuvei.
- Parengtų knygos dokumentų atidavimas spaudai.

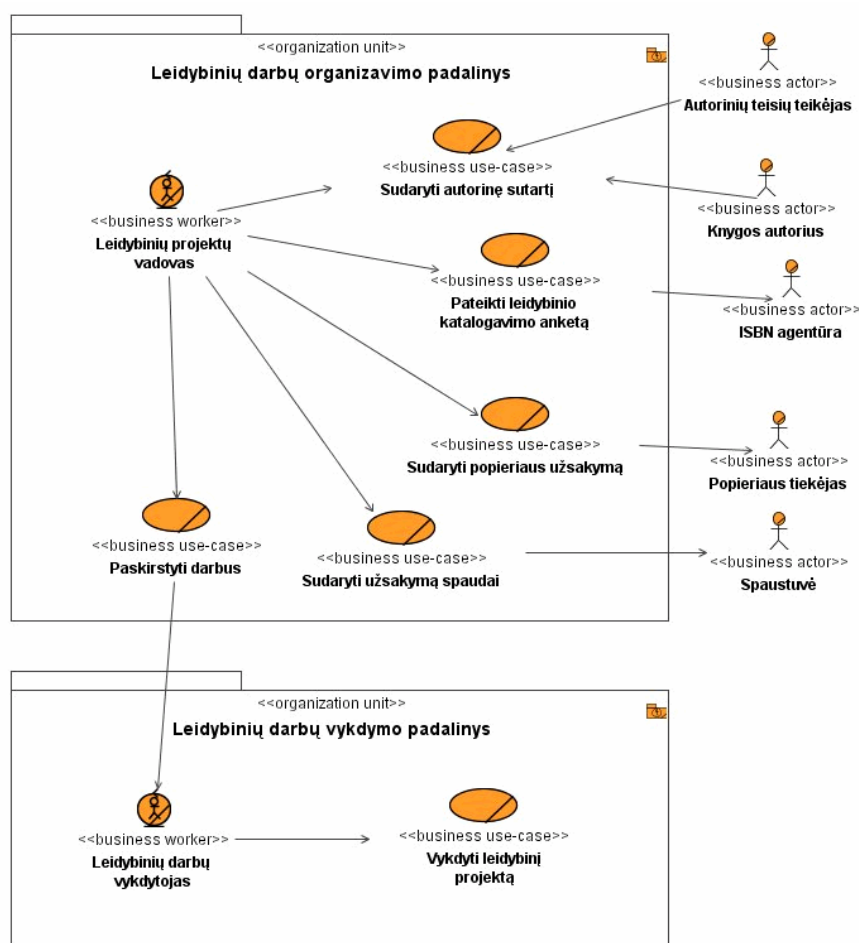
Visų šių veiklų metu rengiami atitinkami dokumentai: knygų leidimo planai, autorinės sutartys, ataskaitos, knygos rengimo failai, ISBN registravimo, užsakymų popieriaus tiekėjams ir spaustuvei dokumentai ir daugelis kitų.

Leidyklos viduje, knygą rengiantys darbuotojai: vertėjai, knygų autoriai, redaktoriai, maketuotojai, dizaineriai ir kt. Už leidybos proceso kokybę atsakingas vyriausias redaktorius.

Kaip matome 9 pav. leidybos procese dalyvauja ne tik pati leidykla, bet ir išoriniai aktoriai, – autorinių teisių teikėjas, spaustuvė, popieriaus tiekėjai, ISBN agentūra, knygos autorius su kuriais keičiamasi informacija – dokumentais.

Pagrindinis kuriamos knygų leidyklos dokumentų valdymo sistemos tikslas – valdyti knygų leidyklos informaciją ir joje vykstančius knygų rengimo spaudai procesus:

- Valdyti leidyklos dokumentus: atlikti greitą dokumentų paiešką, saugoti naujus ir esamus dokumentus, sudaryti vartotojų prieigos prie dokumentų kontrolę.
- Valdyti leidyklos veiklos procesus: sukurti, stebėti ir valdyti užduotis – turėti informaciją apie užduoties gyvavimo ciklą (jos kintančią būseną), kompiuterizuoti užduočių atlikimą – realizuoti atitinkamą darbų seką.



9 pav. Knygų leidyklos organizacinis modelis

2.8.2. Knygų leidykloje saugomos informacijos kategorijos

Elektroniniai dokumentai organizacijoje yra naudojami informacijos valdymui, kaip priemonė grupuoti, organizuoti, saugoti, perkelti ir naudoti informaciją siekiant organizacijos tikslų.

Galima išskirti tris knygų leidykloje kaupiamos informacijos kategorijas, pavaizduotas lentelėje:

3 lentelė. Knygų leidykloje saugomos informacijos kategorijos

	Bendra	Specifinė leidybiniam projektui
Būdinga knygų leidybai	Informacija apie naudojamą popieriaus rūšį, ISBN registravimo tvarką, brūkšninio kodo generavimą ir kt.	
Specifinė knygų leidyklai	Informacija apie knygą, knygos formato, viršelio kūrimo reikalavimai ir kt.	Informacija, kaupiama knygos rengimo metu

Knygų leidyklos vyriausias redaktorius, naudodamas paiešką ir valdymo funkcionalumą, iš savo darbo vietos turi turėti prieigą prie visų išvardintų kategorijų informacijos:

Pirmoji kategorija – informacija, būdinga knygų leidybai – teikiama daugelio aktorių, pvz., spaustuvės, popieriaus tiekėjo, ISBN registravimo agentūros, užsienio leidėjų ir kt.

Antroji kategorija – informacija, būdinga leidyklai, bet ne projektui. Tai anksčiau rengtų knygų leidybinių projektų sprendimų informacija, kurią galima būtų pakartotinai panaudoti naujų knygų rengimui. Informacija yra bendra, tad ji turi būti prieinama visiems leidyklos darbuotojams ir saugoma ilgesnį laikotarpį.

Trečioji kategorija – informacija, sukaupta leidyklos darbuotojams rengiant tam tikrą knygą. Knygą išleidus atsiranda poreikis šią informaciją archyvuoti, kad vėliau būtų galima rasti ir panaudoti kitų knygų leidybos procese.

Šiame darbe pagrindinis dėmesys bus skirtas specifinei knygos leidybinio projekto ir knygų leidyklos informacijai. Šios informacijos kategorijos glaudžiai susijusios.

Knygos ir leidybos proceso Informaciją galima išskaidyti ir kitu aspektu:

- Knygos dokumentai – tekstai originalo kalba, išversti tekstai, iliustracijos, metaduomenys ir kt., susiję su galutiniu produktu – knyga. Šie dokumentai rengiami .doc, .odt, .cdr, .pmd, .indd formatais, naudojant MS Word, OpenOffice, Corel Draw, Page Maker, InDesign programinę įrangą, įdiegtą vykdytojo kompiuteryje, .pdf – galutinis formatas, kuris šiuo metu
- Leidybos proceso dokumentai (administraciniai) – knygų rengimo planai, sutartys, ataskaitos, užsakymai spaustuvėms ir popieriaus tiekėjams ir kt., užtikrinantys knygos rengimo kokybę. Čia vyrauja biuro programomis (MS Office arba OpenOffice) rengiami dokumentai.

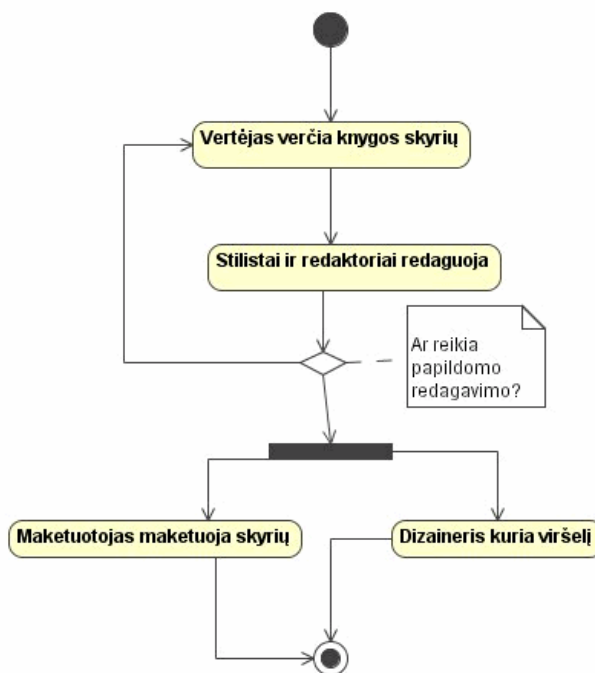
Valdant skaitmeninę informaciją, knygų leidykloje didesnis dėmesys buvo skiriamas knygoms nei jos leidybos proceso dokumentams. Knygos dokumentai yra svarbesni, nes rengiami atitinkamu susitarimu, eiliškumu ir saugomi ilgesnį laiko tarpą. Kad būtų pasiektas gerai struktūrizuoto dokumentų valdymo lygis abu dokumentų tipai turi būti vienodai svarbūs.

