Univerzitet u Novom Sadu
Fakultet tehničkih nauka
Katedra za računarstvo i automatiku

Proširivi sistem za pronalaženje multimedijalnih dokumenata

— doktorska disertacija —

Kandidat:
mr Branko Milosavljević

Mentor:
prof. dr Zora Konjović

maj 2003.

Sadržaj

Predgovor

Oblast pronalaženja informacija (*information retrieval*, IR) bavi se, pre svega, problemima pronalaženja i pristupa informacijama. Pronalaženje potrebnih informacija predstavlja glavnu namenu IR sistema. Osnovni nosioci informacija u IR sistemima su dokumenti. Pojam i modeli dokumenata u IR sistemima su se razvijali u poslednjim decenijama od nestrukturiranih tekstualnih dokumenata do strukturiranih multimedijalnih dokumenata. Pod multimedijalnim dokumentima podrazumevaju se strukture čiji elementi mogu pripadati različitim tipovima medija (npr. tekst, slike, audio i video zapisi i sl.).

Problem pretraživanja multimedijalnih objekata je, globalno posmatrano, u literaturi tretiran na dva načina. Istraživanja u prvoj grupi bave se pretraživanjem objekata koji pripadaju istom tipu medija, ali se posmatrani tip medija označava kao multimedijalni (tzv. *single-media retrieval*). Ovakvo pretraživanje razlikuje se od pretraživanja klasičnih tipova podataka po tome što se u procesu pretrage ne koristi egzaktno već aproksimativno poređenje sadržaja objekata. Ova razlika se u literaturi najčešće naglašava kao razlikovanje pojmova pronalaženje informacija (*information retrieval*) i pronalaženje podataka (*data retrieval*).

Istraživanja iz druge grupe imaju za cilj da omoguće uniforman tretman za različite tipove medija u procesu pronalaženja objekata. U smislu mogućnosti primene istih metoda pronalaženja objekata na različite tipove medija, ove metode se zaista mogu nazvati multimedijalnim (tzv. *multi-media retrieval*). Osnovna jedinica koja predstavlja predmet procesa pronalaženja, multimedijalni objekat, u okviru ove grupe istraživanja shvata se dvojako. Sa jedne strane nalaze se istraživanja koja multimedijalne objekte tretiraju kao istraživanja

koja multimedijalne objekte tretiraju kao nezavisne pojave različitih tipova medija. Sa druge strane, postoje istraživanja koja multimedijalne objekte tretiraju kao složene objekte koji poseduju elementarne multimedijalne sadržaje organizovane u određenu strukturu. Ovakav tretman pojma multimedijalnog objekta odgovara pojmu multimedijalnog dokumenta.

Tema disertacije pripada oblasti pronalaženja strukturiranih multimedijalnih dokumenata. Ovom problemu posvećen je relativno mali broj istraživanja, imajući u vidu ukupan broj objavljenih rezultata koji se odnose na pronalaženje multimedijalnih objekata shvaćenih u širem smislu. Postojeći rezultati u ovoj oblasti definišu modele strukturiranih multimedijalnih dokumenata i modele njihovog pronalaženja koji, po pravilu, nameću svoj model sadržaja i/ili svoj model pronalaženja za sve tipove medija.

Cilj disertacije je razvoj modela proširivog sistema za pronalaženje strukturiranih multimedijalnih dokumenata (*extensible multimedia information retrieval system*, XMIRS) koji omogućava integraciju različitih postojećih rešenja za pronalaženje multimedijalnih objekata i upotrebu različitih modela pronalaženja dokumenata. Na ovaj način, sistem za pronalaženje dokumenata ne nameće svoj model pronalaženja niti svoj model reprezentacije sadržaja njihovih multimedijalnih elemenata.

Modularnost XMIRS sistema predstavlja mogućnost upotrebe postojećih rešenja za pronalaženje multimedijalnih objekata. XMIRS sistem sastoji se od jezgra i modula namenjenih pronalaženju objekata. Sposobnost XMIRS-a da omogući kombinovanje funkcionalnosti različitih modula u okviru izabranog modela pronalaženja dokumenata donosi novi kvalitet procesu pronalaženja. Ovaj kvalitet se ogleda u dva aspekta: (1) mogućnost formiranja kriterijuma za pronalaženje dokumenata koji se ne mogu izraziti funkcijama pojedinih modula posmatranih nezavisno, i (2) mogućnost prilagođavanja sistema potrebama vezanim za konkretnu primenu, kroz dodavanje odgovarajućih modula za pronalaženje objekata i dodavanje modela pronalaženja dokumenata.

Tekst disertacije sastoji se iz četiri poglavlja.

Prvo poglavlje sadrži prikaz objavljenih rezultata u oblasti pronalaženja informacija. Rezultati se mogu podeliti u četiri glavne kategorije: (1) istraživanja koja se bave tekstualnim IR sistemima, (2) istraživanja koja se bave

pronalaženjem slika po sadržaju, (3) ona koja se bave analizom video zapisa i (4) istraživanja koja se bave pronalaženjem multimedijalnih objekata. Tretman drugih tipova medija je u literaturi znatno manje zastupljen, tako da se ove četiri grupe izdvajaju kao osnovni pravci istraživanja.

Centralni deo disertacije dat je u drugom poglavlju. Ono sadrži formalnu specifikaciju XMIRS sistema za pronalaženje strukturiranih multimedijalnih dokumenata. Formalnom specifikacijom obuhvaćeni su:

- definicije odgovarajućih pojmova vezanih za XMIRS,
- model dokumenata XMIRS-a,
- modeli pronalaženja dokumenata koji su prilagođeni za XMIRS i
- softverska arhitektura XMIRS-a koja omogućava proširivost sistema različitim modulima za pronalaženje multimedijalnih objekata i proširivost sistema različitim modelima pronalaženja dokumenata.

Prikazani sistem prevazilazi sledeće probleme postojećih modela, odnosno sistema: (1) rukovanje dokumentima koji imaju unapred određenu, fiksnu strukturu, (2) rad sa ograničenim brojem tipova medija, i (3) nametanje sopstvenog modela pronalaženja za sve podržane tipove medija.

U trećem poglavlju disertacije izvršena je verifikacija prikazanog modela na konkretnom realnom sistemu – mreži digitalnih biblioteka doktorskih i magistarskih radova (*networked digital library of theses and dissertations*, NDLTD). Verifikacija je sprovedena pomoću razvijenog prototipa XMIRS-a konfigurisanog za konkretnu primenu, što obuhvata sledeće aktivnosti:

- modeliranje dokumenata digitalne biblioteke,
- definisanje korisničkih potreba vezanih za mogućnosti procesa pronalaženja dokumenata,
- prikaz upotrebljenih postojećih softverskih modula koji obezbeđuju pojedine elemente funkcionalnosti pretraživanja,
- prikaz konfiguracije sistema koja povezuje jezgro sistema i upotrebljene module u funkcionalnu celinu i
- ilustraciju primera korišćenja ovako formiranog sistema.

Prikazana struktura dokumenata digitalne biblioteke obuhvata četiri tipa medija (strukturirani tekst, nestrukturirani tekst, bitmapirane slike i video zapise). Demonstracija kombinovane upotrebe raznorodnih modula za pretraživanje predstavlja verifikaciju osnovne funkcionalnosti XMIRS-a: pronalaženja

dokumenata po kriterijumima koji obuhvataju ograničenja postavljena nad različitim tipovima medija.

Četvrto poglavlje sadrži zaključna razmatranja, analizu rezultata i doprinosa disertacije, kao i analizu pravaca daljih istraživanja.

Bibliografija sadrži bibliografske jedinice koje su direktno ili indirektno pomenute u tekstu disertacije.

Zahvaljujem svim članovima Komisije koji su svojim korisnim sugestijama doprineli da ona bude jasnija i preglednija. Posebnu zahvalnost dugujem mentoru prof. dr Zori Konjović i prof. dr Dušanu Surli za nesebičnu podršku u toku izrade disertacije.

Novi Sad, maj 2003.

Branko Milosavljević

Poglavlje 1

Uvodna razmatranja

Oblast pronalaženja informacija (*information retrieval*, IR) bavi se problemima reprezentacije, skladištenja, organizacije i pristupa informacijama [23]. Reprezentacija i organizacija objekata koji su nosioci informacija trebalo bi da omoguće korisniku zadovoljenje njegove potrebe za informacijama (*information need*) na jednostavan način. IR sistemi najčešće rukuju dokumentima kao nosiocima informacija. Klasični IR sistemi rukuju tekstualnim dokumentima, dok se savremena istraživanja u ovoj oblasti bave različitim tipovima kolekcija dokumenata, kao što su kolekcije slika, video zapisa i strukturiranih multimedijalnih dokumenata.

Klasična monografija iz ove oblasti [289] i novija monografija [23] naglašavaju razliku između pronalaženja podataka (*data retrieval*) i pronalaženja informacija (*information retrieval*). Pronalaženje podataka se, u kontekstu IR sistema, posmatra kao određivanje dokumenata čije su karakteristike navedene u okviru formulacije upita kojim se pronalaženje inicira. Sa druge strane, korisnik IR sistema je više zainteresovan za dobijanje informacija o nekoj temi nego za dobijanje podataka koji zadovoljavaju postavljeni upit. Pronalaženje podataka ima za cilj pronalaženje svih objekata koji zadovoljavaju jasno definisane uslove. Ukoliko rezultat pronalaženja podataka uključuje bar jedan objekat koji ne zadovoljava postavljene uslove, sam postupak pronalaženja smatra se nekorektnim. U ambijentu IR sistema, međutim, postupak pronalaženja podrazumeva nepreciznost i može da sadrži greške. Osnovni uzrok ove

osobine pronalaženja je što se ono često bavi sadržajima formiranim na prirodnim jezicima koji mogu biti neprecizni ili semantički višeznačni.

Da bi se postigla određena efikasnost u korišćenju IR sistema, oni moraju biti u stanju da „interpretiraju“ sadržaj koji pretražuju i da ga rangiraju prema nivou *relevantnosti* za upit korisnika. Pojam relevantnosti je jedan od centralnih pojmova u oblasti IR.

U [23] data je sledeća formalna definicija modela pronalaženja informacija.

Definicija 1.1 *Model pronalaženja informacija je uređena četvorka* (D, Q, F, G) *gde je*

1. D *je skup reprezentacija dokumenata u kolekciji.*
2. Q *je skup reprezentacija korisničkih potreba za informacijama. Ove reprezentacije nazivaju se* upiti.
3. F *je okvir* (framework) *za modelovanje reprezentacija dokumenata, upita i njihovih veza.*
4. $G : Q \times D \to \mathcal{R}$ *je funkcija koja dodeljuje relan broj svakom paru upita* q_i *i dokumenta* d_j*. Ova funkcija definiše rangiranje dokumenata u odnosu na upit* q_i*.*

Monografija [23] naglašava i razliku između dva tipa interakcije korisnika sa IR sistemom: pronalaženja (*retrieval*) i pregledanja (*browsing*). Klasični IR sistemi su orijentisani na pronalaženje, dok su npr. hipertekst sistemi obično koncipirani kao sistemi za pregled dokumenata. Kombinovanje pronalaženja i pregledanja je trenutno slabo zastupljen pristup.

Članak [25] razmatra karakteristike dva različita tipa pronalaženja: tzv. *ad hoc* pronalaženje i filtriranje (*filtering*). Kod konvencionalnih IR sistema kolekcija dokumenata kojima sistem rukuje relativno je sporo promenljiva tokom vremena, dok su upiti kratkotrajna i promenljiva kategorija. Ovakav način rada nazvan je *ad hoc* pronalaženje. Sa druge strane, filtriranje podrazumeva sistem u kome su upiti relativno statični, dok se novi dokumenti relativno često dodaju sistemu. Sistemi za filtriranje koriste interne reprezentacije korisničkih potreba za informacijama nazvane korisnički profili (*user profiles*). Osnovna razlika profila i upita je u tome što su profili namenjeni opisivanju potreba za informacijama koje su trajnog karaktera. Analiza karakteristika konvencionalnih IR sistema i sistema za filtriranje sprovedena u [25] pokazuje da je sličnost

IR sistema i sistema za filtriranje velika; ovakvi sistemi na sličan način rešavaju problem poređenja upita i sadržaja dokumenata i njihovo rangiranje.

U članku [180] identifikuju se četiri dimenzije dokumenata. U okviru prikazanog modela kaže se da je dokument *jednostavan* ako se ne može dalje dekomponovati na druge dokumente. Jednostavan dokument je uređeni skup simbola koji nose informaciju putem značenja, time učestvujući u onome što se naziva sadržaj dokumenta. Jednostavni dokumenti imaju dve dimenzije: formu (ili sintaksu) i sadržaj (ili semantiku). *Složeni* dokumenti (ili samo dokumenti) su strukture čiji elementi su jednostavni dokumenti. Struktura dokumenata se posmatra kao treća dimenzija karakterizacije dokumenata. Dokumenti, jednostavni ili složeni, postoje kao nezavisni entiteti karakterisani atributima (*attributes*, *metadata*) koji opisuju relevatne osobine dokumenata. Skup ovakvih atributa naziva se profil (*profile*) dokumenta i predstavlja četvrtu dimenziju dokumenata.

Pretraživanje dokumenata po formi uzima u obzir sintaksne osobine dokumenata. Upiti za pretraživanje po formi mogu sami po sebi biti dokumenti (tzv. uzorci, *samples*), naročito u slučaju pretraživanja multimedijalnih dokumenata. Poređenje upita i dokumenta se tada svodi na poređenje njihovih internih reprezentacija.

Pretraživanje zasnovano na semantici podrazumeva korišćenje simboličkih reprezentacija značenja dokumenata, odnosno opisa formulisanih na nekom od jezika za reprezentaciju znanja. Primer ovakvog pretraživanja je pretraživanje zasnovano na analizi prostornih odnosa između objekata na slici [104]. Tipično se reprezentacije značenja konstruišu ručno (od strane čoveka), uz eventualnu upotrebu pomoćnog alata.

Pretraživanje dokumenata po formi može biti shvaćeno kao alternativni pristup problemu koga rešava pretraživanje po sadržaju, u smislu određivanja relevantnosti, odnosno veze između značenja sadržanog u upitu i u dokumentima. Ovakav pristup podrazumeva postojanje sistematske veze između „jednakosti“ osobina niskog nivoa (datih formom) i „jednakosti“ na nivou značenja. U literaturi iz ove oblasti postoji terminološka neusaglašenost pre svega u pogledu višedimenzionalne karakterizacije dokumenata. Tako se dešava da neki autori pod pronalaženjem po sadržaju (*content-based retrieval*) podrazumevaju ono što je ovde definisano kao pronalaženje po formi (*form-based retrieval*). U daljem tekstu će takve razlike biti eksplicitno naglašene.

Za efikasno pronalaženje traženih objekata najčešće se koriste specijalizovane strukture podataka nazvane *indeksi*. Danas je poznat veliki broj ovakvih struktura podataka; mogu se podeliti na indekse opšte namene (npr. binarna stabla i *hash*-tabele [143]) i indekse specijalizovane za određene specifične primene (npr. R-stabla [108] i 2D stringovi [46]). Pregled algoritama i struktura podataka koji se koriste u klasičnim (tekstualnim) IR sistemima dat je u [91].

Sistemi za indeksiranje mogu se klasifikovati prema tri aspekta [283, 289]:

- prema tome da li je sadržaj indeksa generisan automatski ili ručno,
- prema tome da li elementi indeksa pripadaju kontrolisanom rečniku ili ne i
- prema tome da li se elementi indeksa mogu kombinovati u niz elemenata koji predstavlja novi koncept prilikom indeksiranja (*pre-coordinated*) ili se oni kombinuju prilikom izvršavanja upita (*post-coordinated*).

Najznačajnija podela je prema načinu generisanja sadržaja indeksa. Ručno generisanje najčešće podrazumeva upotrebu znanja eksperta iz odgovarajuće oblasti. Eksperti su u mogućnosti da indeksom predstave semantiku dokumenata koje obrađuju. Kvalitet ovih informacija je daleko veći nego kod sistema sa automatskom ekstrakcijom semantike. Sa druge strane, troškovi angažovanja eksperata i vreme potrebno za formiranje indeksa su dovoljan razlog za istraživanja u oblasti automatske ekstrakcije semantike dokumenata.

Sistem prikazan u [187] predstavlja primer sistema sa ručnim generisanjem indeksa. Dokumenti kojima ovaj sistem rukuje su strukturirani opisi bibliotečke građe. Opise formiraju eksperti (bibliotekari) i oni se dalje koriste za pretraživanje građe. Iako se sami dokumenti sistema (bibliografski zapisi) indeksiraju automatski, sam sadržaj dokumenata predstavlja ručno kreirani indeks nad stvarnim dokumentima koje biblioteka poseduje. Sličan princip, predstavljen u [105], zasniva se na dodeljivanju tekstualnih opisa slikama od strane eksperata. Pretraživanje slika obavlja se korišćenjem pridruženih tekstualnih opisa.

Formulacija upita, bez poznavanja kolekcije dokumenata koja se pretražuje i načina funkcionisanja IR sistema, pokazuje se kao veliki problem za korisnika. U slučaju da se prilikom prvog upita ne dobiju potrebni rezultati, korisnik često pristupa reformulaciji upita. Ovakva analiza ponašanja korisnika donosi ideju da se prva formulacija upita tretira kao inicijalni (i obično slabo uspešan) pokušaj korisnika. Tehnike za automatizaciju procesa reformulacije

upita su u značajnoj meri izučavane u istraživanjima, ali su slabo zastupljene u komercijalnim IR sistemima. Tehnike se mogu podeliti u tri kategorije [23]: (a) tehnike zasnovane na povratnoj informaciji (*feedback*) korisnika, (b) tehnike zasnovane na analizi skupa dokumenata koji su inicijalno pronađeni (često nazvanog lokalni skup dokumenata) i (c) tehnike zasnovane na analizi celokupne (globalne) kolekcije dokumenata.

Tehnike za reformulaciju upita zasnovane na povratnoj informaciji korisnika o relevantnosti pronađenih dokumenata nazivaju se *relevance feedback* (RF) tehnike. One predstavljaju najčešće izučavani način za reformulaciju upita. Prva istraživanja RF tehnika sprovedena su u okviru SMART sistema [232], ali je i danas to aktivno polje istraživanja, pre svega za pretraživanje slika [167, 304]. Osnova svih RF tehnika je iterativni pristup formulaciji upita, pri čemu korisnik samo inicijalni upit formuliše korišćenjem upitnog jezika; kasnije reformulacije upita se izračunavaju u okviru sistema na osnovu podataka o relevantnosti pronađenih dokumenata koje je formirao korisnik. Primena RF tehnika najviše je izučavana u okviru pronalaženja tekstualnih dokumenata i slika. U [180] dat je sledeći opis RF procesa koji važi kako za tekst, tako i za slike:

1. Korisnik postavlja upit sistemu.
2. Sistem pronalazi k najbolje rangiranih dokumenata, sortiranih po nivou procenjene relevantnosti za korisnika. Ako je korisnik zadovoljan rezultatima pretrage, proces pronalaženja je završen.
3. Za $p \leq k$ najbolje rangiranih dokumenata korisnik unosi sopstvenu procenu njihove relevatnosti. Najčešće je u pitanju binarna vrednost (relevantan / nije relevantan) ali kod nekih sistema to može biti i *fuzzy* vrednost. Moguće je i ne izraziti nikakvu procenu za dokument.
4. Sistem menja svoje stanje uzimajući u obzir podatke dobijene od korisnika.
5. Korak 2 se ponavlja da bi se pronašlo novih k najbolje rangiranih dokumenata prema tekućem stanju sistema.

Tehnike za reformulaciju upita zasnovane na analizi lokalnog ili globalnog skupa dokumenata polaze od ideje o uočavanju međusobne sličnosti dokumenata po nekom kriterijumu. Reformulacija upita obuhvata pronalaženje novih izraza koji će, u okviru upita, bolje reprezentovati tražene dokumente.

Rad [310] ispituje efikasnost metoda lokalne i globalne analize dokumenata i navodi bolje rezultate metoda zasnovanih na analizi lokalnog skupa. Analiza lokalnog ili globalnog skupa se često tretira kao problem formiranja klastera dokumenata u prostoru definisanom na odgovarajući način. Pretpostavka od koje se polazi [164] je da ako su najbolje rangirani dokumenti relevantni, formiraće klaster koji neće obuhvatati nerelevantne dokumente. Klasteri se, pored toga, mogu koristiti za formiranje struktura nalik tezaurusima koje se, u ovom slučaju, ne odnose na pojmove (tj. reči) nego na dokumente kao celine i predstavljaju asocijacije sličnosti među njima.

1.1 Pronalaženje tekstualnih dokumenata

1.1.1 IR modeli za nestrukturirane dokumente

Klasični IR sistemi omogućavaju pretragu tekstualnih dokumenata. Pri tome se pod tekstualnim dokumentima podrazumevaju nestrukturirani dokumenti, čiji sadržaj je slobodan tekst. Kolekcije dokumenata kao što su TREC [115, 295] i CACM [90] namenjene su vrednovanju performansi tekstualnih IR sistema. Njih čine dokumenti koji imaju određene elemente strukture namenjene za metapodatke (npr. identifikator dokumenta, poreklo), dok je sam sadržaj dokumenta nestrukturiran.

U literaturi je definisan određeni broj modela za pretraživanje nestrukturiranih tekstualnih dokumenata. Neki od ovih modela se mogu modifikovati tako da se prilagode i drugačijim tipovima dokumenata, što će biti i izloženo u narednim odeljcima. U [23] prikazana je sistematizacija do sada poznatih IR modela za nestrukturirane tekstualne dokumente:

- Klasični modeli
  - bulovski model
  - vektorski model
  - probabilistički model
- Alternativni modeli
  - Zasnovani na teoriji skupova
    - *fuzzy* model
    - prošireni Bulovski model

— Algebarski
  * generalizovani vektorski model
  * *latent semantic indexing*
  * neuronske mreže
— Probabilistički
  * *inference network* model
  * *belief network* model

Bulovski model predstavljen je na više mesta [245, 298]. Vektorski model, nastao kao deo istraživanja vezanih za SMART sistem [240], definisan je u [244]. Probabilistički model prikazan je u [230], a detaljno analiziran u [289]. Fazi model predložen je u [201], mada je i pre njega bilo istraživanja na temu proširivanja Bulovskog modela fazi konceptima [221, 222]. Prošireni Bulovski model opisan je u [243]. Generalizovani vektorski model definisan je u [303]. *Latent semantic indexing* model prikazan je u [94]. Model zasnovan na neuronskim mrežama dat je u [300]. Probabilistički modeli zasnovani su na Bajesovim mrežama [207]. *Inference network* model prikazan je u [284, 285]. Njegova generalizacija, *belief network model*, dat je u [229]. U tekstu će biti prikazane osnovne karakteristike nekih od pomenutih modela koji su od interesa za dalje izlaganje u okviru disertacije.

Klasični IR sistemi polaze od ideje da je svaki dokument predstavljen skupom reprezentativnih reči nazvanih izrazi indeksa (*index terms*). Izraz indeksa je, prema [23], reč iz dokumenta čija semantika pomaže u pamćenju osnovne teme dokumenta. Za dati skup izraza indeksa uočava se da nisu svi izrazi jednako korisni za opisivanje sadržaja dokumenta. Zbog toga se izrazima dodeljuju numeričke težine (*weights*), kao kvantitativna mera relevantnosti izraza indeksa. Sledeća definicija daje odnos pojmova izraza indeksa, dokumenta i težine.

Definicija 1.2 *Neka je* $K = \{k_1, k_2, \ldots, k_t\}$ *skup izraza indeksa svih dokumenata u sistemu. Težina* $w_{ij} \geq 0$ *dodeljena je izrazu indeksa* k_i *u dokumentu* d_j*. Za izraz indeksa koji se ne pojavljuje u dokumentu* d_j *važi da je* $w_{ij} = 0$*. Dokumentu* d_j *pridružen je vektor* $\vec{d_j} = (w_{1j}, w_{2j}, \ldots, w_{tj})$*. Funkcija* f_i *je takva da je* $f_i(\vec{d_j}) = w_{ij}$*.*

Bulovski model je zasnovan na teoriji skupova i Bulovoj algebri. Upiti se formiraju kao logički izrazi koji imaju preciznu semantiku. Jednostavan je

za implementaciju i koristi ga većina starijih komercijalnih sistema kao što je Dialog [68]. U okviru bulovskog modela izrazi indeksa ili jesu prisutni u dokumentu ili nisu. Kao posledica, pridružene težine uzimaju vrednost iz skupa $\{0, 1\}$. Upit q se sastoji iz izraza indeksa i logičkih operatora *and*, *or* i *not*. (U nekim sistemima kao što je [187] operator *not* tretira se kao uobičajeni *and not*, čime se postiže da su svi operatori binarni). Upit je, dakle, logički izraz koji se može predstaviti i u disjunktivnoj normalnoj formi. Na primer, upit $q = k_1 \wedge (k_2 \vee \neg k_3)$ može se predstaviti kao $\vec{q}_{dnf} = (1, 1, 1) \vee (1, 1, 0) \vee (1, 0, 0)$, gde su konjuktivne komponente predstavljene odgovarajućim vektorom težina asociranim sa uređenom trojkom (k_1, k_2, k_3). Sledeća definicija daje meru sličnosti dokumenta iz kolekcije i datog upita.

Definicija 1.3 *Neka je q logički izraz i $\vec{q}_{dnf}$ njegova disjunktivna normalna forma. Neka je q_{cc} bilo koja konjuktivna komponenta iz $\vec{q}_{dnf}$. Sličnost dokumenta d_j sa upitom q izražava se vrednošću funkcije*

$$sim(d_j, q) = \begin{cases} 1, & \exists \vec{q}_{cc} \mid (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i \;\; f_i(\vec{d_j}) = f_i(\vec{q}_{cc})) \\ 0, & \text{inače} \end{cases} \tag{1.1}$$

Bulovski model omogućava da se dokument u kolekciji za dati upit karakteriše kao relevantan ili nerelevantan. Nemogućnost izražavanja delimičnog poklapanja dokumenta sa upitom (*partial match*) predstavlja njegovu osnovnu manu.

Vektorski model ispravlja osnovnu manu bulovskog modela tako što omogućava dodeljivanje ne-binarnih težina kao i izražavanje delimičnog poklapanja dokumenta sa upitom. Težine se, u okviru ovog modela, dodeljuju kako izrazima indeksa, tako i izrazima upita.

Definicija 1.4 *Težina $w_{ij} \geq 0$ dodeljuje se paru (k_i, d_j). Težina $w_{iq} \geq 0$ dodeljuje se paru (k_i, q). Vektor upita $\vec{q}$ definiše se kao $\vec{q} = (w_{1q}, w_{2q}, \ldots, w_{t,q})$ gde je t ukupan broj izraza indeksa u sistemu. Vektor dokumenta $\vec{d_j}$ definiše se kao $\vec{d_j} = (w_{1j}, w_{2j}, \ldots, w_{tj})$.*

Na ovaj način, vektor dokumenta i vektor upita mogu se posmatrati kao vektori u t-dimenzionalnom prostoru. Mera sličnosti između dva vektora najčešće se izražava na sledeći način [241]:

$$sim(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \cdot |\vec{q}|} = \frac{\sum_{i=1}^{t} w_{ij} \cdot w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \cdot \sqrt{\sum_{i=1}^{t} w_{iq}^2}} \tag{1.2}$$

Vrednost $sim(d_j, q)$ nalazi se u intervalu $[0, 1]$. Ukoliko se, prilikom postavljanja upita, definiše minimalna vrednost ove funkcije, mogu se kao rezultat izdvojiti samo oni dokumenti čija sličnost sa upitom prelazi dati prag. Rezultat upita se, pored toga, može sortirati po sličnosti sa upitom u opadajućem redosledu tako da se najbolje rangirani dokumenti nađu na vrhu liste rezultata.

Određivanje vrednosti težina za izraze indeksa može se obaviti na više načina. U [245] dat je prikaz različitih načina za ovu kalkulaciju. Često korišćen način zasniva se na formiranju klastera (*clusters*) dokumenata. Problem formiranja klastera ovde se svodi na podelu celokupne kolekcije dokumenata na dva dela: klaster sa dokumentima koji su relevatni za konkretnu potrebu korisnika i ostatak kolekcije dokumenata. Kao mera sličnosti između dokumenata koji se nalaze unutar traženog klastera (*intra-cluster similarity*) koristi se učestanost pojavljivanja izraza k_i u dokumentu d_j. Ova mera, u literaturi obično nazvana *tf factor*, određuje koliko dobro dati izraz opisuje sadržaj dokumenta. Mera različitosti između dokumenata iz različitih klastera (*inter-cluster dissimilarity*) je inverzna frekvencija izraza k_i (tzv. *idf factor*). Ideja formulacije ove mere je da se izrazi koji se pojavljuju u svim dokumentima teško mogu iskoristiti za razlikovanje relevatnih dokumenata od nerelevantnih. U nastavku je data formalna definicija ovih veličina.

Definicija 1.5 *Neka je N ukupan broj dokumenata u kolekciji i n_i broj dokumenata u kojima se pojavljuje izraz indeksa k_i. Neka je $freq_{ij}$ broj pojavljivanja izraza k_i u dokumentu d_j. Normalizovana učestanost pojavljivanja f_{ij} izraza k_i u dokumentu d_j data je kao*

$$f_{ij} = \frac{freq_{ij}}{\max_{l \in d_j} \{freq_{lj}\}} \tag{1.3}$$

gde se maksimum broja pojavljivanja određuje iz skupa svih izraza koji se javljaju u dokumentu d_j. Inverzna učestanost izraza k_i, u oznaci idf_i, definiše se na sledeći način:

$$idf_i = \log \frac{N}{n_i} \tag{1.4}$$

Vrednosti za težine w_{ij} dodeljene izrazima određuju se kao:

$$w_{ij} = f_{ij} \cdot \log \frac{N}{n_i} \tag{1.5}$$

Vrednosti za težine dodeljene upitu w_{iq} date su izrazom [241]*:*

$$w_{iq} = \left(0.5 + \frac{0.5 \ \ freq_{iq}}{\max_{l \in q} freq_{lq}}\right) \cdot \log \frac{N}{n_i} \tag{1.6}$$

Probabilistički model, poznat u literaturi i kao *binary independence retrieval* model, polazi od pojma *idealnog skupa* pronađenih dokumenata za dati upit koga čine svi dokumenti koje bi korisnik ocenio kao relevantne; nerelevantni dokumenti nisu elementi ovog skupa. Kako osobine tog skupa nisu poznate unapred, potrebno je učiniti inicijalno pogađanje. Nakon toga moguće je pronaći inicijalni skup dokumenata. Zatim se interakcija sa korisnikom odvija u cilju unapređivanja probabilističkog opisa idealnog skupa.

Za dati upit q i dokument d_j potrebno je odrediti verovatnoću da će dokument d_j biti relevantan za korisnika. Pretpostavka je da ova verovatnoća zavisi isključivo od karakteristika upita i dokumenta. Dalje, pretpostavlja se da idealan skup pogodaka R postoji.