2.8.3. Reikalavimai knygų leidyklos dokumentų valdymo sistemai

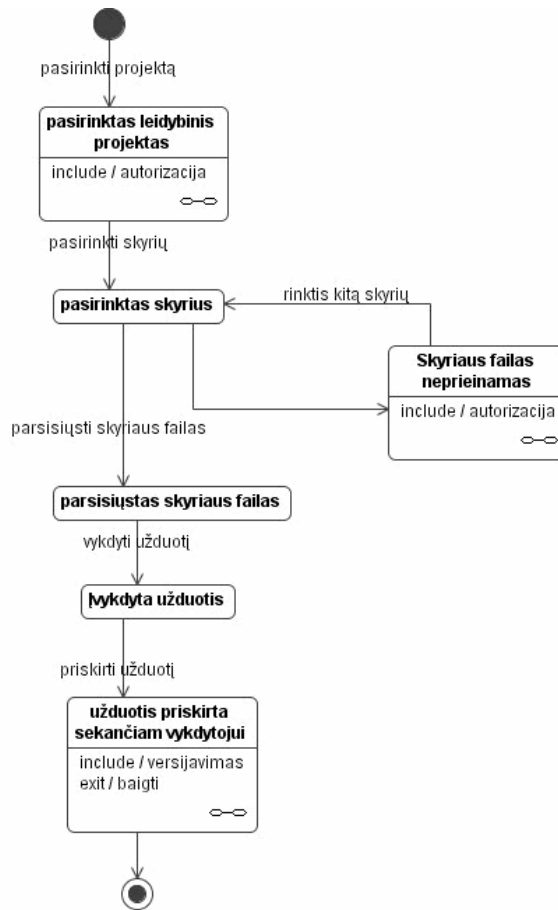
Projektuojant dokumentų valdymo sistemą knygų leidybos procesui, būtina atsižvelgti į leidyklos veiklai keliamus bendrus reikalavimus, kurie yra būdingi grupinio darbo, darbų srautų sistemoms ir specifinius, kurie yra būdingi knygų leidybos procesui: darbų srauto specifiška, prieigos teisių kontrolė, ISBN sudarymas ir registravimas, knygos išskaidymas į užduotis ir kt.

Knygų leidykloje dokumentą rengia kelios žmonių grupės – vertėjai, autoriai, redaktoriai, korektoriai, stilistai, dizaineriai ir maketuotojai. Vartotojas su tuo pačiu vaidmeniu skirtingose grupėse ar rengiamų knygų grupėje gali turėti skirtingas prieigos teises.

Darbų srauto sudarymas – tai dokumento kelio tarp proceso vykdytojų ir atliekamų leidybos veiksmų visuma. Kiekvienas, rengdamas dokumentą, atlieka savo užduotį. Kiekvienai knygų leidyklai būdingas savitas darbų srautas. Sudaryti vientisą darbų srautą praktiškai neįmanoma, nes jis nuolat kinta, nors proceso dalyvių hierarchija grindžiama griežtu darbų paskirstymu, pavyzdžiui, redaktorių suredaguotą knygos skyrių gali siųsti arba vertėjui, arba korektoriui/stilistui, o jei pastarųjų nėra – maketuotojui.



10 pav. Supaprastintas darbų srautas



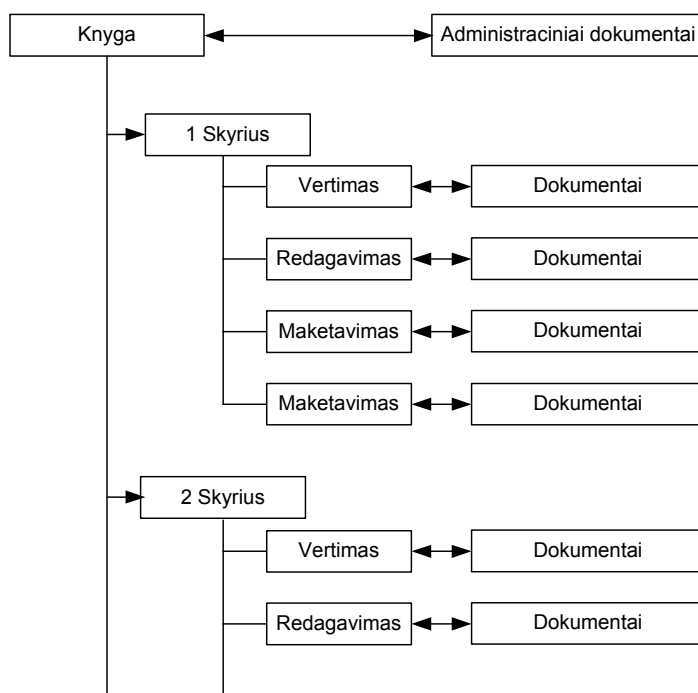
11 pav. Leidybinio projekto būsenų diagrama

Taigi darbe iškeliamas tikslas – sukurti programų sistemą, skirtą įgyvendinti dokumentų valdymo sprendimus knygų leidykloje, – susieti leidžiamos knygos dokumentus, darbuotojus, jų gaunamas/skiriamas užduotis. Projektas apjungia kelias problemines sritis: dokumentų paiešką, dokumentų saugojimą, grupinio darbo, darbų srautų organizavimą, dokumentų archyvų įjungimą į kuriamą sistemą, dokumentų valdymo sistemos integravimą į naudojamą leidyklos informacinę sistemą. Sistema naudosis leidyklos darbuotojas, kuris turės vieną iš sistemos rolių – administratorius, vykdytojas ir stebėtojas. Vykdytojas leidykloje atitinkamai turi savo pareigybių rolę (pvz., redaktoriai, meniniai ir techniniai redaktoriai, dizaineriai, maketuotojai, korektoriai), kurioms suteikiamos atitinkamos teisės. Suformuluotos šios pagrindinės užsakovo pageidaujamos sistemos funkcijos:

- Centralizuotas dokumentų saugojimas, valdymas, paieška.
- Dokumento būsenos valdymas (saugomos metaduomenyse) valdymas (pvz.: jei jis baigtas, jis gali būti prieinamas kitiems vartotojams).
- Metaduomenų indeksacija, leidžianti kontroliuoti dokumentų srautą pagal indekso duomenis (pvz.: pagal autorių, pavadinimą, sukūrimo datą).
- Bendra visos sistemos dokumentų sauga, paremta rolėmis bei grupėmis.

- Dokumentų versijų palaikymas. Versijų istorija ir visų veiksmų atliktų sistemoje auditas (kelių tipų dokumentų išsaugojimas viename įrašė).
- Ataskaitų generavimas.
- Autorinių sutarčių sudarymas.
- ISBN registravimas.
- Dokumento archyvavimas/atstatymas iš archyvo.

Kuriant knygų leidyklos dokumentų valdymo sistemą naudojamas anksčiau aptartas hierarchinis dokumentų modelis, kurio trūkumas, – vieno dokumento priklausymas kelioms failų hierarchijos šakoms (žr. 12 pav.), išsprendžiamas sukuriant nuorodas į tą patį dokumentą iš skirtingų sistemos vietų bei pritaikius dokumentų versijavimą. Dėl knygos rengime naudojamų failų formatų (.doc, .odt, .cdr, .pmd, .indd, .pdf, .ps ir kt.) gausos į dokumentus bus žiūrima kaip į neturinčius struktūros, lankstesnis jų išrinkimas ir paieška realizuojama aprašant dokumentus metaduomenimis.



12 pav. Dokumentų hierarchija knygos rengimo procese

3. PROJEKGINĖ DALIS

Projektinė darbo dalis specifikuoja dokumentų valdymo sistemą, kuri bus orientuota į knygų leidybos sritį. Čia aktualus knygos parengimo spaudai darbų organizavimo procesas, – dokumentų susijusių su ruošiamą išleisti knyga elektroninis gyvavimo ciklas, pradedant nuo knygos vertimo iki jos parengimo spaudai, bei atidavimo archyvui.

3.1. Projekto paskirtis ir apribojimai

Darbo tikslas – suprojektuoti programų sistemą, skirtą įgyvendinti dokumentų valdymo sprendimus knygų leidykloje, – susieti leidžiamos knygos dokumentus, darbuotojus, jų gaunamas/skiriamas užduotis bendroje sistemoje. Projektas apjungia kelias problemines sritis: dokumentų paiešką, dokumentų saugojimą, grupinio darbo, darbų srautų organizavimą, dokumentų archyvų įjungimą į kuriamą sistemą, dokumentų valdymo sistemos integravimą į naudojamą leidyklos informacinę sistemą.

Architektūrinius sprendimus įtakojantys reikalavimai:

- Kontroliuojamas naudojimas sistema (autorizacija).
- Sistemos valdymas ir administravimas per internetinę sąsają.
- Sistema turi būti realizuota kliento-serverio modelyje. Serverinė dalis bus interneto serveryje, o klientinė – bet kokia tinklo naršyklė (pvz.: *Internet Explorer, Mozilla*).
- Sistemos turi būti suprojektuota taip, kad ją galima būtų lengva išplėsti ar prijungti naujus modulius.
- Dokumentų valdymo sistema turi būti įdiegta centralizuotai, – serveryje su Linux operacine sistema. Sistemos vartotojas prie sistemos jungsis per interneto naršyklę. Projekto realizavimui pasirinkta atviro kodo PHP programavimo kalba ir MySQL duomenų bazė.