Definicija 1.6 *U probabilističkom modelu, težine dodeljene izrazima indeksa i elementima upita su binarne, tj. $w_{ij} \in \{0, 1\}$ i $w_{iq} \in \{0, 1\}$. Upit q predstavlja podskup skupa svih izraza indeksa. Neka je R skup dokumenata koji se smatraju relevantnim. Neka je $\bar{R}$ komplement skupa R. Neka je $P(R|d_j)$ verovatnoća da je dokument d_j relevantan za upit q i $P(\bar{R}|d_j)$ verovatnoća da d_j nije relevantan za q. Sličnost dokumenta d_j sa upitom q izražava se kao*

$$sim(d_j, q) = \frac{P(R|d_j)}{P(\bar{R}|d_j)} \tag{1.7}$$

U [23] dat je izvedeni oblik prethodnog izraza koji glasi:

$$sim(d_j, q) \sim \sum_{i=1}^{t} w_{iq} w_{ij} \left(\log \frac{P(k_i|R)}{1 - P(k_i|R)} + \log \frac{1 - P(k_i|\bar{R})}{P(k_i|R)}\right) \tag{1.8}$$

gde je $P(k_i|R)$ verovatnoća da je izraz k_i prisutan u dokumentu slučajno izabranom iz skupa R. Analogno tome definisana je i veličina $P(k_i|\bar{R})$. Kako skup R nije poznat na početku pretraživanja, potrebno je odrediti inicijalne vrednosti za ove veličine. Pre prvog pretraživanja, u trenutku kada još nema pronađenih dokumenata, može se pretpostaviti da je $P(k_i|R)$ jednako za sve izraze k_i (tipično jednako 0.5) i da se raspodela izraza indeksa među nerelevantnim dokumentima može aproksimirati distribucijom svih izraza indeksa među svim dokumentima u kolekciji. Na taj način može se pisati

$$P(k_i|R) = 0.5 \tag{1.9}$$

$$P(k_i|\bar{R}) = \frac{n_i}{N} \tag{1.10}$$

Nakon što se pronađe inicijalni skup dokumenata koji zadovoljavaju upit, moguće je definisati skup V kao skup od prvih r rangiranih dokumenata gde je r unapred definisan prag. Neka je V_i skup onih dokumenata iz V koji sadrže izraz k_i. Usvajaju se sledeće pretpostavke: a) $P(k_i|R)$ može se aproksimirati raspodelom izraza k_i među pronađenim dokumentima, i b) $P(k_i|\bar{R})$ može se odrediti na osnovu pretpostavke da su svi nepronađeni dokumenti ujedno i nerelevantni. Imajući u vidu ove pretpostavke, možemo pisati:

$$P(k_i|R) = \frac{|V_i|}{|V|} \tag{1.11}$$

$$P(k_i|\bar{R}) = \frac{n_i - |V_i|}{N - |V|} \tag{1.12}$$

Prethodne formule se, zbog problema koje u praksi stvaraju određene vrednosti za V i V_i [23], modifikuju tako da glase:

$$P(k_i|R) = \frac{|V_i| - n_i/N}{|V| + 1} \tag{1.13}$$

$$P(k_i|\bar{R}) = \frac{n_i - |V_i| + n_i/N}{N - |V| + 1} \tag{1.14}$$

Poslednje formule mogu se koristiti u svim narednim iteracijama pretraživanja.

Kao mane probabilističkog modela u [23] navedene su sledeće karakteristike: a) potreba za pogađanjem inicijalne podele dokumenata na relevantne i

nerelevantne, b) model ne uzima u obzir učestanost ponavljanja izraza u dokumentu (težine w_{ij} su binarne) i c) pretpostavka da su izrazi indeksa međusobno nezavisni u smislu njihovog pojavljivanja u dokumentima. Glavna vrlina ovog modela je što se dokumenti rangiraju prema verovatnoći da su relevatni za korisnika.

Karakteristike ovde prikazanog probabilističkog modela eksperimentalno su analizirane u [267]. U članku [60] prikazana je varijanta modela koja ne koristi povratne informacije korisnika za procenu verovatnoća. Isti autor u [59] razmatra dodavanje težina vezanih za učestanost pojavljivanja unutar dokumenta u model.

U literaturi postoji opšte mišljenje da je Bulovski model najslabiji od klasičnih metoda, pre svega zbog nepostojanja mogućnosti za delimično poklapanje dokumenta sa upitom. U radu [163] razmatra se integracija Bulovskih upita u probabilistički model. Analiza eksperimenata sprovedena u [59] navodi da probabilistički model ima bolje performanse od vektorskog. Kasniji rad [241] opovrgava ovu tvrdnju i pokazuje da se može smatrati da vektorski model ima bolje performanse od probabilističkog za opšte kolekcije dokumenata. Vektorski model se može danas smatrati najpopularnijim modelom među istraživačima, ali i u praksi, pre svega u projektima web pretraživača [23].

1.1.2 IR modeli za strukturirane dokumente

U ovom odeljku biće reči o IR modelima pronalaženja strukturiranih tekstualnih dokumenata. Modeli pronalaženja multimedijalnih dokumenata najčešće obuhvataju pretragu po tekstu i po strukturi, ali će o njima biti reči u posebnom odeljku. Razvoj IR modela namenjenih pronalaženju strukturiranih dokumenata započeo je znatno kasnije u odnosu na razvoj klasičnih IR modela. Rani radovi [271, 66] bave se integracijom funkcija pretraživanja tekstualnih dokumenata u klasične relacione sisteme za upravljanje bazama podataka. Danas važi stav da kombinovanje sadržaja i strukture dokumenata u procesu pronalaženja dokumenata pruža veće mogućnosti u pretraživanju nego svaki mehanizam ponaosob. Pri tome se za pretraživanje po strukturi smatra da predstavlja pretraživanje podataka, a ne pretraživanje informacija, s obzirom na to da svi modeli podrazumevaju egzaktno poređenje struktura. Članak [22] iz 1996. godine sadrži analizu klasičnih IR modela i zaključak da oni ne mogu efikasno da podrže zahteve koji se postavljaju pred sisteme za pronalaženje

strukturiranih dokumenata. Pored toga, naglašava pravilo recipročnog odnosa između izražajnosti modela i efikasnosti njegove implementacije koje važi za analizirane modele za strukturirane dokumente.

Hibridni model (*hybrid model*, [20]) posmatra kolekciju dokumenata koji mogu imati definisana polja. Polja ne moraju prekrivati tekst u potpunosti, mogu se ugnježdavati i preklapati. Upitni jezik je definisan kao algebra nad parovima (D, M) gde je D skup dokumenata a M skup pozicija u tekstu koje mogu biti poređene sa rečima ili šablonima (*patterns*). Definisan je određeni broj operacija kojima se takve pozicije generišu, npr. pretraživanje po prefiksu reči, blizinski operatori itd. Skupovne operacije unije, preseka, razlike i komplementa definisane i za skupove dokumenata i za skupove pozicija (za ograničavanje pozicija samo unutar datih polja ili za pronalaženje polja koja sadrže datu poziciju). Zbog svoje jednostavnosti model je jednostavan za implementaciju.

PAT izrazi (*PAT expressions*, [237]), implementirani u okviru sistema *PAT Text Searching System* [84], sadrže indekse samo nad pozicijama, a ne i nad strukturom dokumenta. Prikazani jezik omogućava dinamičko definisanje strukture zasnovano na izrazima za definisanje pozicija koje određuju početak i kraj kontinualnog regiona teksta. Izrazi moraju biti specifičnog oblika, tako da su vezani za odgovarajući tip označavanja blokova. Sistem je uspešno primenjen u projektu OED (*Oxford English Dictionary*) [98].

Preklapajuće liste (*overlapped lists*) prikazane su u [55, 56]. Struktura dokumenata predstavlja se pomoću više lista koje se sastoje od disjunktnih, kontinualnih regiona teksta. Definisane operacije nad regionima su relativno jednostavne: selekcija datog regiona ili reči; selekcija regiona koji (ne) sadrži dati region; selekcija regiona koji (ni)je sadržan u drugom regionu. Pored toga, definisane su i klasične IR operacije kao što je rangiranje po relevantnosti.

Liste referenci (*lists of references*, [171, 170]) se, prilikom modeliranja dokumenata, oslanjaju na koncepte definisane SGML standardom [127] i koncepte poznate iz objektno-orijentisanih baza podataka, mada nisu direktno vezane za njih. Dokumenti poseduju tačno jednu hijerarhijsku strukturu, gde hijerarhija označava vezu sadržavanja među regionima dokumenta. Čvorovi strukture mogu posedovati atribute koji se mogu upotrebljavati u pretraživanju. Postoji i koncept linkova preuzet iz hipertekst sistema. Model poseduje veliku

izražajnost u modeliranju dokumenata i formiranju upita ali donosi probleme prilikom dizajna efikasne implementacije [171].

Poređenje stabala (*tree matching*, [139]) koristi koncept sadržavanja stabla (*tree inclusion*) za opisivanje strukture upita i baze podataka, pri čemu se kao osnovni problem posmatra pronalaženje onih delova strukture u bazi podataka koji odgovaraju šablonu datim pomoću upita. Razmatrane su dve varijante; uređeno sadržavanje (*ordered inclusion*) vodi računa o redosledu čvorova potomaka, dok neuređeno sadržavanje (*unordered inclusion*) taj redosled zanemaruje. Listovi šablona upita mogu biti izrazi koji sadrže tekstualni šablon. Rezultati upita su takođe stabla. Upitni jezik poseduje koncept promenljivih koje omogućavaju formiranje izraza za jednakost upita samo sa pojedinim delovima tražene strukture, kao i operacije unije i preseka i simulaciju operacije spoja poznate iz relacionog modela podataka. Rad [139] je formiran pre svega kao analiza osobina procesa poređenja stabala. Prema analizi u [140], neuređeno sadržavanje je NP-kompletan problem [96]. Samim tim, model nije pogodan za implementaciju u realnim sistemima.

Radovi [194, 193] predstavljaju model bliskih čvorova (*proximal nodes*) koji obuhvata model dokumenata i jezik pretraživanja. Osnovna namera autora je da se definiše model koji ima ravnotežu u pogledu izražajnosti modeliranja strukture dokumenata i upita i efikasne implementacije. Kolekcija (tj. baza u datoj terminologiji) dokumenata poseduje dve komponente:

- *tekst* koji se posmatra kao sekvenca simbola (bili oni karakteri, reči, ili nešto drugo). Nije bitno da li tekst sadrži oznake strukture (*markup*) ili ne; one se smatraju delom teksta.
- *strukturu* koja predstavlja skup međusobno nezavisnih hijerarhija. Regioni teksta koje pokrivaju čvorovi različitih hijerarhija mogu se preklapati, ali se to ne može desiti unutar iste hijerarhije. Hijerarhije ne moraju da pokriju celokupan tekst.

Definisana je algebra nad skupovima čvorova, sa odgovarajućim operatorima nad tim skupovima. Prikazana je i softverska arhitektura moguće implementacije ovakvog sistema. Performanse sistema su analizirane na nivou modela; implementacija sistema (u trenutku pisanja ovog teksta) ne postoji kao ni podaci o njenim performansama. Koncept modela je takav da se oslanja na neki od postojećih modela za fizičko rukovanje podacima (relacioni, objektno

orijentisani, itd) i ponavlja tezu istih autora datu u [22] da je poželjno iskoristiti postojeće modele podataka i njihove implementacije za one segmente rukovanja podacima u kojima su se pokazali kao efikasni. U radu se takođe razmatra i mogućnost upotrebe ovog modela za rad sa multimedijalnim dokumentima.

Rad [22] donosi i sistematizaciju modela namenjenih pronalaženju strukturiranih tekstualnih dokumenata. Sistematizacija je data po više aspekata koji će i ovde biti navedeni.

Izražajnost modela strukture. Kako u trenutku objavljivanja rada [22] nije postojao konsenzus oko strukturiranja baze dokumenata, izvršena je analiza izražajnosti pojedinih modela. Posmatrane su sledeće karakteristike:

- *Tip strukture dokumenata*
  - **Ravan.** Slučaj kada jedan element strukture ne može da sadrži drugi element.
  - **Hijerarhijski.** Najčešće korišćeni tip strukture, gde hijerarhija predstavlja odnos sadržavanja elemenata. Modeli se međusobno razlikuju po tome da li dozvoljavaju rekurzivne strukture, postojanje više nezavisnih hijerarhija ili preklapanje regiona teksta koji odgovaraju elementima strukture.
  - **Mrežni.** Nijedan od razmatranih modela ne poseduje karakteristike mrežnog modela, mada liste referenci poseduju mogućnost proizvoljnog povezivanja elemenata ali pre svega za potrebe navigacije.
- *Implicitna ili eksplicitna struktura.* Implicitnu strukturu koriste modeli koji se oslanjaju na parsiranje teksta ili posebne oznake strukture u tekstu (*markup*). Drugim rečima, ne poseduju odvojene podatke o strukturi dokumenata, već se oni nalaze unutar dokumenata. Modeli sa implicitnom strukturom ne mogu imati osobine koje se teško mogu predstaviti *markup* mehanizmom, kao što su rekurzivne strukture. Sa druge strane, preklapanje elemenata strukture se jednostavno izražava na ovaj način. Izračunavanje upita je drugi važan aspekt ove klasifikacije: modeli sa implicitnom strukturom mogu strukturne upite svesti na izraze za poređenje šablona

(*pattern matching*) i time koristiti implementacije poznate iz klasičnih IR sistema, dok modeli sa eksplicitnom strukturom mogu jednostavnije odgovoriti na upite o odnosima predak/potomak među elementima strukture.

- *Statička ili dinamička struktura.* Neki modeli funkcionišu dovoljno dobro uz pretpostavku da je struktura dokumenata manje ili više statička, tj. nepromenljiva tokom vremena. Drugi su više orijentisani na manipulaciju dinamičkim strukturama. Može se uopšteno reći da modeli sa implicitnom strukturom koriste dinamičke strukture dok modeli sa eksplicitnom strukturom očekuju statičke strukture.
- *Veza između sadržaja i strukture.* Ovaj aspekt prikazuje važnu osobinu svakog modela u pogledu njegove orijentacije ka određenoj vrsti upita. Neki modeli su orijentisani više ka pretraživanju teksta, dok su drugi namenjeni pretraživanju po strukturi. Podeljeni su u tri kategorije:
  - dominantno tekstualni (*strongly text-bound*)
  - dominantno strukturni (*strongly structure-bound*)
  - srednji (*intermediate*)
- *Struktura rezultata.* Struktura rezultata u pojedinim modelima ne mora biti ekvivalentna strukturi dokumenata. Uočene su tri tipa strukture rezultata:
  - ravni (*flat*)
  - preklapajući (*overlapped*)
  - ugnježdeni (*nested*)

Upitni jezik. Analiza upitnog jezika obuhvata nekoliko osobina:

- *Pretraživanje teksta.* Ovaj aspekt analizira mogućnosti upitnog jezika koje se odnose na poređenje teksta (*pattern matching*).
- *Manipulacija skupovima.* Svi modeli predstavljaju rezultat upita kao skup entiteta. Modeli koji nemaju ugnježdene rezultate predstavljaju taj skup kao uređenu listu. Većina modela definiše operacije unije, preseka i razlike skupova rezultata, mada svaki poseduje i svoje specifične mogućnosti.

- *Veze sadržavanja.* Pronalaženje elemenata strukture koji sadrže druge elemente ili su sadržani u drugim elementima podrazumeva poznavanje veza sadržavanja u dokumentima. Prikazani modeli najviše se razlikuju po tretmanu ovog koncepta.
- *Rastojanja.* Mogućnost izražavanja ograničenja na međusobno rastojanje elemenata teksta ili strukture je na različit način zastupljena kod različitih modela. Neki modeli zanemaruju ovu mogućnost koja se u praksi pokazuje kao izuzetno važna.

Složenost izračunavanja upita. Tipično je izražajnost modela i upitnog jezika obrnuto srazmerna računskoj složenosti izračunavanja upita. Uočeno je nekoliko klasa složenosti:

- $O(n)$: hibridni model, PAT izrazi i preklapajuće liste.
- Skoro uvek $O(n)$: bliski čvorovi omogućavaju implementaciju linearne složenosti u većini situacija.
- $O(n \log n)$: većina operacija u modelu liste referenci ima ovu složenost.
- Ne-polinomijalna: poređenje stabala definiše operacije koje imaju složenost veću od polinomijalne.

Nijedan od prikazanih modela ne razmatra problem rangiranja dokumenata prilikom formiranja rezultata upita. Jedna varijanta rangiranja strukturiranih dokumenata data je u [137]. Dokumenti se, za potrebe rangiranja, posmatraju kao nizovi pasusa, gde se pod pasusom podrazumeva niz reči fiksne dužine koji se pojavljuje bilo gde u tekstu. Prilikom pronalaženja dokumenata moguće je korisniku prikazati samo odgovarajući pasus, tako da je ovaj sistem posebno pogodan u situacijama gde se rukuje dokumentima izuzetno velike dužine (npr. sudski stenogrami). Iako je prikazani metod namenjen pre svega rangiranju nestrukturiranih dokumenata, za potrebe izračunavanja ranga koristi se struktura nad tekstom.

1.1.3 XML standard

Nakon objavljivanja XML preporuke [37], interesovanje za sisteme za rukovanje strukturiranim tekstualnim dokumentima je naglo poraslo. Izuzetna podrška softverske industrije ovom standardu, realizovana kroz razvojne alate,

alate za krajnje korisnike, mnoštvo standarda za razmenu dokumenata u pojedinačnim oblastima (finansije, medicina, hemija, geografski informacioni sistemi) zasnovanih na XML-u ustanovili su konsenzus u istraživačkoj zajednici oko usvajanja XML-a kao standardnog modela strukturiranih dokumenata. Iako SGML [127] standard postoji od 1986. godine, zbog svoje složenosti bio je ograničen na mali broj specifičnih primena. XML, zamišljen kao funkcionalni podskup SGML-a, za kratko vreme je usvojen kao standardan jezik u velikom broju primena zahvaljujući, između ostalog, i svojoj jednostavnosti.

Kao sredstvo za definisanje strukture XML dokumenata inicijalno je korišćen DTD (*Document Type Definition*) format, sastavni deo XML preporuke. Ograničene mogućnosti DTD-a u pogledu definisanja ograničenja na tipove podataka koji se koriste u dokumentima dovele su do donošenja novog standarda za definiciju strukture dokumenata, XML Schema [281, 30].

XML je praćen određenim brojem drugih preporuka koje se bave pojedinim aspektima rukovanja XML dokumentima. XSLT [53] je namenjen za transformacije XML dokumenata, XLink [65] definiše linkove među njima, a XPath [54] sintaksu za referenciranje pojedinih delova dokumenta. Pitanje standardnog upitnog jezika rešeno je definisanjem XQuery jezika [43], mada je pre njenog usvajanja definisano nekoliko upitnih jezika za XML (XQL [231], XML-QL [67], itd.) sa donekle različitim karakteristikama. Komparativna analiza karakteristika nekoliko predloženih upitnih jezika za XML data je u [31].

Interes istraživača iz oblasti IR tekao je u pravcu adaptiranja postojećih modela strukturiranih dokumenata za rad sa XML dokumentima. Rad [19] donosi analizu mogućnosti primene modela bliskih čvorova za implementaciju XQL upitnog jezika. Jedinstveni model strukture XML dokumenata ne postoji čak ni kada su u pitanju formalne specifikacije standarda. U okviru aktivnosti W3 konzorcijuma razvijena su četiri modela strukture dokumenata: *XML Information Set* model [58], XPath 1.0 model [54], DOM model [123] i XQuery 1.0 model [85]. Novina koju XML donosi u odnosu na ranije prikazane modele strukturiranih dokumenata je koncept atributa vezanog za element dokumenta. Atributi se, prema [238], tipično koriste za metapodatke, mada je i njihova upotreba za smeštanje podataka nekad moguća (da bi se izbeglo nametanje redosleda među čvorovima potomcima, ili za referenciranje na eksterno smeštene delove dokumenta).

Pitanje ekvivalencije XML dokumenata nema jedinstven odgovor. Neki autori zanemaruju redosled čvorova potomaka [4], dok ga drugi uzimaju u obzir [81]. W3C preporuka *Canonical XML* [35] definiše tzv. kanoničku formu dokumenata koji učestvuju u poređenju. Međutim, prevođenje sadržaja dokumenta u kanoničku formu zanemaruje neke podatke iz originalnog dokumenta.

Primena XML jezika kao standarda za razmenu podataka između raznorodnih informacionih sistema privukla je i istraživače iz oblasti baza podataka [292]. U radu [238] analizirane su potrebne karakteristike sistema za rukovanje XML dokumentima. Data je definicija baze XML dokumenata (*XML document database*) kao „kolekcija XML dokumenata i njihovih delova kojom rukuje sistem sposoban za upravljanje kolekcijom i informacijama reprezentovanim kolekcijom“. Rad [13] navodi sledeće osobine sistema za rukovanje strukturiranim dokumentima:

- kreiranje dokumenata na zahtev (*on-the-fly creation of renditions*)
- automatske transformacije dokumenata
- kontrola pristupa na nivou elemenata
- pristup samo pojedinim elementima
- verzije dokumenata
- opisi promena dokumenata čitljivi za čoveka
- podrška za rad zasnovan na tokovima dokumentima
- proširene mogućnosti pretraživanja (kombinovanje pretrage po sadržaju i strukturi, rangiranje pogodaka).

Rad [238] identifikuje važne karakteristike XML baza podataka. Prikazane karakteristike se ne odnose ni na jedan postojeći sistem, već su namenjene za kreiranje konteksta za vrednovanje pojedinih sistema. Kao osnovne karakteristike navedene su sledeće:

- podrška za bogat skup osnovnih tipova podataka, npr. kakav definiše XML Schema [30]
- mogućnost definisanja više tipova dokumenata, pomoću njihovih DTD-a ili XML Schema opisa
- rukovanje kolekcijama dokumenata, pri čemu se kolekcije dokumenata vide kao „ravne“, tj. ne postoji mogućnost ugnježdavanja kolekcija
- mogućnost rukovanja kolekcijama tipova dokumenata, npr. za potrebe definisanja strukture više verzija istog dokumenta

- rad sa više nivoa provere validnosti dokumenta
- pravilno rukovanje *entity reference* poljima u XML dokumentu i URI [27] adresama
- podrška za XML prostore imena (*namespaces*, [38])
- mogućnost izbora različitih tipova indeksa prilikom indeksiranja i pretraživanja dokumenata
- kontrola pristupa zasnovana na korisničkim ulogama (*role-based access control*)
- pretraživanje korišćenjem nekog od upitnih jezika
- transformacije dokumenata za potrebe prikazivanja, integracije sa drugim sistemima, evolucije strukture dokumenata i generisanja pogleda
- ažuriranje dokumenata, sa proverom referencijalnog integriteta u XML smislu (za IDREF atribute, *entity reference* polja i linkove ka dokumentima koji su u istoj bazi podataka) i transakcionim režimom rada.

U literaturi je objavljen i određeni broj radova na temu mapiranja modela XML dokumenata na relacioni model podataka. Mogu se uočiti dva pristupa rešavanju ovog problema:

1. mapiranje logičke strukture dokumenata (date DTD-om ili XML Schema dokumentom) na relacionu šemu i
2. mapiranje opšteg modela XML dokumenata na relacionu šemu.

Mapiranje logičke strukture dokumenata na relacionu šemu podrazumeva kreiranje zasebne relacione šeme za svaki tip dokumenta. Takvo mapiranje obično obuhvata kreiranje posebne relacije za svaki tip elementa u dokumentima. Ovakav metod prikazan je u radovima [51, 3, 144, 290]. Nešto složeniji metod mapiranja, koji uzima u obzir detaljnu analizu DTD-a dat je u [257].

Mapiranjem opšteg modela XML dokumenata na relacionu šemu dobija se jedinstvena relaciona šema koja se koristi za sve tipove dokumenata. Nekoliko radova bavi se ovim načinom mapiranja [325, 88, 34, 316].

Rad [316] kao situacije pogodne za usvajanje prvog pristupa navodi velike kolekcije dokumenata koji pripadaju malom broju različitih tipova, pri čemu su tipovi nepromenljivi tokom vremena. Sa druge strane, aplikacije koje koriste veći broj tipova dokumenata ili gde su tipovi dokumenata nepoznati ili promenljivi tokom vremena mogu bolje iskoristiti metode mapiranja zasnovane na drugom principu.

Sistemi za upravljanje relacionim bazama podataka su trenutno najrašireniji [149]. Istraživanja o mapiranju strukture XML dokumenata na druge tipove modela podataka (npr. objektno-orijentisani) nisu toliko brojna.

Pretraživanje XML dokumenata zasnovano je na nekom od predloženih upitnih jezika. U [316] prikazan je algoritam za generisanje SQL upita na osnovu datih XQL upita za XRel sistem. Rad [290] prikazuje arhitekturu X-Database sistema. Upiti u ovom sistemu formiraju se kao XML dokumenti strukturirani prema datoj XML Schema specifikaciji. Rad [302] prikazuje osobine SODA2 sistema, koji koristi proširenu XQL sintaksu za formiranje upita. Najznačajnije proširenje XQL jezika predstavlja mogućnost upotrebe regularnih izraza (*regular expressions*). Definisana je i odgovarajuća indeksna struktura za podršku ovakvim upitima. Radovi [250, 249] prikazuju ApproXQL, proširenje XQL-a koje omogućava upite po strukturi dokumenata koji pronalaze i parcijalna slaganja dokumenata sa upitom. Implementacija se zasniva na problemu neuređenog sadržavanja stabala analiziranog i ranije u okviru [139, 140], ali koristi novi algoritam koji je eksponencijalne složenosti samo u najgorem slučaju.

Kvalitativna novina koju donosi ApproXQL je mogućnost parcijalnog slaganja dokumenta sa upitom i rangiranje dokumenata prema stepenu slaganja sa upitom. Rangiranje pronađenih XML dokumenata obrađivano je u okviru konstrukcije još dva upitna jezika, XXL [280] i ELIXIR [48].

Orijentacija svih pomenutih upitnih jezika, osim ApproXQL, na egzaktno poređenje podataka posledica je polaznog pristupa njihovih autora koji dolaze iz oblasti baza podataka. Tretman upotrebe XML dokumenata u tom ambijentu obično se naziva *data-centric*: XML se uglavnom posmatra kao novi model podataka koji ima veću izražajnost u odnosu na relacioni model. Sa druge strane, tretiranje XML jezika kao standarda za formiranje strukturiranih tekstualnih dokumenata, u kojima se pretraživanje shvata kao pronalaženje informacija a ne kao pronalaženje podataka, naziva se *document-centric*. Autori *document-centric* orijentacije obično dolaze iz oblasti IR.

1.1.4 Relevance feedback u tekstualnim sistemima

Tehnike sa interakcijom sa korisnikom

U radu [167] data je sledeća definicija za *relevance feedback*: „RF je proces u kome se upit selektivno modifikuje da bi pronašao relevantnije dokumente u kolekciji nego njegova inicijalna verzija“. U tekstualnim sistemima modifikacija upita obuhvata dve operacije: 1) promenu težinskih koeficijenata dodeljenih elementima upita (*term reweighting*) i 2) dodavanje novih izraza u upit (*query rewriting*). Novi izrazi se pronalaze u okviru inicijalno pronađenih dokumenata za koje se smatra da su relevantni. Modifikacija težinskih koeficijenata u okviru klasičnog vektorskog modela prvi put je razmatrana u [232, 125]. Ovi rezultati i danas predstavljaju osnovu za implementaciju RF tehnike u vektorskom modelu.

Neka je, u okviru vektorskog modela, definisan sledeći skup veličina:

- D_r: skup relevantnih dokumenata kao podskup skupa pronađenih dokumenata, prema proceni korisnika,
- D_n: skup nerelevantnih dokumenata u okviru skupa pronađenih dokumenata,
- C_r: skup relevantnih dokumenata u okviru celokupne kolekcije,
- $|D_r|$, $|D_n|$, $|C_r|$: broj elemenata skupova D_r, D_n i C_r, tim redosledom i
- α, β, γ: konstante za podešavanje.

Idealan slučaj predstavlja situacija kada je skup C_r poznat unapred za dati upit q. Može se pokazati da je najbolji vektor upita koji razdvaja relevantne dokumente od nerelevantnih glasi [23]:

$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\forall \vec{d}_j \in C_r} \vec{d}_j - \frac{1}{N - |C_r|} \sum_{\forall \vec{d}_j \notin C_r} \vec{d}_j \tag{1.15}$$

Kako skup C_r nije poznat unapred, RF tehnika pokušava da inkrementalno dođe do idealnog rešenja polazeći od inicijalnog upita. Inkrementalne modifikacije zasnivaju se na podacima o relevantnim i nerelevantnim dokumentima među onima koji su upravo pronađeni. U literaturi su poznate tri klasične formule za modifikaciju vektora upita:

$$\vec{q}_m = \alpha\vec{q} + \frac{\beta}{|D_r|} \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|} \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j \tag{1.16}$$

$$\vec{q}_m = \alpha\vec{q} + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j \quad (1.17)$$

$$\vec{q}_m = \alpha\vec{q} + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \; max_{nonrelevant}(\vec{d}_j) \quad (1.18)$$

gde je $max_{nonrelevant}(\vec{d}_j)$ oznaka za najbolje rangirani nerelevantni dokument. U originalnim formulacijama [232, 125] vrednosti konstanti za podešavanje su $\alpha = \beta = \gamma = 1$. Danas se smatra da sve tri tehnike daju približno jednake rezultate [23]. Prikazane tehnike karakteriše jednostavnost (modifikacije vektora se računaju na osnovu skupa pronađenih dokumenata) i dobri rezultati koji su eksperimentalno ustanovljeni. Sa druge strane, ne postoji formulacija kriterijuma optimalnosti za iterativni proces.

Prikaz RF tehnike za probabilistički model dat je u odeljku 1.1.1. Kao mane ovog pristupa u [23] navodi se sledeće:

- težine izraza u dokumentima se ne uzimaju u obzir tokom *feedback* ciklusa
- težine dodeljene izrazima upita u prethodnim iteracijama se ne uzimaju u obzir
- upit se ne proširuje novim izrazima

Kao rezultat ovih osobina, RF tehnika u probabilističkom modelu je manje efikasna od tehnike razvijene za vektorski model. Rad [114] analizira eksperimente sa RF tehnikama za vektorski i probabilistički model i zaključuje da vektorski model pokazuje dobre rezultate sa većinom standardnih test kolekcija, dok probabilistički ima problema sa određenim kolekcijama. Varijante proširivanja upita novim izrazima u probabilističkom modelu predložene su u [305, 116]. Proširivanje Bulovskog modela probabilističkim tehnikama procene težina za RF istraživano je u više radova [223, 239, 61]. Rad [112] analizira upotrebu RF tehnike kod *inference network* modela. Radovi [112, 61] prikazuju RF tehnike za strukturirane upite ali pod strukturom podrazumevaju fraze kao sekvence reči u nestrukturiranim tekstualnim sistemima. Takav pojam strukture ne može se povezati sa modelima za strukturirane dokumente prikazanim u odeljku 1.1.2.