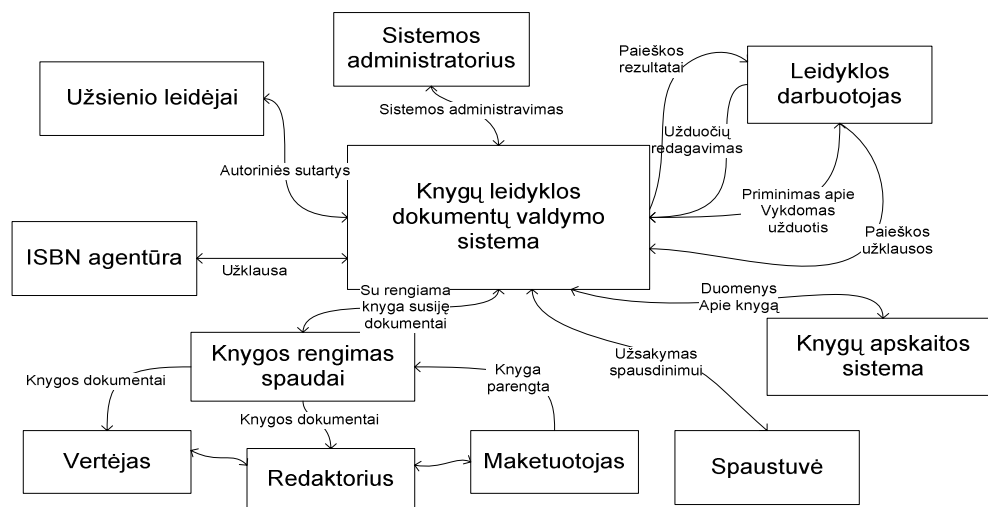
3.2. Veiklos sąveikų modelis

3.2.1. Veiklos sudėtis

Nagrinėjamai veiklos sričiai apibrėžti naudojama konteksto diagrama. Šią sritį tenka ištirti norint sukurti sistemą. Veiklos kontekstas apima daugiau, nei kuriamos sistemos atliekamos funkcijos. Veiklos kontekstas apibrėžia dominančią veiklą ir jos naudojamus bei formuojamus informacijos srautus. Veiklos „atsakomybė“ prasideda tuomet, kai informacijos srautas įeina į

sistemą ir baigiasi, kai rezultatinis srautas išeina iš sistemos. Išorinės esybės diagramoje modeliuoja kaimynines (gretimas) sistemas.

Žemiau esančiame 13 pav. pateikiama veiklos konteksto diagrama.



13 Pav. Veiklos konteksto diagrama

3.2.3. Veiklos padalinimas

Sudaromas veiklos įvykių sąrašas, kuris apima visus veiklos įvykius, už kuriuos yra atsakinga nagrinėjama veikla. Veiklos įvykiai – tai vartotojo išskiriami veiksmai, atliekami veiklos metu. Reakcija (atsakymas) į kiekvieną įvykį atvaizduoja veiklos dalį, įeinančią į bendras veiklą sudarančias funkcijas.

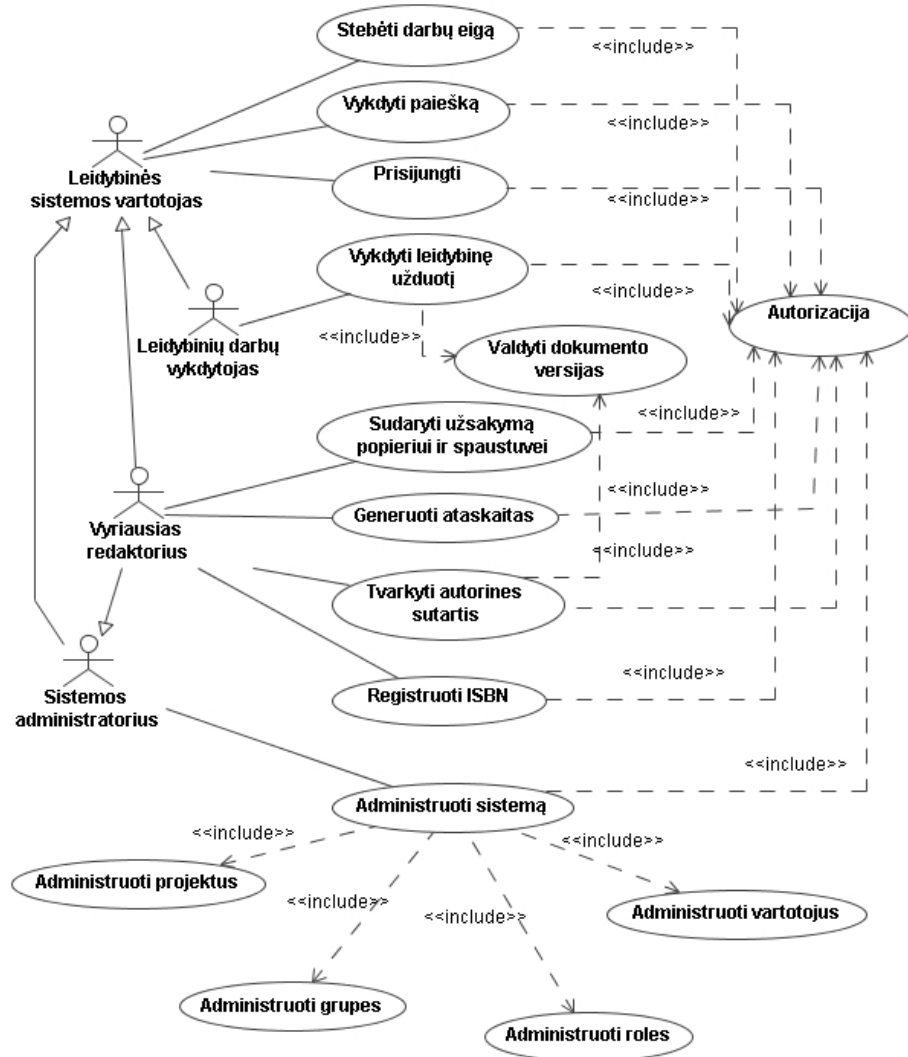
4 Lentelė. Veiklos padalijimas

Nr.	Įvykio pavadinimas	Įeinantys (in) /išeinantys (out) informacijos srautai
1.	Leidykla siunčia užklausą verčiamos knygos užsienio leidėjams, – autorinių teisių turėtojams ir laukia atsakymo	Užklauskos duomenys (in) Gražinamas rezultatas (out)
2.	Administratorius konfigūruoja sistemą	Esami sistemos parametrai (out) Koreguojami /kuriami sistemos parametrai (in)
3.	Leidyklos darbuotojas peržiūri vykdomas užduotis	Vykdomos užduoties būseną (out)
4.	Leidyklos darbuotojas redaguoja vykdomas užduotis	Keičiama užduoties būseną (in)
5.	Leidyklos darbuotojas vykdo paiešką	Užklauskos duomenys (in) Gražinamas rezultatas (out)
6.	Siunčiama užklausa ISBN agentūrai dėl knygos ISBN ir UDK.	Užklauskos duomenys (in) Gražinamas rezultatas (out)

7.	DVS ima duomenis iš knygų apskaitos sistemos	Užklauskos duomenys (in) Gražinamas rezultatas (out)
8.	DVS teikia dokumentų valdymo paslaugą knygų rengimo posistemėi. Šioje posistemėje organizuojamas knygos parengimas: vertėjas, išvertęs dokumentą perduoda jį redaktoriui, pastarasis suredaguoja ir gražina dokumentą peržiūrai vertėjui. Po vertėjo peržiūros atliekamas pakartotinas redagavimas ir išsiuntimas maketavimui. Pagal poreikius atliekamos papildomos korektūros.	Gaunamas dokumentas (in) Išsiųstas dokumentas (out)
9.	Pasibaigus knygos rengimo procesui siunčiama užsakymas spaustuvei	Išsiųstas užsakymas (out) Gautas patvirtinimas (in)

3.3. Sistemos panaudojimo atvejai

Ribas tarp sistemos ir vartotojo nusako panaudojimo atvejų diagrama (žr. 14 pav.).



14 pav. Sistemos panaudojimo atvejų vaizdas

3.3.1. Sistemos aktorių aprašymas

- Leidybinių projektų vykdytojas. Knygų leidykloje esantis darbuotojas (redaktorius, korektorius, stilistas, dizaineris, maketuotojas), knygos autorius arba vertėjas, dirbantis su knygos dokumentais, t.y. vykdančias leidybinę užduotį. Šis aktorius turi prieigą tik prie knygos, su kuria jis dirba, dokumentų, informacijos apie knygas, bei kitų bendrų leidyklos dokumentų.
- Vyriausias redaktorius. Knygų leidykloje dirbantis darbuotojas, atsakingas už knygos rengimo spaudai procesą: darbų eigos koordinavimą, stebint darbų eigą, peržiūrint tarpines dokumentų versijas, autorių sutarčių valdymą, ataskaitų rengimą, ISBN registravimą, užsakymo popieriui ir spaustuvei sudarymą. Taigi, jam suteiktos visos priegijos prie leidyklos dokumentų teisės. Jam suteikta ir sistemos administravimo galimybė.
- Sistemos administratorius. Aktorius atliekantis standartinius sistemos administravimo veiksmus.

3.3.2. Panaudojimo atvejų aprašymas

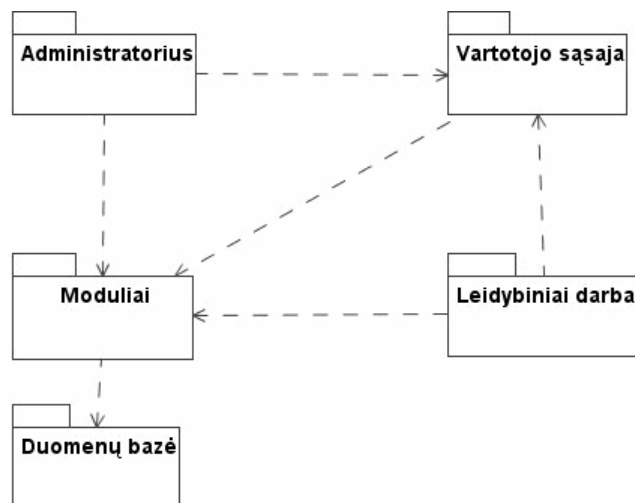
- Prisijungti. Prisijungimo metu autorizuojamas prisijungęs vartotojas tikrinant jo įvestus duomenis (vartotojo vardą ir slaptažodį). Priskirti atitinkamas teises.
- Autorizacija. Tikrinamos vartotojo teisės ir atitinkamai suteikiamos sistemos funkcionalumo valdymo galimybės.
- Stebėti darbų eigą. Galimybė peržiūrėti kiekvienos rengiamos knygos dokumentus, jų tarpines versijas, įrašyti komentarus vykdytojui.
- Vykdyti paiešką. Galimybė ieškoti dokumento pagal įvestus ir pasirinktus parametrus.
- Vykdyti leidybinę užduotį. Atliekami vykdytojui priskiriamos užduotys. Apdorojami su knyga susieti dokumentai.
- Generuoti ataskaitas. Vyriausias redaktorius gali peržiūrėti ir atsispausdinti knygos rengimo trukmės ataskaitas.
- Tvarkyti autorines sutartis. Vyriausiam redaktoriui suteikta galimybė peržiūrėti, įkelti naujas ir atsispausdinti įvairias sutartis, sudarytas su vertėjais, užsienio leidėjais, teikiančiais autorines teises knygai ir kt.
- Registruoti ISBN. Suteikia vartotojui galimybę automatiškai sugeneruoti registracijos formą pasirinktai knygai trūkstantiems laukeliams užpildžius rankiniu būdu ir el. paštu išsiųsti ISBN agentūrai.

- Sudaryti užsakymą popieriui ir spaustuvei. Galimybė pasirinkti užsakymo šabloną, jį užpildyti reikiamais duomenimis ir išsiųsti el. paštu popieriaus tiekėjams ir spaustuvei.
- Administruoti sistemą. Galimybė administruoti leidybinio projekto vykdytojus, jų grupes ir roles, administruoti projektus
- Valdyti dokumentų versijas. Galimybė Išsaugoti redaguojamo dokumento ir modifikuotą ir seną versijas, bei galimybė vėliau jas peržiūrėti.

3.4. Sistemos architektūra

3.4.1. Sistemos komponentai

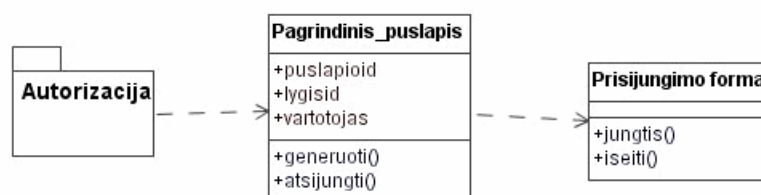
Pateikiamas sistemos loginis vaizdas, kuris išskaidomas į paketus (žr. 15 pav.), kurie detalizuojami aprašant juos sudarančios klases.



15 pav. Sistemos išskaidymas į paketus

3.4.1.1. Paketas vartotojo sąsaja

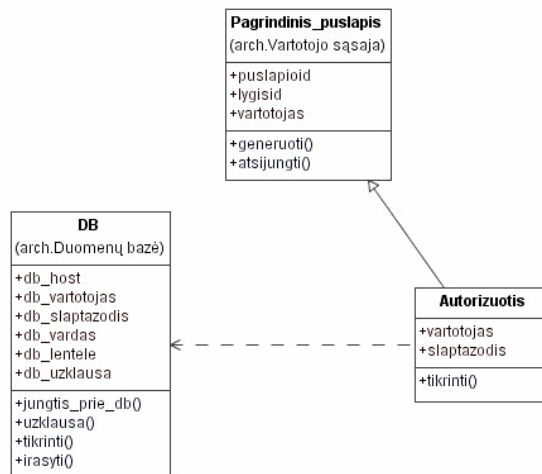
Paketas atlieka sistemos vartotojo sąsajos generavimą. Šis paketas naudoja autorizaciją (žr. 16 pav.).



16 pav. Paketo Vartotojo sąsaja klasių diagrama

Paketas Autorizacija

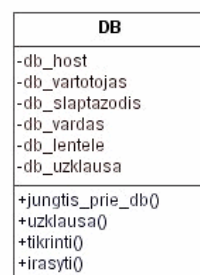
Paketas skirtas prisijungiančių vartotojų verifikavimui bei jų teisių sistemoje nustatymui. Naudoja išorinę paketo „Duomenų bazė“ klasę „DB“, kuri skirta darbui su duomenų bazės įrašais (žr. 17 pav.).



17 pav. Paketo Autorizacija klasių diagrama

3.4.1.2. Paketas Duomenų bazė

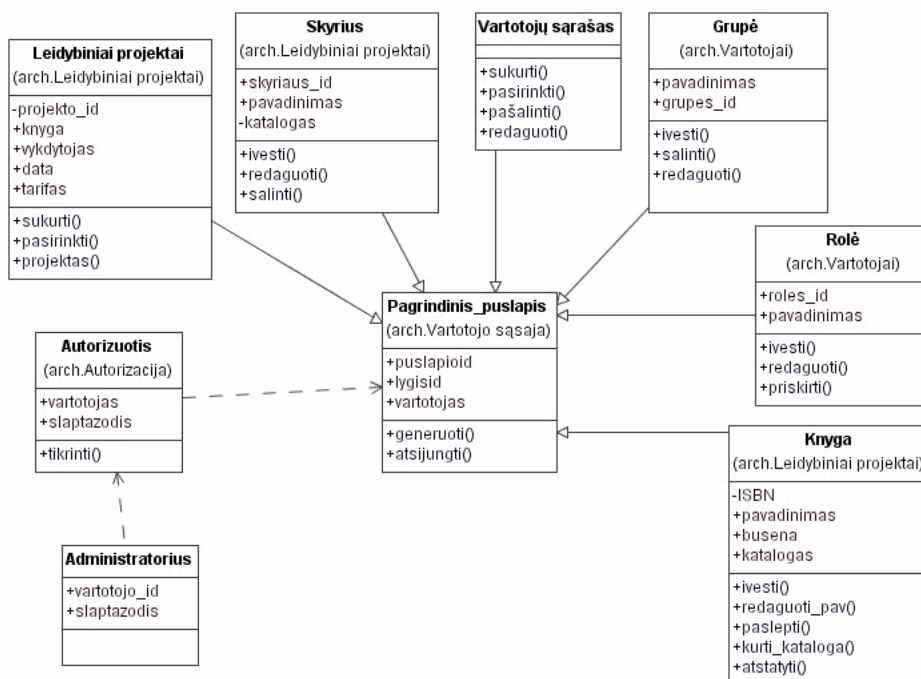
Šis paketas skirtas konfigūracijos bei surenkamai informacijai saugoti bei pateikti ją kitoms klasėms. Tai klasė skirta darbui su duomenų baze. Ji naudojamos daugumoje paketų (žr. 18 pav.).



18 pav. Paketo Duomenų bazė klasių diagrama

3.4.1.3. Paketas Administravimas

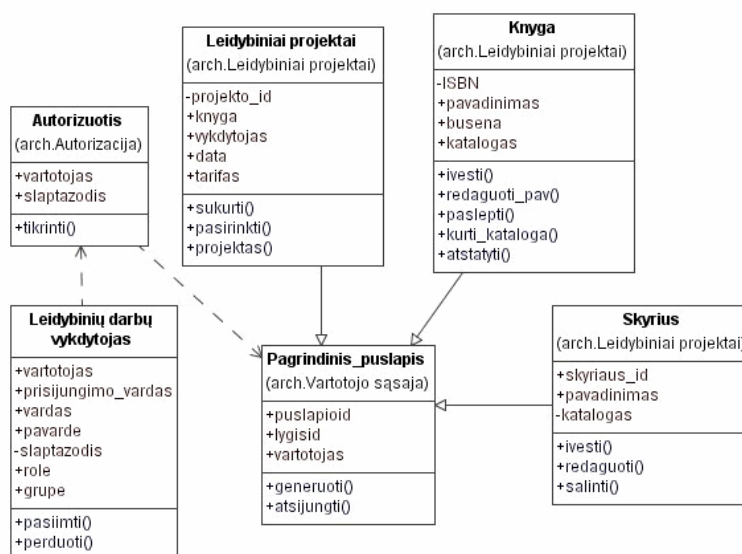
Paketas skirtas vartotojų bei leidybinių projektų administravimui realizuoti. Naudoja išorinę paketo „Duomenų bazė“ klasę „DB“, kuri skirta darbui su duomenų bazės įrašais. Taip pat naudojamas paketas „Autorizacija“, tikrinantis, ar vartotojas turi reikiamas teises (žr. 19 pav.).



19 pav. Paketo Administratorius klasių diagrama

3.4.1.4. Paketas Leidybiniai darbai

Pakete pateikiamos klasės skirtos leidybinių darbų vykdytojo sąsajos realizacijai (žr. 20 pav).



20 pav. Paketo Leidybiniai darbai klasių diagrama

3.4.1.5. Paketas Moduliai

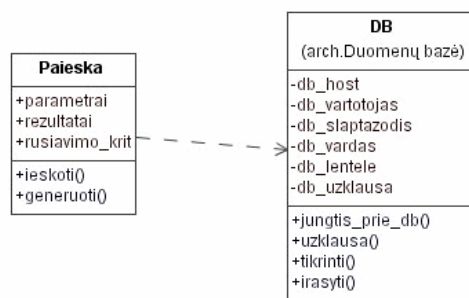
Pakete „Moduliai“ saugomi klasės realizuojančios tokias sistemos teikiama funkcijas kaip paieška, ataskaitų generavimas Leidybinio darbo organizavimas ir vykdymas ir t.t. Sistema nesunkiai galima praplėsti naujais moduliais (žr. 21 pav).