Relevance feedback tehnike u modelima sa strukturiranim dokumentima nisu razmatrane. Većina ovih modela nema mogućnost parcijalnog slaganja

dokumenata sa upitom, pa tako ni mogućnost rangiranja rezultata, odnosno upotrebe RF metoda za korekciju rezultata. Jezik ApproXQL [250, 249] omogućava parcijalno slaganje dokumenata sa upitom i ima mehanizam rangiranja rezultata ali upotreba RF tehnika nije razmatrana.

Rad [268] analizira *relevance feedback* sa stanovišta interakcije korisnika sa IR sistemom i uočava pet tipova interakcije:

- *Content relevance feedback* (CRF). Korisnik analizira odgovor sistema na inicijalni upit i procenjuje relevantnost pronađenih dokumenata. Ti podaci se koriste za novi *feedback* ciklus.
- *Term relevance feedback* (TRF). Korisnik analizira odgovor sistema na inicijalni upit i bira nove izraze iz pronađenih dokumenata za upit u narednom ciklusu.
- *Magnitude feedback* (MF). Korisnik analizira veličinu prikazanog skupa pogodaka i zahteva manji, veći ili skup nepromenjene veličine u narednom ciklusu.
- *Tactical review feedback* (TCR). Korisnik analizira izraze korišćene u prethodnim ciklusima pretrage radi određivanja dalje strategije formiranja upita.
- *Term review feedback* (TMR). Korisnik određuje dalju strategiju pretrage nakon pregleda izraza koji se nalaze u indeksu.

Efikasnost različitih tipova interakcije i njihov udeo u ukupnom broju interakcija analizirani su na uzorku upita koje su studenti redovnih i poslediplomskih studija postavljali Dialog sistemu. Kao osnovna mera efikasnosti interakcije posmatran je broj *feedback* petlji (*loops*). Autori zaključuju da najveći broj ciklusa bio MF i CRF tipa. TRF tip, iako detaljno obrađen u istraživanjima, nije bio značajno zastupljen. Poslednja dva tipa, usmereni ka formiranju dalje strategije pretrage, iako ređe zastupljeni od ostalih, uočeni su kao važni za uspešan ishod procesa pretrage u nekim situacijama.

Automatske tehnike

RF tehnike koje vrše reformulaciju upita ne mogu se nazivati automatskim jer koriste procenu korisnika o relevantnosti pronađenih dokumenata. Automatske RF tehnike podrazumevaju potpuno odsustvo učešća korisnika u toku reformulacije upita. Takve tehnike polaze od pretpostavke da je određeni

broj najbolje rangiranih dokumenata relevantan. Sistemi zasnovani na vektorskom modelu koriste klasične RF formule za vektorski model u kojima je deo koji se odnosi na nerelevantne dokumente zanemaren. Članak [242] analizira učinak automatske RF tehnike zasnovane na formuli 1.16 sa konstantom $\gamma = 0$. Iako je postignuti rezultat slabiji nego kada se uzmu u obzir i nerelevantni dokumenti, postignuto je poboljšanje u odnosu na pronalaženje dokumenata bez RF tehnike.

1.1.5 Klasteri dokumenata

Klaster analiza (*cluster analysis*) je naziv za veliku familiju metoda za klasifikaciju objekata. Objekti koji su predmet klasifikacije se, za potrebe klasterovanja, posmatraju kao tačke u odgovarajućem metričkom ili vektorskom prostoru. U [77] klasteri su opisani kao kontinualni regioni u prostoru koji imaju relativno veliku gustinu tačaka, razdvojeni od drugih takvih regiona regionima koji imaju relativno malu gustinu tačaka.

Algoritmi za klasterovanje koji se koriste u IR sistemima mogu se grubo klasifikovati u dve osnovne grupe [299, 120, 153]. Prvu grupu čine iterativni particioni algoritmi, koji formiraju klastere počevši od polaznog skupa tačaka i inicijalne particije prostora. Potom se iterativnim postupkom vrši optimizacija izabranog kriterijuma. Algoritmi ove grupe razlikuju se, pre svega, u pogledu izbora optimizacionog kriterijuma i načina određivanja inicijalnog rešenja. Najpoznatiji predstavnik ove grupe algoritama je *k-means* algoritam koji klastere reprezentuje svojim centroidima, tačkama prostora koje predstavljaju srednju vrednost koordinata svih tačaka članica klastera.

k-means formira particiju prostora sa n tačaka u k klastera na sledeći način:

1. Odredi se inicijalnih k klastera. Inicijalno rešenje najčešće se određuje kao k međusobno najudaljenijih tačaka ili prvih k tačaka u posmatranom skupu.
2. Sve tačke posmatranog skupa dodeljuju se najbližem klasteru, tj. klasteru čiji je centroid najbliži posmatranoj tački.
3. Vrši se ponovna kalkulacija položaja centroida klastera.
4. Korak 2 se ponavlja dok se kriterijum zaustavljanja ne zadovolji.

Osnovna vrlina *k-means* algoritma je linearna vremenska zavisnost od broja tačaka, što ga čini pogodnim za klasterovanje velikih skupova. Sa druge strane, ovaj algoritam pretpostavlja da klasteri imaju sferni oblik (u skladu sa korišćenom metrikom), što ne mora uvek biti slučaj. Pored toga, rezultujući broj klastera k mora biti određen unapred.

Hijerarhijski algoritmi [83, 120] imaju za cilj formiranje stabla čiji čvorovi su podskupovi polaznog skupa tačaka. Koren stabla je kompletan polazni skup. Listovi stabla su pojedinačne tačke skupa. Grupa divizionih algoritama formira hijerarhiju polazeći od korena stabla. Mnogo češći, aglomerativni algoritmi formiraju hijerarhiju polazeći od listova. Način rada ovih algoritama je sledeći:

1. Formira se početni skup klastera tako što se svaka tačka skupa nalazi u posebnom klasteru.
2. Dva međusobno najbliža klastera spajaju se u jedan klaster.
3. Korak 2 se ponavlja sve dok se kriterijum zaustavljanja ne zadovolji.

Varijante aglomerativnih hijerarhijskih algoritama međusobno se razlikuju po načinu izračunavanja međusobnog rastojanja klastera i kriterijumu zaustavljanja. Neke varijante, kao što je *complete-link* [153], omogućavaju formiranje klastera koji nisu sfernog oblika. Međutim, vremenska zavisnost ovih algoritama je $O(n^2)$ ili $O(n^3)$. Kao kriterijum zaustavljanja postupka uzima se ili maksimalan broj klastera ili minimalno rastojanje između konačnih klastera. Ako je minimalno rastojanje između klastera konstantno tokom svih sesija pronalaženja, gustina klastera teži konstanti [153].

Poseban problem svim algoritmima za formiranje klastera predstavljaju usamljene tačke pojedinim delovima posmatranog prostora (*outliers*). Formiranje klastera nad skupovima sa velikim brojem ovakvih tačaka je složen problem [21]. Pored dve prikazane grupe algoritama, za klasterovanje dokumenata koriste se i Kohonenove samoorganizujuće mape (*self-organizing maps*, SOM) [118] i algoritmi zasnovani na teoriji grafova [119].

Klasterovanje primenjeno na kolekcije tekstualnih dokumenata sprovodi se na dva načina: (1) klasterovanje cele kolekcije unapred i (2) klasterovanje pronađenih dokumenata u cilju olakšavanja pregleda rezultata (*browsing*). Pored toga, moguće je vršiti i klasterovanje upita, radi poboljšanja performansi klasičnih metoda pronalaženja [294], ili radi formiranja FAQ (*frequently asked questions*) mapa i reformulacije upita [299].

Veća efikasnost metoda klasterovanja u odnosu na klasične metode rangiranja je retko potvrđivana, i to samo u situacijama za specifične kolekcije dokumenata ili u situacijama kada se klasterovanje koristi za pregled dokumenata u kombinaciji sa klasičnim rangiranjem [120]. U [293] upotreba klasterovanja ispitana je na nekoliko različitih kolekcija, ali nije ustanovljen napredak u odnosu na klasične metode rangiranja.

1.2 Pronalaženje slika

Problem pronalaženja slika se u literaturi posmatra, pre svega, kao problem pronalaženja rasterskih slika. Tretman vektorskih slika može se smatrati delom prethodnog problema, u oblastima pronalaženja slika po prostornim odnosima elemenata slike (v. odeljak 1.2.2) i pronalaženja po apstraktnom sadržaju (v. odeljak 1.2.3).

Ovaj odeljak razmatra probleme pronalaženja rasterskih slika. U knjizi [258], prilikom analize problema digitalizacije slika, data je sledeća definicija analogne i digitalne slike:

Definicija 1.7 *Analogna slika predstavljena je funkcijom* $f(x, y) : R^2 \rightarrow R$ *gde su* x *i* y *prostorni parametri, a vrednost funkcije predstavlja intenzitet slike u tački* (x, y). *Digitalna slika predstavljena je funkcijom* $I(r, c)$ *gde* $r \in \{0, \ldots, m\}$ *i* $c \in \{0, \ldots, n\}$ *predstavljaju kvantizacije prostornih parametara, a vrednost funkcije* I *je kvantizacija vrednosti funkcije* f.

Knjiga [258] detaljno se bavi karakteristikama digitalnih slika i metodama za njihovu transformaciju. U ovom odeljku biće reči o onim osobinama slika koje su od interesa za problem pronalaženja slika u okviru date kolekcije.

U okviru pregleda istraživanja u oblasti pronalaženja informacija u multimedijalnim bazama podataka [317] navedena su sledeća tri razloga koja konvencionalne sisteme za upravljanje bazama podataka (relacione i objektno-orijentisane) čine neprimerenim u slučaju multimedijalnih podataka:

1. Nedostatak mehanizama za efikasno rukovanje prostorno-vremenskim odnosima u okviru multimedijalnih objekata.
2. Prepoznavanje odnosno interpretacija sadržaja multimedijalnih objekata podrazumeva posedovanje baze znanja teško se uklapa u koncepte konvencionalnih sistema.

3. Za multimedijalne baze podataka izražavanje upita pomoću tekstualnih odnosno numeričkih izraza nije uvek zadovoljavajuće. Upiti po sličnosti sa uzorkom (*query by example*) zahtevaju formu reprezentacije upita koja je sličnija tipu podataka koji se pretražuju.

Rad [204] klasifikuje pretraživanje slika po sadržaju po tri nivoa apstrakcije:

- *Osnovni podaci (raw data).* Na najnižem nivou objekti se posmatraju kao nizovi piksela. Poređenje objekata ili pojedinih regiona odvija se piksel-po-piksel korišćenjem mera sličnosti kao što su koeficijent korelacije ili euklidsko rastojanje.
- *Osobine (features).* Izvedene karakteristike slike ili regiona slike kao što su luminansa, tekstura, histogram boja ili deskriptor oblika koriste se prilikom poređenja slika.
- *Semantika (semantic).* Izvedene karakteristike su grupisane u objekte koji predstavljaju značenje dato slikom.

Najbrojnije reference su iz oblasti pretraživanja pomoću osobina (*features*). Zajedničko svim ovim radovima je težnja da se perceptivne osobine slike predstave vektorom realnih brojeva što manje dimenzije kako bi se takvi vektori mogli porediti u odgovarajućem prostoru sa izabranom metrikom. Postoji i grupa radova koja kao meru sličnosti dve slike izražava potrebnim brojem elementarnih operacija za modifikaciju interne reprezentacije slike kako bi se dve interne reprezentacije izjednačile.

1.2.1 Pronalaženje po osobinama slike

Članci [106, 107] sistematizuju probleme u oblasti pretraživanja vizuelnih medija. U [107] uočene su četiri kategorije koncepata vezanih za vizuelne objekte: osobine (*features*), prostor osobina (*feature space*), grupe osobina (*feature groups*) i prostor slika (*image space*).

Osobina (*feature*) je izvedeni atribut dobijen transformacijom originalnog objekta pomoću algoritma za analizu slike. Osobina se tipično predstavlja nizom brojeva i često se naziva vektor osobina (*feature vector*). Operacije koje se vrše nad ovakvim vektorima su sledeće:

Projekcija. Projekcija kreira novi vektor manje dimenzije od originalnog izbacivanjem nekih od dimenzija vektora. Time se određene karakteristike objekta svesno zanemaruju.

Primena funkcije. Vektor osobina se transformiše primenom odgovarajuće funkcije. Domen ove funkcije čini skup vektora osobina, a rezultat je novi vektor iste dimenzije. Tipičan primer je primena filtera za boje na histogram boja za izdvajanje određenih boja u slici.

Rastojanje. Rastojanje je mera sličnosti dva vektora osobina. Ovo je najvažnija operacija sa vektorima u mnogim sistemima jer predstavlja direktnu meru sličnosti dva objekta.

Objekti predstavljeni vektorima osobina predstavljaju tačke u prostoru osobina (*feature space*). Tipične operacije koje se vrše u ovakvom prostoru su sledeće:

Traženje granice. Za dat skup tačaka u prostoru, operacija pronalazi minimalni hiperpoliedar koji obuhvata sve tačke.

Izbor putem prostornih ograničenja. Ova operacija predstavlja upit za pronalaženje svih tačaka koje se nalaze unutar (ili izvan) date hiperkocke. Način zadavanja regiona može biti različit. Na primer, moguće je definisati region putem njegovih numeričkih parametara ili crtanjem regiona pomoću duži i krivih u prostoru u okviru odgovarajuće aplikacije.

Izbor ograničenjem rastojanja. Specijalan oblik prethodnog upita, gde korisnik bira jednu tačku u prostoru i traži sve objekte čije se tačke nalaze na rastojanju od izabrane tačke ne većem od zadatog.

k **najbližih suseda.** Ovo je najšire rasprostranjen tip upita. Za datu tačku prostora (koja je najčešće definisana uzorkom koji predstavlja upit) operacija vraća k najbližih suseda, rangirajući ih u rastućem poretku rastojanja od date tačke.

Particija prostora. Pronalaženje regiona u prostoru u kojima su tačke grupisane (tj. određivanje klastera).

Dodela imena. Određenom delu prostora osobina dodeljuje se ime. Takvo ime se potom može koristiti prilikom formiranja upita da označi dati deo prostora.

Agregatne operacije. Operacije kao što su srednja vrednost, standardno odstupanje, prečnik klastera.

Grupe osobina (*feature groups*), dobijene grupisanjem elementarnih osobina, mogu imati veću izražajnost od pojedinačnih elemenata. U [89] govori se

o kombinovanju detekcije boje ljudske kože i selekciji putem prostornih ograničenja (posmatranjem svih zatvorenih objekata u slici) za otkrivanje lica ili ljudskih figura u rasterskim slikama.

U [258] definisane su četiri osnovne klase mera sličnosti: (1) sličnost boja, (2) sličnost tekstura, (3) sličnost oblika i (4) sličnost međusobnih prostornih odnosa objekata.

Sličnost boja

Jedna od najčešće upotrebljavanih mera za sličnost boja koristi histograme boja. Histogram boja slike je diskretna funkcija koja vraća broj piksela date boje u slici. Broj boja koje histogram sadrži može biti manji od broja stvarno prisutnih boja; prilikom izračunavanja histograma vršiće se kvantizacija. Kvantizacija se obično vrši pomoću klasterovanja sličnih boja [80]. Mera sličnosti odnosno rastojanje dva histograma se tada može definisati kao [87, 258]:

$$d_{hist}(I,Q) = (h(I) - h(Q))^T \ A \ (h(I) - h(Q)) \tag{1.19}$$

gde su $h(I)$ i $h(Q)$ histogrami dvaju poređenih slika, a A je matrica međusobne sličnosti boja. Vrednosti u matrici su u rasponu od 0 (za boje koje nisu slične) do 1 (za jednake boje). Druga varijanta uzima u obzir i prostorni raspored boja na slici tako što se slika podeli na matricu segmenata pa se histogram formira za svaki segment posebno. Tada je ukupno rastojanje jednako sumi svih pojedinačnih rastojanja:

$$d_{grid}(I,Q) = \sum_g d_{hist}(I_g, Q_g) \tag{1.20}$$

Značajan broj radova bavi se analizom upotrebe pojedinih tipova metrike. U [276, 93] koriste se L_1 metrike, [198, 131] koriste L_2 metrike, dok [272] upotrebljava L_∞. Problemi efikasnog indeksiranja histograma boja razmatrani su u [111, 205].