21 pav. Paketas Moduliai.

Paketas Paieška

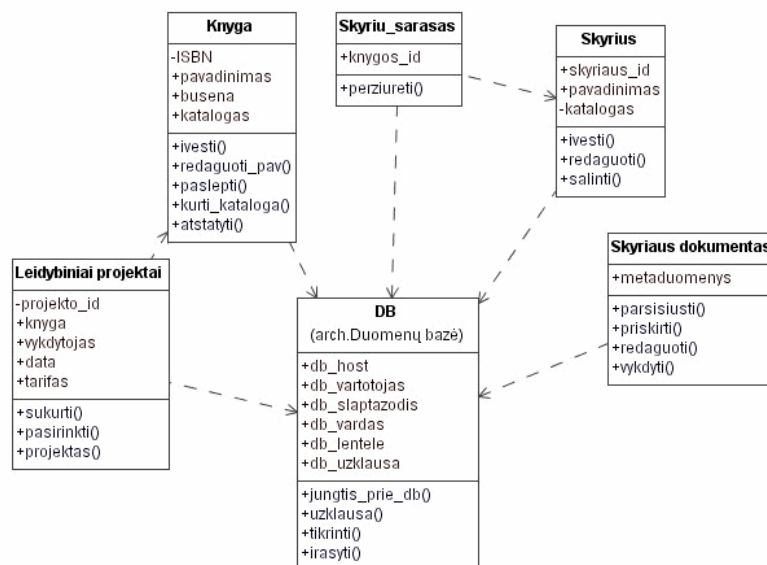
Pakete pateikiamos klasės skirtos paieškos realizacijai (žr. 22 pav.).



22 pav. Paketo Paieška klasių diagrama

Paketas Leidybiniai projektai

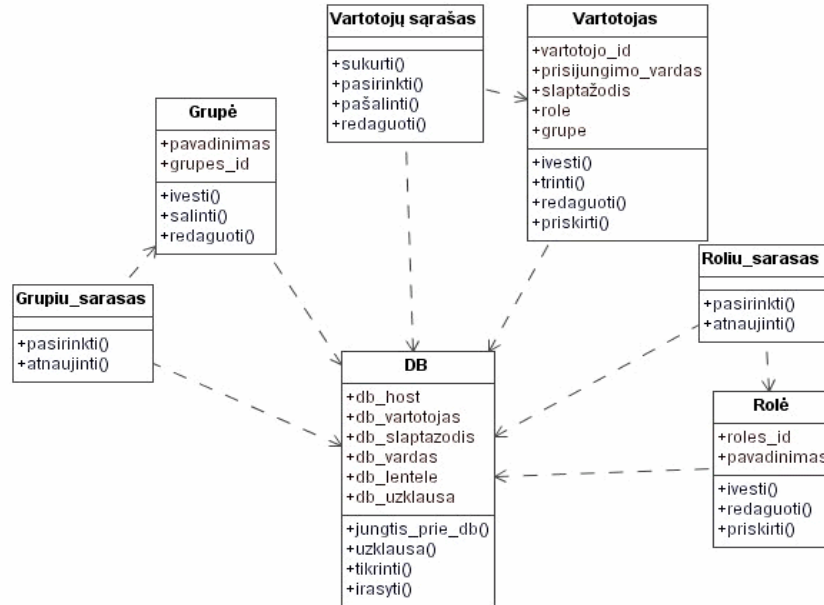
Paketas skirtas leidybinių projektų sąsajai formuoti. Naudoja išorinę paketo „Duomenų bazė“ klasę „DB“, kuri skirta darbui su duomenų bazės įrašais (žr. 23 pav.).



23 pav. Paketo Leidybiniai projektai klasių diagrama

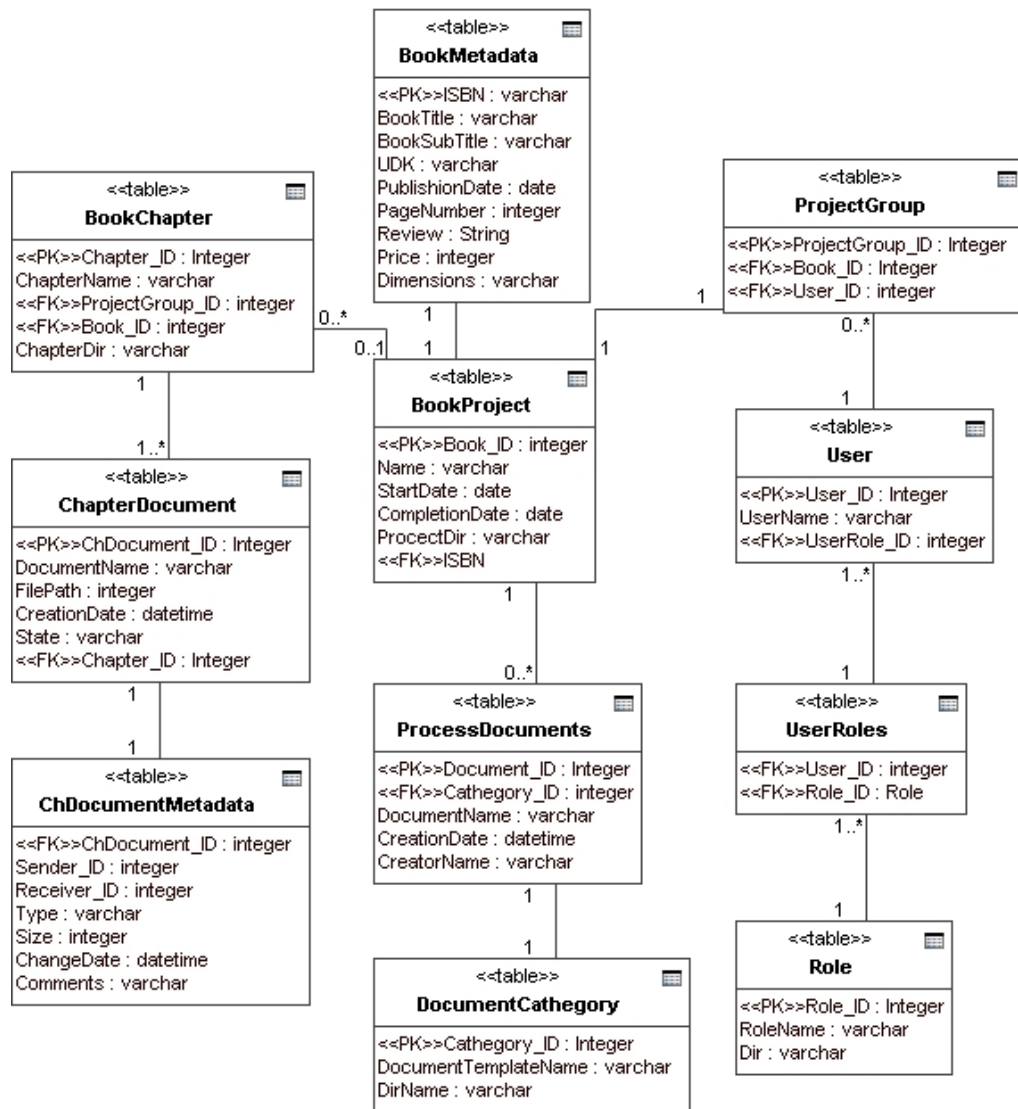
Paketas Vartotojai

Pakete pateikiamos klasės leidžiančios administruoti priemonių sistemos vartotojus, jų grupes ir roles. Paketo klasės teikia funkcijas skirtas sistemos administratoriui.



24 pav. Paketo Vartotojai klasių diagrama

3.4.2. Duomenų vaizdas



25 pav. Duomenų vaizdo schema

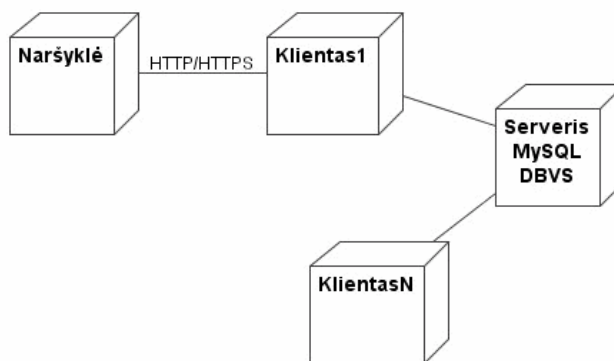
5 lentelė. Esybių aprašymas

Esybė	Aprašymas
BookProject	Saugoma informacija apie knygų leidykloje vykdomus leidybinius projektus. Kiekvienam jų priskiriama atitinkama knyga, su ja susieti knygos rengimo ir leidybinio proceso dokumentai, saugomi atskirose lentelėse.
BookMetadata	Saugoma informacija apie leidyklos knygas.
BookChapter	Saugoma informacija apie knygos skyrius, į kuriuos išskaidomas leidybinis projektas, kad būtų patogiau rengti knygą.
ChapterDocument	Saugoma informacija apie dokumentą, priskirtą knygos skyriui.
ChDocumentMetadata	Saugomi skyriaus dokumento metaduomenys – papildoma informacija.
ProcessDocuments	Saugoma informacija apie administracinius dokumentus,

	reikalingus knygos parengimo procesui, pvz., ISBN registravimo anketos, autorinės sutartys.
DocumentCategory	Saugoma informacija apie administracinių dokumentų kategorijas, į kurias pastarieji surūšiuojami, kad būtų galima juos lengviau rasti.
ProjectGroup	Leidybinio projekto grupė, – vykdytojai, priskirti atitinkamam leidybiniam projektui. Vienas vykdytojas gali priklausyti kelioms grupėms.
User	Saugoma informacija apie leidyklos darbuotojus.
UserRoles	Saugoma informacija apie vartotojui priskirtas roles, dažniausiai pagal jo pareigybę.
Role	Saugoma informacija apie vartotojo roles.

3.4.3. Išdėstymo vaizdas

Naudojamas centralizuotas modelis, kai visa programų sistema vykdoma viename kompiuteryje. Naudojamos atviro kodo technologijos Apache web serverio programinė įranga, PHP interpretatorius ir MySQL duomenų bazių serveris (žr. 26 pav.).



26 pav. Išdėstymo vaizdas

Naršyklė. Interneto naršyklės pagalba galima naudotis sistemos funkcijomis. Interneto naršyklė gali būti bet kokia HTTP ir/arba HTTPS protokolą palaikanti programa. Reikalavimai vartotojo sistemai priklauso nuo naudojamos naršyklės.

Klientas. Tai sistemos įrenginys, kuris gali būti bet kur, kol turi kontaktą su centriniu serveriu duomenų perdavimo tinklu. Tai paprastas kompiuteris, kuriame turi būti suinstaliuota Interneto naršyklė.

Serveris. Esminis sistemos įrenginys, kuris valdys visą sistemos darbą. Serveris kartus bus ir matavimų valdymo įrenginys, ir duomenų bazės įrenginys

4. TYRIMO DALIS

4.1. Sukurtos sistemos kokybės tyrimas

4.1.1. Testavimas

Programinės įrangos kokybės valdymas taip pat tiesiogiai susijęs su testavimu, Programos testavimas buvo vykdomas keletą etapų.

Sistema kuriama atskirais moduliais, kurie kartu apjungiami ir integruojami į bendrą sistemą. Atlikus vienetų testavimą, vienetai yra apjungiami į bendrą sistemą ir testuojama jų sąveika. Bus naudojamas stambinantis „bottom-up“ integracinis testavimas, kai pirmiausiai bus ištestuoti žemiausi duomenų lygio komponentai, vėliau sisteminiai moduliai ir galiausiai prieinant iki vartotojo sąsajos komponentų. Komponentų testavimui bus naudojami testų draiveriai.

Testuojant programų sistemą buvo naudotas rankinis testavimas t.y. įvedant pradines reikšmes ir lyginant gaunamus rezultatus su apskaičiuotais rezultatais. Taip pat numatytas statistinis testavimo metodas, kuris pagrįstas atsitiktinių testinių reikšmių įvedimu, generavimu bei rezultatų skaičiavimu, palyginant rezultatus ir įvertinant jų teisingumą. Šis testas įvertina sistemos patikimumo lygį, kuo daugiau rezultatų sutampa ir klaidų skaičius mažesnis, tuo sistema veikia patikimiau.

Kai kuriems komponentams atliktas automatizuotas loginis testavimas. Kiekvienas komponentas privalo veikti pagal numatytą teisingumą. Tam buvo išbandomi visi galimi komponento įėjimų variantai ir išėjimų kombinacijos bei lyginamos su laukiamu rezultatu. Tik įsitikinus komponento veikimo teisingumu, jis buvo prijungiamas prie bendros programos sistemos. Visi komponentai pratestuoti ir atitiko atitinkamus reikalavimus.

Atskirų vienetų (klasių ar modulių) testavimui bus naudojami struktūrinis ir funkcinis testavimo metodai.

Struktūrinio testavimo metu yra atsižvelgiama į algoritmo struktūrą („balta dėžė“) ir tikrinamas metodo veikimas kai kiekvienas algoritmo sakiny yra įvykdomas bent vieną kartą.

Funkcinio testavimo metu tikrinama ar pagal metodui perduotus pradinius duomenys gaunami laukiami rezultatai. Duomenys parenkami neatsižvelgiant į metodo struktūrą („juoda dėžė“). Testiniai atvejai sudaromi pagal sistemos specifikaciją.

4.1.2. Peržiūros

Sukurtos knygų leidyklos dokumentų valdymo sistemos kokybės tyrimas buvo atliekamas ir kokybės peržiūros metodu. Pasitelkus užsakovo atsakingus asmenis, buvo atliktas interviu, reikalavimų specifikacijoje suformuluotus reikalavimus lyginant su realizuotais.

4.1.2.1. Kokybės kriterijai

Į PĮ kokybės valdymą įeina ne tik teisingų rezultatų gavimas su tam tikrais pradiniais duomenimis, bet ir kiti gana svarbūs aspektai, pvz., programos struktūros aiškumas, teksto skaitomumas, veikimo efektyvumas, vartotojo sąsajos patogumas, ypatingų situacijų apdorojimas, pakartotinis kodo panaudojamumas.

Kokybės nagrinėjimas buvo atliktas kodo analizės ir testavimų pagalba, kurių metu buvo tikrinamos šios esminės metrikos (pagal ISO-9126 standartą):

- Našumas – kiek dokumentų galima peržiūrėti vienu metu ir koks dokumentų srautas gali būti, kad sistema dirbtų našiai.
- Saugumas – kiek lengvai galima pasinaudoti sistemos kode esančiomis spragomis ir gauti nelegalų priejimą prie sistemos.
- Klaidų apdorojimas – kaip sistema reaguoja į neteisingus rezultatus ir koku lygiu apie tas klaidas pranešama.

Be šių įvertinimo kriterijų viso projekto metu buvo naudojami pasyvūs kokybės užtikrinimo metodai, kaip:

- Sudėtingų sistemos kodo vietų įmanomas supaprastinimas, pašalinant tipines žmonių daromas klaidas (pvz. vartotojo įvedamos informacijos nekorektiškumas).
- Probleminių būsenų vengimas – klaidos atveju sistema stengsis veikti toliau, tačiau kartu bus įrašomas pranešimas apie klaidą į klaidų informacinį žurnalą ir/arba į vartotojo sąsają.

4.1.2.2. Kokybės vertinimo rezultatai

ISO-9126 kokybės charakteristikų vertinimas:

6 lentelė. Knygų leidyklos dokumentų valdymo sistemos kokybės vertinimo parametrai

Kriterijus	Aprašymas
Funkcionalumas	Funkcionalumas atitinka užsakovo numatytus panaudojimo atvejus, bei jų realizavimas padarytas netgi lankstesnis negu to buvo reikalaujama (patogesnis naudojimas). Dėka paprastos ir efektyvios architektūros, supaprastintos bei kuo mažiau sukonkretintos duomenų bazės struktūros, sistema gali būti papildyta papildomais panaudojimo atvejais be sudėtingų sprendimų priėmimo ar projektavimų.
Patikimumas	Sistemos rezultatų labai priklauso nuo užklausų duomenų sudarymo, t.y.

	užklauso tipo parinkimo, užklauso dažnio ir atsakymo laukimo parametrų atitikimo realioms tinklo pajėgumams. Visų galimų klaidingų situacijų rezultatai yra saugomi sistemos sisteminiuose žurnaluose, kas leidžia esant klaidai ją pašalinti taip dar padidinant jos patikimumą.
Efektyvumas	Realizuotas efektyvus dokumentų valdymas, jų atrinkimui sistemoje panaudojant užsakovo pageidaujamus kriterijus. Dokumentų paieškos funkcijos efektyvinimui panaudotas metaduomenų rinkinys, aprašantis tiek bendruosius tiek dokumento gyvavimo ciklo būsenų duomenis. Darbų srautų ir grupinio darbo palaikymas sistemoje leidžia efektyviai koordinuoti ir pagreitinti knygos dokumentų rengimo procesą.
Palaikomumas.	Palaikomumas. Sistemos architektūra leidžia lengvai įdiegti naujus panaudojimo atvejus. Naudojant PHP programavimo kalbą, labai lengva daryti pakeitimus, nes nereikia sistemos perkompiliuoti iš naujo. Taikomi
Perkeliamumas.	Sistemai realizuoti buvo naudojamos PHP kalba. Įdiegus ji gali veikti visose operacinėse sistemose, kuriose yra įdiegtas Interneto serveris ir PHP interpretatorius. Kadangi PHP yra viena populiariausių Web kūrimo programavimo kalbų, jos varikliai yra daugelyje žinomiausių WWW servisų. Naudojama ADOdb duomenų bazės abstrakcijos biblioteka, užtikrina įvairių duomenų bazių palaikymą.

7 lentelė. Knygų leidyklos dokumentų valdymo sistemos atitikimas kokybės vertinimo kriterijams

	1	2	3	4	5
Funkcionalumas				✓	
Patikimumas			✓		
Efektyvumas				✓	
Palaikomumas.					✓
Perkeliamumas.					✓
Našumas			✓		
Saugumas				✓	
Klaidų apdorojimas			✓		

4.2. Tolesnės vystymo galimybės

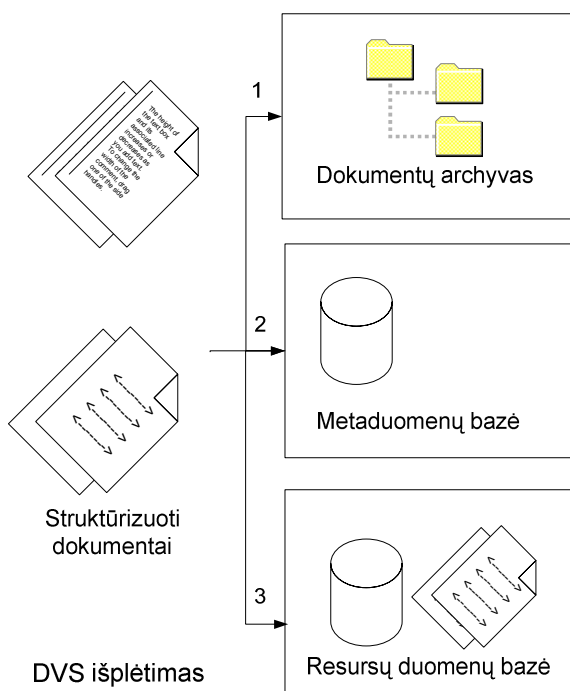
Knygų leidyklos dokumentų valdymo sistemos išplėtimas gali būti padarytas skirtingais aspektais. Dauguma jų priartintų DVS prie analizės dalyje nagrinėtų sistemų, tokių kaip žinių ir turinio valdymo, grupinio darbo sistemų ir kt.