Sličnost tekstura

Analiza teksture slika je razvijena oblast istraživanja, prvenstveno u okviru disciplina računarske vizuelne percepcije (*computer vision*) i digitalne obrade signala. U literaturi je prikazan veliki broj metoda za analizu teksture slike. U

[258] dat je pregled klasičnih metoda. Prikazani su (1) metode zasnovane na detekciji ivica i određivanju gustine i usmerenosti ivica, (2) L_1-poređenje histograma formiranih pomoću lokalne binarne particije, (3) matrice zajedničkog pojavljivanja (*co-occurence matrices*), (4) Laws-ov metod određivanja energije teksture i (5) autokorelacija i analiza spektra snage.

Jedna od efikasnih reprezentacija teksture, prikazana u [277], predstavlja teksturu pomoću tri parametra: krupnoće (*coarseness*), kontrasta (*contrast*) i usmerenosti (*directionality*) i koristi se u okviru QBIC sistema [80].

Novija istraživanja često se zasnivaju na Wold-ovoj dekompoziciji slučajnih procesa na više međusobno nezavisnih procesa. Eksperimentalna analiza ljudske percepcije teksture u [225] identifikovala je (1) ponavljanje (*repetitiveness*), usmerenost (*directionality*) i (3) granularnost i složenost (*granularity and complexity*) kao tri najvažnije dimenzije percepcije teksture. Veza između rezultata Wold-ove dekompozicije i ovih karakteristika opisana je u [208]. Analiza teksture na ovaj način rezultuje kompaktnom reprezentacijom koja očuvava perceptivne atribute [269]. Radovi [214, 92, 160] koriste neku od varijanti Wold-ove dekompozicije kao model analize teksture.

Druge metode za analizu teksture uključuju upotrebu Gaborovih filtera [168, 173, 308], proširenje Laws-ovog metoda određivanja energije teksture za rad na paralelnim računarskim arhitekturama [319], SPCA model [213] i MRSAR model [174].

Sličnost oblika

Jedan od jednostavnih načina za reprezentaciju oblika u slici je histogram oblika (*shape histogram*) [258], koji je rezultat projekcije oblika na obe ose i prebrojavanja piksela koji pripadaju datom obliku za svaku diskretnu vrednost na osama. Ovakav koncept obezbeđuje invarijantnost reprezentacije na veličinu i položaj objekta, ali ne i na rotaciju. Varijanta ovog koncepta koja uključuje i izračunavanje uglova osa elipse koja najbolje obuhvata dati oblik poseduje i invarijantnost na rotaciju. Poređenje histograma se tipično vrši L_p metrikama. Rad [131] prikazuje upotrebu histograma za opisivanje poligonalne aproksimacije oblika.

Poređenje granica (*boundary matching*) [258] zasniva se na izračunavanju koeficijenata Furijeovog reda kojim se opisuje granica datog oblika. Kako je granica oblika data nizom piksela, koriste se aproksimativne formule za određi-

vanje koeficijenata. Vektori koeficijenata se tada porede L_p metrikom. Druga varijanta se zasniva na aproksimaciji granica oblika poligonom, što donekle smanjuje računsku složenost metode.

Metoda za poređenje skica (*sketch matching*) [258] svodi uzorak i slike u kolekciji na posebnu formu (tzv. *abstract image*) pomoću niza transformacija (skaliranje na jednaku veličinu, uklanjanje šuma, detekcija ivica i sl.) i potom poredi dobijene apstraktne reprezentacije. Sličnost se meri korelacijom pojedinih segmenata slike raspoređenih u pravougaonu rešetku.

Rad [211] prikazuje sistem koji omogućava pronalaženje objekata po obliku. Oblik se karakteriše pomoću tri komponente: veličine (izračunavanjem površine koju oblik pokriva), zaokrugljenosti (izračunavanjem odnosa najmanjeg i najvećeg drugog momenta) i orijentacije (ugao između horizontalne ose i ose najmanjeg drugog momenta). Mera sličnosti je L_p metrika.

Sistem QBIC [80] koristi površinu, kružnost, ekscentričnost, dominantnu osu orijentacije i skup invarijantnih algebarskih momenata (vektor sa ukupno 20 elemenata). Kao rastojanje koristi se L_p metrika sa težinama, gde težine označavaju važnost pojedinih karakteristika prilikom poređenja. Kako je korišćena metrika kvadratne složenosti, u slučajevima velikog broja dimenzija njeno izračunavanje može biti nedovoljno efikasno. Kao korekciju ovog problema, autori predlažu neku od metoda za smanjivanje dimenzionalnosti problema, pri čemu se ne dozvoljava pojava odbacivanja objekata koji bi zadovoljili kriterijum sličnosti sa inicijalnom metrikom (tzv. *false dismissal* problem). Kao metode koje obezbeđuju ovu funkcionalnost navedene su Karhunen-Loève transformacija, diskretna kosinusna, Furijeova ili *wavelet* transformacija [218].

Članak [253] definiše skup osnovnih tipova oblika i skup transformacija za definisanje izvedenih oblika. Za sve oblike u kolekciji slika određuju se nizovi transformacija kojim se dati oblik može dobiti od polaznih tipova. Poređenje oblika svodi se tada na poređenje nizova transformacija odgovarajuće definisanom metrikom. U radu [288] opisana je metoda koja kombinuje upotrebu generisanih vektora osobina za grubo pronalaženje kandidata za pogotke i fino poređenje preostalih objekata pomoću analize nizova transformacija.

Photobook sistem [208] se oslanja na varijantu metoda konačnih elemenata [252] za generisanje matrice koja opisuje međusobni odnos ključnih tačaka ob-

lika. Sopstveni vektor ove matrice koristi se kao reprezent oblika prilikom pretraživanja. Značenje pojedinih elemenata ovog vektora određeno je empirijski.

Radovi [190, 189] analiziraju upotebu CSS (*curvature scale space*) reprezentacije oblika za pronalaženje slika morskih životinja. Istaknuta je velika sličnost rezultata pronalaženja sa procenama koje su vršili korisnici. Sistem je pokazao invarijantnost pretrage na veličinu i prostornu orijentaciju objekata ali je u analizi ograničen na slike na kojima se nalazi tačno jedan objekat.

Ostale mere

Značajna grana istraživanja reprezentacije osobina slike koristi *wavelet* objekte. *Wavelet* objekti se koriste za konstrukciju efikasnog indeksiranja za pretraživanje po boji [172, 8]. Rad [255] koristi kombinaciju *wavelet*-a i Kohonenovih neuronskih mreža za pronalaženje slika.

Grupa radova [208, 227, 142] bavi se problemom određivanja vektora osobina koji najbolje reprezentuje vizuelnu pojavu slike (*appearance*), pri čemu se kvalitet reprezentacije utvrđuje empirijski. Za tu namenu formirani su različiti matematički modeli. Upotreba Gausovih filtera za generisanje vektora osobina prikazana je u [227]. Upotreba Karhunen-Lòeve transformacije za redukovanje dimenzionalnosti prostora u kome se predstavljaju pojavni atributi slike data je u [208]. Članak [142] koristi sličan metod za potrebe prepoznavanja ljudskih lica.

U članku [328] prikazana je ideja formiranja rečnika „ključnih blokova“, analogno formiranju rečnika ključnih reči za kolekcije tekstualnih dokumenata. Pretraživanje po „ključnim blokovima“ se, kroz prikazanu analizu, pokazalo kao efikasnije od klasičnog pretraživanja po osobinama.

1.2.2 Pronalaženje po prostornim odnosima elemenata slike

Istraživanja u oblasti pronalaženja slika po prostornim odnosima sadržanih objekata koncentrišu se, pre svega, na određivanje reprezentacija prostornih odnosa koje obezbeđuju efikasnu pretragu. Reprezentacija samih objekata je tipično u drugom planu – podrazumeva se da je identifikacija objekata i reprezentacija njihove semantike zadatak za eksperta. Članak [104] definiše pojam simboličke slike (*symbolic image*) kao rezultat procesa identifikacije objekata

u okviru tzv. fizičke slike i dodeljivanja naziva objektima. Uz same objekte evidentira se i položaj njihovih centroida.

Rani radovi iz ove oblasti koriste 2D stringove [46]. Svakom objektu slike dodeljuje se jedinstveno ime. Relativni međusobni položaj između objekata reprezentuje se pomoću dva jednodimenzionalna stringa. Pretraživanje se svodi na pronalaženje svih 2D stringova koji sadrže 2D string upita kao svoj podstring. Strukture namenjene indeksiranju 2D stringova za velike kolekcije slika prikazane su u [210]. 2D C-stringovi [150] bave se situacijama preklapanja objekata složenih oblika.

Članak [151] bavi se delimičnim poređenjem prostornih odnosa i definiše odgovarajuću meru sličnosti. Simboličke slike i upit se predstavljaju 2D stringovima. Slika koja poseduje najveću podsliku datog upita smatra se za najrelevantniju za taj upit. Pronalaženje slika se, prema ovoj ideji, svodi na pronalaženje najdužih sekvenci zajedničkih za 2D string upita i skup 2D stringova slika u kolekciji. Definisane su tri vrste mera sličnosti. Mera nazvana Type-2 je najstrožija: slika iz kolekcije zadovoljava upit u slučaju da postoji njena podslika koja sadrži sve objekte iz upita, ali i nijedan drugi objekat, u istom redosledu duž obe ose. Komplement podslike ne sadrži objekte čije se projekcije na obe ose nalaze između objekata iz podslike. Mera Type-1 dozvoljava da komplement podslike sadrži objekte čija se projekcija nalazi između projekcija objekata podslike. Mera Type-0 dozvoljava da se prostorni raspored objekata u upitu projektuje na isti položaj u okviru neke od osa.

U članku [45] simbolička slika se prikazuje kao skup uređenih trojki (σ_i, σ_j, r_{ij}) gde su σ_i i σ_j objekti sadržani u slici, a r_{ij} izražava položaj objekta σ_i u odnosu na σ_j pomoću osam oznaka tipa *sever*, *severozapad*, itd. i oznake *na istom mestu*. Zahteva se da za svaku uređenu trojku važi $\sigma_i < \sigma_j$ u leksičkom smislu. Problem pretraživanja se posmatra kao problem poređenja skupa uređenih trojki koje pripadaju upitu i skupova uređenih trojki koje pripadaju slikama u kolekciji. Prema analizi sprovedenoj u [104], prikazani koncept omogućava efikasnu implementaciju ali ne poseduje mogućnost delimičnog slaganja slike sa upitom.

Reprezentacija simboličke slike u [104] koristi grafove sa težinama dodeljenim granama. Čvorovi grafa predstavljaju objekte na slici, dok težine grana predstavljaju ugao pod kojim jedan objekat „vidi“ drugog u odnosu na referentnu osu. Prikazani algoritam za izračunavanje sličnosti dva grafa uzima u

obzir težine dodeljene granama grafa i ima kvadratnu vremensku složenost. Karakteristika ovog modela je i invarijantnost rezultata pronalaženja na rotaciju, translaciju i skaliranje elemenata slike.

Članak [211] prikazuje upotrebu ARG grafova (*attributed relational graphs*) za reprezentaciju prostornih odnosa. Pored toga, prikazani model reprezentuje i objekte pomoću tri parametra: veličine, zaokrugljenosti i orijentacije što je opisano u prethodnom odeljku. Objekti slike su predstavljeni čvorovima grafa dok grane grafa izražavaju međusobne prostorne odnose pomoću dva dodeljena parametra: rastojanja i ugla pod kojim jedan objekat „vidi“ drugog. Rad prikazuje metod za mapiranje ovakvih grafova na višedimenzionalni prostor koji se pretražuje pomoću posebno generisanih indeksnih struktura zasnovanih na R-stablima [108]. Kao mera sličnosti koristi se L_p metrika nad vektorima čiji su elementi atributi čvorova i grana grafa.

Nešto drugačiji koncept pretrage opisan je u [204]. Rad definiše sopstvenu reprezentaciju objekata zasnovanu na projekciji dimenzija objekata na koordinatne ose i predstavi dobijenih intervala pomoću nizova bitova. Za datu reprezentaciju definisana je i odgovarajuća metrika. Umesto klasičnih algoritama za pretraživanje celokupne kolekcije (koji su eksponencijalne složenosti), rad analizira upotrebu genetskih i *hill-climbing* algoritama za pronalaženje suboptimalnih rešenja u slučajevima kada je vreme pronalaženja ograničeno.

Sistem VisualSEEk [265] omogućava kombinovanje pretrage slika po globalnim karakteristikama i po prostornim ograničenjima. Sistem je sposoban i za automatsku ekstrakciju regiona slike. Prostorne karakteristike pronađenih objekata opisuju se pomoću njihovih centroida i minimalnih pravougaonika koji ih obuhvataju. Ovako definisani prostorni položaji regiona mogu se iskoristiti za nekoliko tipova upita, pri čemu sistem koristi dva tipa indeksa: položaji centroida regiona smeštaju se u *spatial quad-trees* [247], dok se obuhvatajući pravougaonici smeštaju u R-stabla [108].

1.2.3 Pronalaženje po apstraktnom sadržaju

U prethodnim odeljcima razmatrani su rezultati postignuti u oblasti ekstrakcije osobina slika niskog nivoa (*low level features*) kao što su boja, tekstura, oblik i prostorni raspored elemenata slike. Apstraktni sadržaj slike definisan je u [178] kao „rezultat čovekovog procesa interpretacije koji proizvodi mentalnu rekonstrukciju scene prikazane slikom“. Automatska ekstrakcija ap-

straktnog sadržaja iz slika je izuzetno težak problem [16]. Sistemi za pronalaženje slika po apstraktnom sadržaju mogu se podeliti na one koji podrazumevaju učešće čoveka u procesu indeksiranja, odnosno zahtevaju ručno proizvedene metapodatke i sisteme za automatsku ekstrakciju apstraktnog sadržaja.

Problem reprezentacije ručno generisanih metapodataka vezanih za slike rešavan je na različite načine. Rad [178] definiše model pronalaženja zasnovan na predikatskoj logici kao sredstvu za reprezentaciju sadržaja. Model je zamišljen tako da može da obuhvati i pretraživanje po osobinama slike niskog nivoa.

Značajan broj istraživanja bavi se pretraživanjem slika po pridruženim tekstualnim opisima (*captions*). Članak [105] prikazuje sistem za pretraživanje slika putem pretraživanja pridruženih opisa i drugih podataka. Upiti se od strane korisnika formulišu kao izrazi na engleskom jeziku. Na osnovu toga generiše se tzv. logička forma upita zasnovana na modelu iz [109]. Pretraživanje se sastoji iz dve faze: prva faza koristi klasične indeksne strukture za eliminaciju slika koje nemaju odgovarajuće izraze u pridruženim opisima, a druga faza preostale slike pretražuje putem poređenja logičkih formi upita. Karakteristično za ovaj sistem je i postojanje hijerarhije tipova koja obuhvata imenice i glagole i klasifikuje pojmove u hijerarhijsku strukturu sličnu bibliotečkom UDK sistemu (prikazuje odnos specijalizacije/generalizacije).