Kitas DVS išplėtimo būdas – automatiškai apdoroti duomenis ir jų struktūrą, esančią dokumento viduje. Papildomos informacijos apie dokumentų turinį trūkumas kliudo dokumento turinio apdorojimui. Ši padėtis keistūsi dėl XML dokumentų naudojimo. Žinoma, vieno XML naudojimas nepagerintų apdorojimo palyginti su kitais nuosavybiniais failų

formatais. Bet XML naudojimas kartu su DTD bei XML schema, apibrėžiančiomis dokumento struktūrą ir tipą. Nuo to laiko kai *Microsoft* perėjo prie XML pagrįsto formato, atsirado galimybė apdoroti ir biuro programų dokumentus. Papildomas *Office* produktas *InfoPath*, – XML failų redaktorius, gali kurti dokumentus pagrįstus DTD arba XML schemomis ir sukuria pagrindą tokio turinio redagavimui. Aišku yra ir kitų produktų, pvz., XML Spy, kurie yra tobulesni, bet vis tik pokyčiai *Microsoft Office* produktuose turi didžiausią įtaką dokumentų valdymui.

XML pagrįstas tekstinis redaktorius leidžia ne tik manipuluoti puslapio išdėstymo, lygiavimo, šriftų ir kt. informacija, iš esmės jis generuoja formą, kurią tik reikia užpildyti (šablonas). Susiję dokumento tušti laukeliai kategorizuojami, kad vėliau juos būtų galima išskirti. Po to kai dokumentas yra išsaugomas dokumentų valdymo sistemoje, galimas tolesnis jo apdorojimas.

Naudojant XML duomenų bazę galima būtų realizuoti efektyvų XML apdorojimą (žr. 27 pav.). Paprasti dokumentai gali būti saugomi archyve (1), o jų metaduomenys saugomi duomenų bazėje (2), kaip bet kokioje hierarchiniu modeliu pagrįstoje DVS. Papildomai gali būti saugomi struktūriniai dokumentai XML pagrįstoje duomenų bazėje (3).

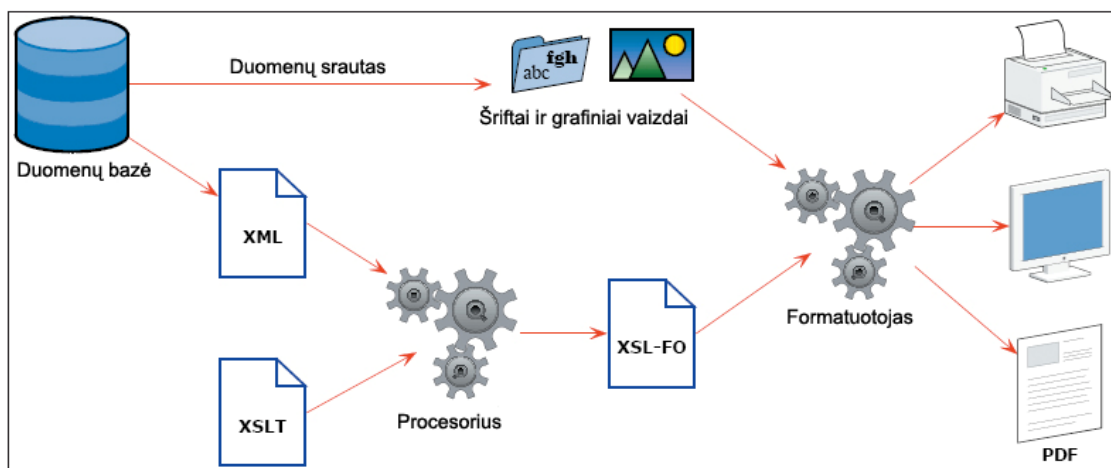


27 pav. DVS išplėtimas

Leidybinio proceso galutinio etapo suformuotas galutinis failas pateikiamas PDF formate. Naudojant XML ir su juo susijusias technologijas galima būtų valdyti struktūrizuotus dokumentus. Kaip jau buvo nagrinėta analitinėje dalyje, XSL standartizuotam darbų srautui iš XML į PDF apibrėžia taisykles. Konceptinis ir techninis transformavimo ir formatavimo

atskyrimas atitinkamai sustiprina išskaidytos į modulius produkcijos srautą. Atsakomybių išskaidymas suteikia galimybę naujam, efektyviam darbų srautui.

Sistemos tolesniam tobulinimui taip pat galima būtų naudoti XSL formatavimo objektus (XSL-FO) – galingą žodyną, generuojantį aukštos kokybės spaudą kaip fiksuoto dydžio puslapių kolekciją. XML-FO nebūtina kurti rankiniu būdu, – galima pasinaudoti XSLT galimybe XML objekto transformavimui į XSL-FO objekto žodyną. XSL-FO formatavimo įrankis interpretuoja XSL-FO objektą ir generuoja puslapių vaizdus. (žr. 28 pav.)



28 pav. Duomenų srautas XSL-FO.

Pradiniai duomenys yra saugomi duomenų bazėje, vėliau jie transformuojami į XML formatą. XSL stilių lentelė XML objektą transformuoja į XSL-FO žodyno objektą. Šriftai ir grafiniai vaizdai įtraukiami į formatavimo įrankį kartu su XSL-FO dokumentu ir sugeneruojama išvestis į printerį, monitoriaus ekraną ar PDF.

Taigi XSL-FO – tai puslapiavimo žymėjimo kalba. Tai kitas žodynas, kuris gali būti išreikštas kaip XML objektas, kuriam reikia PĮ formatuojamam turiniui interpretuoti, kad būtų pasiektas tikslas. Tai kaip HTML interneto naršyklei.

Aišku jis nepakeis maketavimo programų tokių kaip Adobe InDesign ar PageMaker, bet jis būtų naudingas generuojant paprastesnius dokumentus kaip sutartys, finansiniai planai ir kt.

Tolesniam tyrimui galima būtų rinktis taip pat automatizuotą turinio valdymą. Interaktyviam automatizuotam turinio išdėstymui įėjimo duomenys būtų tekstiniai failai, šriftai ir vartotojo nustatymai. Tekstas automatiškai formatuojamas pagal vartotojo nustatymus. Galima būtų realizuoti sinchronišką dokumento turinio išdėstymo keitimą ir redaguojamą į dokumentų valdymą integravus efektyvų versijų valdymą. Standartines įvedimo formas gali būti sukurti MS Word programa, kurios vėliau gali būti transformuojamos į XML ir naudojamos dokumentų valdymo sistemoje.

5. Išvados

1. Dokumentų valdymo problemos plačiai nagrinėjamos visame pasaulyje, su jomis susiduriama daugelyje veiklos sričių, taip pat ir knygų leidyboje. Atlikus dokumentų valdymo sistemų kūrime naudojamų metodologijų ir technologijų analizę, apžvelgus užsienyje egzistuojančių komercinių produktų sprendimus, užsakovui buvo pasiūlytas alternatyvus, atviro kodo technologijomis pagrįstas sprendimas knygų leidyklos dokumentų valdymui.
2. Analizuojant tiek komercinius, tiek akademinis straipsnius, išskirti du pagrindiniai dokumentų valdymo modelių tipai, kurių esminis skirtumas – dokumentų struktūrizavimo lygis. Hierarchinio dokumentų valdymo modelyje su dokumentu elgiama kaip su neturinčiu struktūros. Dokumento modeliu pagrįstame dokumentų valdyme dokumentas struktūrizuojamas, kad būtų galima manipuluoti jo dalimis.
3. Sukūrus bei įdiegus knygų dokumentų valdymo sistemą užsakovo įmonėje buvo atliktas sistemos kokybės tyrimas. Realizuota sistema paremta hierarchiniu modeliu, kuriame dokumentai nėra struktūrizuojami. Tačiau nepaisant to, tai atitiko pradinius užsakovo lūkesčius – vienoje dokumentų valdymo sistemoje integruoti leidyklos dokumentus, leidyklos darbuotojų prieigos teises prie jų, darbų srautų valdymą, dokumentų paiešką ir versijų valdymą.
4. Ateityje bus galima toliau tęsti analitinėje dalyje pradėtą XML technologijų analizę, peržiūrėti atviro kodo XML formatu pagrįstas duomenų bazes ir paeksperimentuoti su struktūriniais XML dokumentais. Įprastos reliacinės duomenų bazės sunkiai susidoroja su mažai struktūrizuotais XML dokumentais, o specialios XML užklausų kalbos smarkiai lenkia SQL, dirbant su hierarchiniais duomenimis. Taip pat galima gilintis į XSL formatavimo objektų (XSL-FO) technologiją, iš XML dokumentų generuojančią PDF formatą, kuris ir yra knygos rengimo skaitmeninei spaudai reikalaujamas galutinis rezultatas.

6. LITERATŪRA

- [1] ISO/TC 10 N 688. [interaktyvus] 1997 [žiūrėta 2006-03-20]. Prieiga per internetą: <http://isotc.iso.org/livelink/livelink/fetch/2000/2122/6112/16156/18708/18674/597360/TC_10_N_688__Chinese_comments_on_different_terms.pdf?nodeid=23572&vernum=0>.
- [2] Sprague R. H. Jr. Electronic Document Management: Challenges and Opportunities for Information Systems Managers [interaktyvus]. [žiūrėta 2006-03-20]. Prieiga per internetą: <<http://www.cba.hawaii.edu/sprague/MISQ/MISQfina.htm>>.
- [3] Bartsch, Gerhard U. Introduction to Electronic Document Management [interaktyvus]. 2005 [žiūrėta 2006-03-20]. Prieiga per internetą: <http://www.macrodome.com/macrodome.nsf/2ce5a108780e440e85256ceb0081883b/<FILE/EDMSPrimer.pdf>>.
- [4] Cleveland G. Overview of Document Management Technology [interaktyvus]. 1995. [žiūrėta 2006-03-20]. Prieiga per internetą: <<http://www.ifla.org/VI/5/op/udtop2/udtop2.htm>>
- [5] Schmitz M.J.W.M. How to manage project knowledge. A ranked, semi-structured, role-orientated approach [interaktyvus]. 2001. [žiūrėta 2006-03-20]. Prieiga per internetą: <<http://www.cs.unimaas.nl/~wiesman/msctheses/schmitz.thesis.pdf>>.
- [6] Dourish P., Edwards W. K., LaMarca A., Lamping J., Petersen K., Salisbury M., Terry D. B. and Thornton J. Extending Document Management Systems with User-Specific Active Properties. ACM Transactions on Information Systems, 2000, 18/2, 140–170.
- [7] Easwaran G. Nadhan. Service-Oriented Architecture: Implementation Challenges. 2004. [Žiūrėta 2005 11 10] Prieiga per internetą: <<http://msdn.microsoft.com/architecture/soa/default.aspx?pull=/library/en-us/dnmaj/html/aj2soaimpc.asp>>.
- [8] Gudas S., Sabaliauskaitė G. Organizacijų veiklos modeliavimas valdomų procesų metodu. [interaktyvus] 1999. [Žiūrėta 2005 11 11] Prieiga per internetą: <<http://www.leidykla.vu.lt/inetleid/inf-mok/10/str8.html>>.
- [9] AIIM International. How ODMA Integrates Application and DMS. 2001. [Žiūrėta 2007 02 10] Prieiga per internetą: <<http://odma.info/downloads/Q000500.htm>>.
- [10] W3C XML working group. Extensible Markup Language (XML) 1.0 (Fourth Edition), 2006 rugpjūčio 16, [žiūrėta 2007-12-10]. Prieiga per internetą: <<http://www.w3.org/TR/REC-xml/>>.
- [11] NISO, Understanding Metadata, 2001. [Žiūrėta 2005 11 10] Prieiga per internetą: <<http://www.niso.org/standards/resources/UnderstandingMetadata.pdf>>.
- [12] A. Brand, F. Daly, B. Meyers. Metadata demystified. A guide for publishers. NISO Press. 2003. [Žiūrėta 2005 11 10] Prieiga per internetą: <http://www.niso.org/standards/resources/Metadata_Demystified.pdf>.
- [13] XML Schema Part 0: Primer Second Edition. W3C. [interaktyvus] 2004 [žiūrėta 2006-12-10]. Prieiga per internetą: <<http://www.w3.org/TR/xmlschema-0/>>.
- [14] Mertz D., XML Matters: Transcending the limits of DOM, SAX, and XSLT [interaktyvus]. 2000. [žiūrėta 2007 03 12] Prieiga per internetą: <<http://www-128.ibm.com/developerworks/xml/library/x-matters14.html>>.
- [15] XSL Transformations (XSLT) Version 1.0. W3C [interaktyvus]. 1999 [žiūrėta 2007-03-10]. Prieiga per internetą: <<http://www.w3.org/TR/xslt.html>>
- [16] Ford P. Why Berkeley DB XML: An Embedded XML Database [interaktyvus]. 2003. [žiūrėta 2005 11 08] Prieiga per internetą: <<http://www.xml.com/pub/a/2003/05/07/bdb.html>>.

- [17] Steegmans B., Bourret R., Cline O., XML for DB2 Information Integration. 2004. IBM.Com/Redbooks, 696 p.
- [18] Y. Goland, E. Whitehead, A. Faizi, S. Carter, D. Jensen. HTTP Extensions for Distributed Authoring – WEBDAV [interaktyvus]. 1999. [žiūrėta 2007 01 22] Prieiga per internetą: <<http://www.ietf.org/rfc/rfc2518.txt>>.
- [19] Rezgui Y., Debras P. An Integrated Approach for a Model Based Document Production and Management. Electronic Journal of Information Technology in Construction, 1996, 1, 1–21 p.
- [20] Dourish P., Edwards W. K., LaMarca A., Lamping J., Petersen K., Salisbury M., Terry D. B. and Thornton J. Extending Document Management Systems with User-Specific Active Properties. ACM Transactions on Information Systems, 2000, 18/2, 140–170 p.
- [21] Foo J. DocPlayer: Design Insights from Applying the Non-Hierarchical Media-Player model to Document Management [interaktyvus]. 2003. [žiūrėta 2007 04 20] Prieiga internete: <www.diva-portal.org/diva/getDocument?urn_nbn_se_liu_diva-2086-1__fulltext.pdf>.
- [22] Jones J. An Alternative Document Management Model: Query-based Access to Hierarchical File systems, 2002. [žiūrėta 2007 05 14] Prieiga per internetą: <http://innovexpo.itee.uq.edu.au/2002/projects/s354151/thesis.pdf>.
- [23] Lyytikäinen V. Contextual and Structural Metadata in Document Management [interaktyvus]. 2004. [žiūrėta 2007 04 14] Prieiga per internetą: <<http://selene.lib.jyu.fi:8080/vaitos/studies/studcomp/9513917835.pdf>>.
- [24] Paganelli, F. A Model-driven Method for the Design and Deployment of Web-based Document Management Systems. 2005. <<http://journals.tdl.org/jodi/article/view/jodi-170/149>>.
- [25] AVATAR. Distribution accounting software for publishers [interaktyvus]. [žiūrėta 2006 04 14] Prieiga per internetą: <<http://www.avatar-software.com/modules/>>.
- [26] The ultimate database for publishers who recognize the significance of product management [interaktyvus]. [žiūrėta 2005 11 27] Prieiga per internetą: <http://www.qsolution.com/Publishing_Software.htm>.
- [27] Integrating the Typefi Publishing System to a Content Management System [interaktyvus]. [žiūrėta 2007 04 27] Prieiga per internetą: <http://www.typefi.com/whitepapers/Integrating_TPS_with_Content_Management_Systems.pdf>

7. TERMINŲ IR SANTRUMPŲ ŽODYNAS

PI – programinė įranga.

DBVS – duomenų bazės valdymo sistema.

HTML – hiperteksto žymų kalba.

SMTP – paprastas pašto persiuntimo protokolas.

PHP – programavimo kalba (interpretatorius).

SQL – struktūrinė užklausų kalba.

UML (*Unified Modeling Language*) – unifikuota modeliavimo kalba.

HTTP (*Hyper Text Transfer Protocol*) – protokolas, kuriuo perduodama informacija tarp serverio ir interneto naršyklės.

HTTPS – apsaugotas tinklo protokolas interneto duomenims perduoti.

OS – operacinė sistema.

DVS – dokumentų valdymo sistema.

XML (*eXtensible Markup Language*) – išplėstinė žymėjimo kalba.

DTD (*Document Type Definition*) – XML dokumento struktūros aprašas.

XSLT (*Extensible Stylesheet Language Transformations*) – XML transformacijų kalba.

XSL-FO (*Formatting Objects*) – kalba, atsakinga už XML dokumento vaizdą po XSLT transformacijos.

Onix (*Online Information Exchange*) – tarptautinis knygų prekybos metaduomenų standartas.



UAB „Smaltijos“ leidykla

Įmonės kodas 135276155, Kapsų g. 82, 44144 Kaunas, telefonas (37) 42 54 02, faksas (37) 20 89 92

PERDAVIMO-PRIĖMIMO AKTAS

DĖL KNYGŲ LEIDYKLOS DOKUMENTŲ VALDYMO SISTEMOS ĮDIEGIMO

2007 m. sausio mėn. 26 d.

Kaunas

UAB „Smaltijos“ leidykloje įdiegta *Knygų leidyklos dokumentų valdymo sistema*, atitinka reikalavimų specifikacijoje suformuluotus techninius, funkcinius bei nefunkcinius užsakovo reikalavimus. Realizuoti ir leidyklos serveryje įdiegti šie programinės įrangos moduliai:

- Informacijos apie knygą valdymas.
- Leidybinių knygos rengimo projektų valdymas.
- Dokumento būsenos valdymas.
- Administracinių dokumentų, susijusių su rengiama knyga, valdymas.
- ISBN registravimas.
- Užimtumo ataskaitų generavimas.
- Dokumento paieška.
- Dokumentų versijų palaikymas.
- Sistemos vartotojų, jų grupių ir rolių administravimas.

Direktorius

Zigmas Žygus

Autorius

Aistė Trepekaitė