Chabot sistem [202] koristi podatke o boji i tekstualne opise za pretraživanje slika. Definisan je pojam „upita po konceptu“ gde je koncept (npr. zalazak sunca) rezultat analize slike po boji. Reprezentacija znanja zasnovana na frejmovima se određuje unapred za svaku sliku i smešta se u okviru definisanog relacionog modela. Rad [17] razmatra probleme prilikom skladištenja i pretraživanja slika u relacionim bazama podataka i definiše algebru zasnovanu na izračunavanju sličnosti između objekata (umesto egzaktnog poređenja) prilagođenu rukovanju slikama.

Sistem SCORE [15, 14] koristi koncepte ER dijagrama (entitete, atribute i veze) za reprezentaciju sadržaja slike. Definisane su mere sličnosti za poređenje entiteta, atrubuta i veza kao i algoritam za poređenje celokupnog sadržaja slika. U radu [16] analizirana je upotreba WordNet sistema [186] za proširivanje upita i pridruženih opisa novim pojmovima. WordNet je proizvod istraživačkog projekta koji predstavlja pokušaj da se modeluje leksičko znanje čoveka (za engleski jezik).

Rad [264] prikazuje pristup koji u osnovi ima prethodno određivanje sličnosti između izraza bazirano na njihovoj semantici (uz pomoć WordNet sistema) i korišćenje tih podataka u izračunavanju sličnosti između dokumenta i upita. Primena ovog koncepta testirana je na sistemu za pretraživanje pridruženih tekstualnih opisa slikama.

Rezultati u automatskoj ekstrakciji apstraktnog sadržaja postignuti su u pojedinim usko specijalizovanim oblastima. Najbrojniji su radovi koji se bave otkrivanjem [296] i prepoznavanjem ljudskog lica [246, 29], određivanjem doba starosti osobe na slici [146], određivanjem pola osobe [311] i izraza lica [33, 177, 73, 141]. Druga posebno razvijena oblast je analiza medicinskih snimaka (NMR snimci [211], histološki nalazi [278]).

Automatska ekstrakcija sadržaja često se zasniva na korišćenju rezultata procesa ekstrakcije osobina niskog nivoa, tj. vektorima osobina. Dobijeni vektor osobina se prosleđuje specijalizovanom modulu za semantičku analizu, koji na osnovu sadržaja dobijenog vektora i postojeće baze znanja (nastale treniranjem na poznatom skupu) generiše semantički opis sadržaja slike. Pregled nekoliko pristupa problemu automatske ekstrakcije sadržaja dat je u [42]. Rad [278] prikazuje karakteristike ovakvog sistema specijalizovanog za pretraživanje histoloških nalaza u medicini. Kao indeks za pretragu koriste se i vektori osobina i reprezentacije semantike. Sistem poseduje i generator pridruženih tekstualnih opisa koji se takođe mogu koristiti u pronalaženju slika.

Rad [36] predstavlja sistem za klasifikaciju slika po dva kriterijuma: prepoznavanje prirodnih objekata i objekata proizvedenih od strane čoveka i prepoznavanje scena koje su snimljene u zatvorenom i otvorenom prostoru. Za potrebe analize slika se deli na segmente i to na više hijerarhijskih nivoa. Svaki segment se karakteriše svojim vektorom osobina na osnovu analize boje i teksture. Na osnovu tog vektora vrši se procena oblika funkcije raspodele verovatnoće (podrazumeva se normalna raspodela) da posmatrani fragment pripada nekoj od klasa. Tako procenjena raspodela koristi se za klasifikaciju segmenata slike. Segmenti višeg nivoa se klasifikuju na osnovu podataka o verovatnoći za segmente nižeg nivoa. Prepoznavanje otvorenog odnosno zatvorenog prostora vrši se na isti način, s tom razlikom da se slika ne segmentira nego se posmatra kao jedna celina. Prema datoj analizi, upotreba ovog sistema uz dodatno evidentiranje metapodataka vezanih za sadržaj slike (npr. datum fotografisanja) omogućava efikasno pretraživanje kolekcije slika.

1.2.4 Relevance feedback u pronalaženju slika

Upotreba RF tehnike za reformulaciju upita za pronalaženje slika tema je većeg broja radova. Klasičan pristup ovom problemu podrazumeva korišćenje vektora osobina, upotrebu L_p metrike kao mere sličnosti i pretragu tipa k najbližih suseda. Reformulacija upita svodi se na modifikaciju prostora osobina [183, 226] ili modifikaciju težinskih koeficijenata metrike vezanih za pojedine elemente vektora [129, 235, 236, 82]. Izračunavanje vrednosti metrike u slučaju pretraživanja velikih kolekcija ili velikog broja dimenzija prostora osobina je računski zahtevan zadatak. Jedno rešenje ovog problema je sukcesivna upotreba više metrika, pri čemu je prva metrika – definisana tako da bude računski manje složena, npr. L_∞ – namenjena za „grubu“ eliminaciju tačaka koje ne ulaze u skup k najbližih suseda, dok sledeća metrika – tipično L_2 – vrši precizan izbor suseda. Situacija u kojoj prva korišćena metrika odbacuje neku tačku koju bi druga metrika odredila kao deo skupa rezultata naziva se *false dismissal*. Izbor metrika koji garantuje nepostojanje *false dismissal* situacija određen je teoremom o donjoj granici funkcije (*lower bounding*) [80].

Prethodni primer optimizacije izračunavanja upita koristi se i u sistemima koji nemaju implementirane RF tehnike. Nešto drugačiji koncept, prikazan u [307], optimizuje naredne iteracije pretrage za k najbližih suseda particijom prostora na hiperkocke koje sadrže određeni broj tačaka. Radovi [273, 78, 44, 40] koriste PCA analizu [69] za redukciju broja dimenzija prostora osobina na osnovu analize datih tačaka u prostoru. Ovako redukovani prostor koristi se tokom RF iteracija u [273]. Korišćenje stabala odlučivanja za particiju prostora osobina radi smanjenja skupa ispitivanih tačaka opisano je u [169].

Problem definisanja RF tehnike ima i određeni broj rešenja koja potiču iz oblasti mašinskog učenja (*machine learning*). Radovi [282, 47, 122] koriste SVM (*support vector machines*) kao mehanizam učenja i klasifikacije tačaka u prostoru osobina na dve klase – relevantne i nerelevantne. U [327] data je uporedna analiza performansi metoda zasnovanih na *discriminating transformations*. Formiranje klastera tačaka rešavano je upotrebom Kohonenovog LVQ algoritma i neuronskih mreža [304] i pomoću samo-organizujućih mapa (SOM) [147]. Konstrukcija bajesovske funkcije za klasifikaciju tokom RF iteracija prikazana je u [274].

Rad [165] predstavlja, za sada, jedini pokušaj definisanja RF tehnike za pretraživanje po semantici slike, umesto po vektorima osobina. Kolekcija koja

se pretražuje sastoji se od slika i ključnih reči koje služe kao opisi semantike. Ključne reči i slike povezane su težinskim vezama. Težina izražava meru u kojoj ključna reč odgovara semantici slike. Upiti se postavljaju navođenjem ključnih reči; odgovor korisnika na rezultate upita je procena da li slika pripada skupu relevantnih slika ili ne. Prikazana RF tehnika bavi se modifikacijom težina na osnovu analize odgovora korisnika.

1.3 Pronalaženje video zapisa

Video zapisi predstavljaju digitalizovani oblik video signala. Signal može da potiče od kamere kojom upravlja čovek ili sintetički generisanih scena. Za razliku od statičnih slika, video zapisi poseduju i vremensku dimenziju, što stvara novo okruženje u kome se rešava problem pronalaženja video zapisa. Digitalni video zapis posmatra se kao niz statičnih slika (frejmova) koje se tokom prikazivanja smenjuju u odgovarajućim vremenskim trenucima.

1.3.1 Struktura video zapisa

Tradicionalna filmska teorija definiše tri nivoa hijerarhije u segmentaciji video zapisa: kadar, scena i sekvenca. Kadar (*shot*) je interval video zapisa snimljen u jednoj neprekidnoj operaciji. Scena (*scene*) se obično definiše kao potpun kontinualan niz događaja na jednoj lokaciji. Sekvenca (*sequence*) predstavlja grupu scena povezanih zajedničkom radnjom.

Analiza strukture digitalnih video zapisa do sada je tekla uglavnom u pravcu automatskog otkrivanja kadrova. Granica između dva kadra u literaturi se obično naziva *scene change*, iako se misli na *shot change*. Nakon procesa montaže, kadrovi mogu biti razdvojeni rezovima (*cuts*) ili posebnim prelaznim segmentima, npr. *fade* i *dissolve*. Sa stanovišta detekcije kadrova, postoje dva tipa granice: *abrupt* i *gradual* [312].

1.3.2 Detekcija, reprezentacija i pretraživanje kadrova

Detekcija granice kadrova u velikoj meri zavisi od formata korišćenog za kodiranje video zapisa. U analizi [1] tehnike su klasifikovane na one koje operišu nad nekompresovanim i kompresovanim video zapisima.

Tehnike zasnovane na analizi vizuelnog sadržaja sukcesivnih frejmova tipično su zasnovane na korišćenju određene metrike za određivanje sličnosti između susednih frejmova. Analiza više ovakvih tehnika [192] kao najuspešniju navodi jednu varijantu poređenja histograma boja. Slične metode prikazane su i u [323, 259]. U [10] za ovu namenu koriste se neuronske mreže, dok [124] koristi klasterizaciju frejmova. Algoritam koji omogućava detekciju i *abrupt* i *gradual* prelaza prikazan je u [262]. U radu [72] prikazan je proces izdvajanja ključnih frejmova nakon koga se dobijeni skup filtrira radi dalje eliminacije manje značajnih frejmova.

Analiza kompresovanog video zapisa oslanja se prevashodno na MPEG format [191]. Veći broj radova bavi se detekcijom promene kadra u ovakvim video zapisima (npr. [184, 161, 256]). Određeni broj radova povezuje informacije sadržane u video zapisu sa njegovim audio kanalima za precizniju segmentaciju zapisa [156, 117].

Nakon analize video zapisa i identifikacije kadrova (proces u literaturi često nazivan parsiranje videa, *video parsing*) potrebno je formirati određene reprezentacije kadrova radi kasnijeg pretraživanja. Dva osnovna pristupa ovom problemu su izdvajanje ključnih frejmova (*key-frame extraction*) i slaganje mozaika (*mosaicking*). Slaganje mozaika (npr. [279, 248]) zasniva se na superponiranju susednih frejmova kako bi se dobila reprezentacija koja obuhvata sve frejmove kadra.

Rezultat izdvajanja ključnih frejmova (npr. [324, 9]) su statične slike koje reprezentuju sadržaj celog kadra. Ovakve slike se potom pretražuju pomoću osnovnih karakteristika slika – boje, teksture i oblika – na način kako je to opisano u odeljku 1.2. Sistem [18] predstavlja primer ovakvog pristupa. Kako ključni frejmovi ne mogu dovoljno dobro da predstave vremenske karakteristike kadra, pored klasičnih tehnika za pretraživanje slika koriste se i tehnike koje obuhvataju vremensku dimenziju kadra. Na primer, rad [326] koristi varijansu histograma boja svih frejmova u kadru kao predstavu promenljivosti sadržaja u toku vremena.

Analiza pokreta kamere i objekata u kadru omogućava izdvajanje interesantnih objekata od pozadine [76]; tada se samo interesantni objekti mogu indeksirati. Analiza trajektorija objekata sprovođena je u više radova, npr. [233, 212, 270, 57].

Analiza zvuka je retko zastupljena u istraživanjima analize video zapisa [312]. Rad [286] definiše skup operacija nad vremenskim intervalima koje omogućavaju kombinovanu pretragu nad vizuelnim i audio indeksom video zapisa. Nekoliko radova npr. [320, 197] bave se identifikacijom značajnih trenutaka u snimcima sportskih nadmetanja (postizanje gola ili koša) pomoću analize audio kanala.

1.3.3 Analiza grupe kadrova

Pretraživanje velike kolekcije video zapisa se, prema do sada prikazanim tehnikama, svodi na pretraživanje velike kolekcije statičnih slika koje predstavljaju ključne frejmove. Međutim, broj ključnih frejmova je i dalje izuzetno velik naročito u situacijama kada je potrebno prikazati ih korisniku. Postoji više pristupa problemu grupisanja kadrova u strukture višeg hijerarhijskog nivoa – scene.

Otkrivanje kadrova sličnog sadržaja koji su bliski po trenutku prikazivanja tema je istraživanja zasnovanog na vremenski ograničenim klasterima (*time-constrained clustering*) [314]. Proces klasterovanja obuhvata vizuelne karakteristike i vremensku blizinu kadrova. Na osnovu rezultata klasterovanja moguće je formirati grafove prelaza scena (*scene transition graphs*, STG) gde čvorovi grafa predstavljaju kadrove (prikazane ključnim frejmom), a grane predstavljaju tok prikazivanja kadrova [315].

Generisanje skraćenih verzija video zapisa koji dovoljno dobro opisuju sadržaj originalne verzije, a ubrzavaju proces pregledanja (*browsing*) kolekcije zapisa tema je većeg broja radova. Različiti modeli za generisanje video rezimea prikazani su u [291, 266, 313]. Sistem MoCA [157] u potpunosti je namenjen generisanju rezimea. Članak [217] predstavlja upitni jezik za kolekcije video zapisa gde su rezultati upita takođe video zapisi generisani na osnovu sadržaja kolekcije. Kako je mera kvaliteta generisanog video rezimea proizvod čovekove subjektivne procene, vrednovanje ovakvih sistema je i dalje težak zadatak [157].

1.3.4 Prostorno-vremenski odnosi

Analiza prostorno-vremenskih odnosa između pojedinih objekata sadržanih u videu tema je velikog broja istraživanja. Mnogi autori (npr. [7]) ovakav

tip analize svrstavaju u analizu semantike video zapisa. U okviru ovog teksta smo se opredelili da napravimo razliku između analize prostorno-vremenskih odnosa i analize semantike višeg nivoa sadržane u zapisima. Naime, upit tipa „pronađi sve scene na kojima pas trči po peščanoj plaži“ teško može biti zadovoljen samo uočavanjem objekata na snimku i analizom njihovih trajektorija.

Grafički jezik za specifikaciju ponašanja objekata u prostoru i vremenu prikazan je u [63]. Upiti nad kolekcijama ovakvih reprezentacija mogu se postaviti na dva načina: iskazima namenski razvijene predikatske logike ili grafičkim putem. Bez obzira na način iskazivanja upita, sistem obavlja samo egzaktno poređenje formiranih reprezentacija. Jezik CVQL za postavljanje upita nad video zapisima u pogledu prostorno vremenskih odnosa objekata dat je u [145]. Specifikacija upitnog jezika za trajektorije objekata V-QBE data je u [318]. Model prostornih i vremenskih odnosa objekata u okviru objektno-orijentisanih baza podataka prikazan je u [154, 155]. Model prostornih odnosa između objekata u okviru jednog frejma i trajanja tih odnosa u toku vremena (tj. narednih frejmova) dat je u [215]. Analiza trajektorija objekata u okviru kadra i pretraživanje formiranih reprezentacija trajektorija opisano je u [297]. Algoritam za određivanje trajektorija objekata direktno iz MPEG zapisa dat je u [70, 71].

1.3.5 Analiza semantike

Osnovni način reprezentacije semantike vezane za video zapise je formiranje tekstualnih opisa (anotacija). Prema [121], tekstualne anotacije generišu se ili ručno ili pomoću analize zvučnog kanala i prepoznavanja govora. Sadržaj tekstualnih anotacija se, kod nekih sistema, bira iz unapred definisanog konačnog skupa izraza (rečnika), dok je kod drugih u pitanju slobodan tekst. Asociranje tekstualnih anotacija sa segmentima video zapisa analizom telopa u snimcima televizijskih vesti prikazano je u [11]. Opšti metod za izdvajanje teksta iz video snimaka i anotaciju segmenata dat je u [158]. Članak [132] opisuje arhitekturu sistema čiji model ima više hijerarhijskih nivoa segmentacije video zapisa sa pridruženim anotacijama. VideoText model [133] koristi tekstualne anotacije vezane za video segmente i definiše jezik za pretraživanje ovakvih struktura.

Tehnike za ekstrakciju semantike iz statičnih slika vezane za specifične domene primene mogu se primeniti i u analizi video zapisa. Primeri ovakvih

sistema obuhvataju analizu slika ulica (sa ekstrakcijom osobina kao što su dubina, širina, orijentacija ulice) [260], reklamnih spotova [41], prepoznavanje lica [12], prepoznavanje ključnih reči u govoru i kombinovanje sa prepoznavanjem lica [296, 136], i prepoznavanje ljudske kože [101]. Upotreba neuronskih mreža za *off-line* klasifikaciju ključnih frejmova u predefinisane kategorije i potom pretraživanje zadavanjem traženih kategorija prikazani su u [321].

Poseban interes među istraživačima postoji za analizu snimaka televizijskih vesti. Analiza snimaka vesti i klasifikacija kadrova u nekoliko kategorija (spiker čita, snimak na ulici, reklame, itd.) opisana je u [322]. Druga oblast od većeg interesa je analiza snimaka sportskih nadmetanja, gde je poseban akcenat stavljen na pojam događaja (*event*). Članak [7] donosi pregled nekoliko modela koji se mogu upotrebiti za ovu namenu.

Ekstrakcija semantike za kolekcije zapisa opšteg tipa rešavana je na više načina. Treniranje neuronskih mreža za asociranje šablona u vektorima osobina za pojedine interpretacije semantike prikazano je u [234]. *Data mining* metode za otkrivanje veza između osobina i anotacija u indeksu opisano je u [100]. Sistem prikazan u [121] bavi se formiranjem pravila koja definišu mapiranje vektora osobina na tekstualne anotacije. Vektori osobina su određeni na osnovu rezultata analize granica kadrova.

Upitni jezik HTL dat u [263, 162] posmatra video zapise kao hijerarhiju više podela na segmente. Za segmente najnižeg nivoa vezani su metapodaci. Konstrukcija složenih upita vrši se kombinovanjem elementarnih upita (po segmentima najnižeg nivoa) i vremenskih (*until*, *next*, *eventually*), modalnih (*at-next-level*, *at-scene-level*, *at-level-i*) i logičkih (*and*, *not*) operatora. Rad [209] bavi se ekstrakcijom semantičkih opisa višeg nivoa iz generisanih reprezentacija nižeg nivoa. Pri tome, u radu se primenjuju dva pristupa: jedan je zasnovan na pravilima kojima se mapiraju prostorno-vremenske strukture na semantičke koncepte višeg nivoa, a drugi je stohastički. Članak [110] definiše model podataka i upitni jezik za pronalaženje video zapisa koji obuhvata i karakteristike niskog nivoa i semantičke koncepte.

MPEG-7 standard [261] za kodiranje digitalnih video zapisa obuhvata i elemente za opis sadržaja prikazanog samim video snimkom. Ovi elementi omogućavaju pronalaženje video zapisa po sadržaju korišćenjem sledećih karakteristika: boje, teksture, oblika objekata, globalnog pokreta u snimku i

pokreta objekata. Upotreba *inference network* modela nad MPEG-7 video zapisima razmatrana je u [99].

1.4 Pronalaženje multimedijalnih dokumenata

Pregled istraživanja u oblasti pretrage multimedijalnih baza podataka po sadržaju (*content-based retrieval*, CBR) dat u članku [317] uočava razliku između *multi-media* CBR (jedinstvenog pretraživanja raznorodnih tipova medija) i *single-media* CBR (metoda pretraživanja za pojedinačne tipove medija). Pri tome se pod pojmom CBR podrazumeva i pretraživanje po formi i pretraživanje po sadržaju, onako kako su definisani na početku prvog poglavlja. Istaknuta je i disproporcija u rezultatima istraživanja između *multi-media* CBR i *single-media* CBR. Pored toga, članak [317] naglašava potrebu za ekstrakcijom implicitnog sadržaja iz semantički povezanih heterogenih tipova medija. Navodi i dve prednosti ovakvog pristupa: (1) sadržaj iz dva ili više semantički povezanih heterogenih tipova podataka može doneti više implicitnog sadržaja koji se ne može dobiti analizom pojedinačnih objekata i (2) sadržaj koji potiče iz interpretacije podataka iz dva ili više objekata može doneti rezultate sa većim nivoom sigurnosti.

Ovaj odeljak donosi prikaz rezultata istraživanja koja se bave pretraživanjem multimedijalnih tipova podataka, u smislu definisanja jedinstvenih metoda pretrage za različite tipove medija (tj. *multi-media* CBR). Istraživanja se mogu podeliti u dve grupe. Sa jedne strane nalaze se istraživanja koja potiču iz oblasti baza podataka i bave se sistemima koji rukuju sa više tipova medija. Osnovna zajednička karakteristika ovih radova je uniforman tretman različitih tipova medija pri čemu se multimedijalni objekti koji čine kolekciju posmatraju nezavisno jedni od drugih. Ovakav pristup omogućava formiranje kolekcije raznorodnih multimedijalnih objekata (npr. slika i video zapisa) i njeno pretraživanje pomoću jedinstvenih alata. Međusobno povezivanje pojedinih multimedijalnih objekata u veće semantičke celine (npr. dokumente) ostaje izvan opsega razmatranja ovih radova.

Sa druge strane, istraživanja koja potiču iz oblasti pronalaženja informacija polaze od pojma dokumenta kao osnovne jedinice posmatranja. U multimedijalnom ambijentu dokumenti se posmatraju kao strukturirani dokumenti koji obuhvataju više elemenata. Pojedini elementi mogu pripadati različitim

tipovima medija. Pretraživanje u ovakvim sistemima predstavlja pronalaženje dokumenata, pri čemu kriterijum pretraživanja može da obuhvata ograničenja postavljena različitim elementima dokumenta, odnosno tipovima medija.

1.4.1 Pretraživanje multimedijalnih objekata

Reprezentacija multimedijalnih objekata pomoću skupova vrednosti osobina (*feature sets*) posmatra problem pretraživanja objekata kao problem pretraživanja njima odgovarajućih tačaka u n-dimenzionalnom prostoru. Uz preslikavanje objekata na skupove osobina, mora postojati i metrika koja definiše sličnost objekata u dobijenom n-D prostoru. Za potrebe efikasnog pretraživanja u n-D prostoru mogu se upotrebiti strukture podataka poznate kao prostorne strukture za pristup podacima (*spatial access methods*, SAM). SAM strukture omogućavaju efikasno pretraživanje za sledeće četiri klase upita:

- **Izbor ograničenjem rastojanja.** Za datu kolekciju objekata O_1, O_2, ..., O_n i upit Q potrebno je pronaći sve objekte koji se nalaze na rastojanju manjem od ϵ od upita Q, tj. $D(Q, O_i) < \epsilon$.
- **Izbor poređenjem podskupova osobina.** Slično kao prethodni slučaj, ali je omogućeno da upit Q specificira podskup traženih osobina. Na taj način, za kolekciju objekata $O_1, O_2, \ldots, O_n$, upit Q i rastojanje ϵ, moguće je pronaći *delove* objekata koji zadovoljavaju upit.
- k **najbližih suseda.** Pronalaženje k objekata u kolekciji koji su najbliži (tj. najsličniji) datom upitu.
- **Svi parovi.** Pronalaženje svih parova objekata koji su međusobno udaljeni za rastojanje manje od ϵ.

Najpoznatija SAM struktura podataka je R-stablo, inicijalno prikazana u [108]. R-stablo tretira objekat u prostoru pomoću njegovog minimalnog obuhvatajućeg pravougaonika (*minimum boundind rectangle*, MBR). Pravougaonici koji sadrže objekte odgovaraju listovima R-stabla. Oni se dalje hijerarhijski grupišu u veće pravougaonike koji predstavljaju roditeljske čvorove čime se konačno dobija struktura stabla.

Kasnije varijacije ove strukture predstavljaju R*-stablo [24] i Hilbertovo R-stablo [135]. Druge slične strukture su razvijane posebno za rad sa prostorima visoke dimenzionalnosti. TV-stabla [159] adaptivno koriste samo neke od postojećih dimenzija. SR-stabla [138] koriste hipersfere pored hiperpravougaonika

kao obuhvatajuće regione. X-stabla [26] prelaze na sekvencijalnu pretragu za prostore izuzetno visoke dimenzionalnosti. Generalna mana svih SAM struktura je što pretraživanje tačaka u prostorima visoke dimenzionalnosti ima eksponencijalnu zavisnost od dimenzije prostora [52].

Pored ovih SAM struktura u literaturi se, za ovu namenu, koriste i *linear quadtrees* [80, 247] i *grid files* [199], ali su se, prema [79], strukture proistekle iz koncepta R-stabla pokazale kao najefikasnije.

Projekat GEMINI (*Generic Multimedia Indexing*) [79] oslanja se na SAM strukture za pretraživanje n-D prostora, ali definiše dodatne korake u procesu pretrage koji ga čine efikasnijim. GEMINI polazi od ideje da se za svaki tip medija definiše metrika D koja predstavlja meru međusobne sličnosti objekata. Umesto sekvencijalnog prolaska kroz celokupnu kolekciju objekata i pronalaženja onih koji zadovoljavaju postavljeni uslov izračunavanjem vrednosti metrike (što može biti računski složen zadatak), GEMINI pristup definiše sledeći postupak:

1. Određivanje preslikavanja F koje dati tip objekata preslikava na f-dimenzionalni prostor osobina, tako da je $F(O)$ tačka u f-D prostoru koja odgovara objektu O.
2. Određivanje metrike D_f u f-dimenzionalnom prostoru osobina za koju važi $D_f(F(O_1), F(O_2)) \leq D(O_1, O_2)$ (tzv. *lower-bounding* uslov).
3. Upotreba neke od SAM struktura za pretraživanje tačaka u f-D prostoru, pri čemu se za poređenje koristi metrika D_f.

Formiranje preslikavanja F zapravo predstavlja redukciju dimenzionalnosti prostora u kome se vrši pretraživanje zanemarivanjem manje značajnih komponenti vektora. Eksponencijalna zavisnost SAM struktura od broja dimenzija (tzv. "*dimensionality curse*") je osnovni razlog za ovu operaciju. SAM strukture, sa druge strane, obezbeđuju efikasno pronalaženje svih tačaka koje zadovoljavaju upit bez testiranja svih tačaka u kolekciji. Rezultat pretrage pomoću SAM strukture je skup tačaka koji sadrži sve tačke koje zadovoljavaju upit (zbog *lower bounding* uslova) ali može da sadrži i određeni broj tačaka koje, iako zadovoljavaju upit korišćenjem metrike D_f, ne zadovoljavaju upit uz korišćenje metrike D. Takve tačke se iz dobijenog skupa moraju ukloniti sekvencijalnom pretragom.

Primena ovog postupka na primeru vremenskih serija [6], gde su objekti predstavljeni kao vektori u prostoru R^n (npr. $n = 365$ za slučaj godišnjeg kretanja cena akcija na berzi), koristi L_2 normu kao osnovnu metriku D. Preslikavanjem F se kompletna vremenska serija zamenjuje sa prvih nekoliko koeficijenata njene diskretne Furijeove transformacije, i za tako dobijen skup tačaka koristi se R*-stablo [24] kao SAM struktura za pretragu.

Primena GEMINI postupka u slučaju statičkih slika [80] obuhvata upotrebu histograma boja kao osnovnog reprezenta slike i funkcije iz jednačine 1.19 kao osnovne metrike. Redukovani 3-dimenzionalni prostor dobija se izračunavanjem prosečne vrednosti R, G i B komponenti histograma, a u takvom prostoru kao metrika koristi se L_2 norma. Upotrebljena SAM struktura je ponovo R*-stablo.

Struktura ContIndex [306] predstavlja indeksnu strukturu koja omogućava predstavljanje objekata pomoću skupa osobina gde je svaka osobina izražena kao tačka u zasebnom prostoru ili pomoću simbola koji predstavlja reprezentaciju semantike sadržane u objektu.

Svaki nivo ContIndex stabla predstavlja klasifikaciju objekata po posebnom kriterijumu. Čvorovi stabla su povezani, pored uobičajenih veza prema roditelju i potomcima, i između čvorova istog nivoa. Ovakva struktura omogućava i efikasnu navigaciju kroz kolekciju. Kreiranje indeksa vrši se pomoću samoorganizujuće neuronske mreže uz učešće eksperta. Ova tehnika je eksperimentalno potvrđena u sistemu za indeksiranje slika ljudskog lica i registrovanih robnih marki.

Formiranje matematičkog modela koji predstavlja apstrakciju multimedijalnih objekata i upitnog jezika nad takvim apstrakcijama iskorišćeno je kao polazište sistema MACS [39, 175]. MACS definiše pojam apstrakcije medija (*media abstraction*) kao uređenu sedmorku $M = (ST, fe, \lambda, R, F, Var_1, Var_2)$ gde je ST skup objekata nazvanih stanja (*states*); fe je skup objekata zvanih osobine (*features*); λ je preslikavanje $ST \times fe \rightarrow [0, 1]$; Var_1 je skup promenljivih stanja (*state variables*) koje uzimaju vrednosti iz skupa stanja; Var_2 je skup promenljivih osobina (*feature variables*) koje uzimaju vrednosti iz skupa osobina; R je skup *fuzzy* relacija između stanja iz ST; F je skup *fuzzy* relacija između osobina. Svaka relacija iz F je ili preslikavanje $fe^i \rightarrow [0, 1]$ (kada su relacije među osobinama nezavisne od stanja) ili preslikavanje iz $fe^i \times ST \rightarrow [0, 1]$ (kada su relacije među osobinama zavisne od stanja). Multimedijalna

baza podataka d, u MACS terminologiji, data je skupom apstrakcija medija $M = \{M_1, M_2, \ldots, M_n\}$. U slučaju statičnih slika, MACS definiše apstrakciju medija na sledeći način:

- *Stanja*. Svaka slika u kolekciji smatra se stanjem.
- *Osobine*. Sadržaji koji se nalaze u okviru slika (osobe, objekti, itd.).
- *Preslikavanje osobina*. Preslikavanje λ izražava stepen sigurnosti sa kojim se data osobina pojavljuje u datom stanju.
- *Relacije*. Primer relacije koja zavisi od stanja je relacija sa nazivom *is_wearing* koja ima četiri argumenta: ime osobe, odevni predmet, boja (sva tri su osobine) i stanje. Jedna n-torka ove relacije može da glasi ('Pera Perić', 'košulja', 'crvena', file10) : 0.99, sa značenjem da sa sigurnošću od 99% Pera Perić nosi crvenu košulju na slici file10.

MACS sistem definiše i sopstveni upitni jezik koji poseduje četiri primitivne funkcije:

- $featureInState(d, f, s, c)$. Rezultat funkcije je vrednost $v \in [0, 1]$ ako se osobina f javlja u stanju s u okviru baze d sa sigurnošću $v \geq c$.
- $featureInAnyState(d, f, c)$. Rezultat funkcije je skup stanja u kojima se osobina f javlja sa sigurnošću $v \geq c$.
- $anyFeatureInState(d, s)$. Rezultat funkcije je skup osobina i njima pridruženih nivoa sigurnosti sa kojima se one javljaju u stanju s baze d.
- $anyFeatureInAnyState(d, c)$. Rezultat funkcije je skup uređenih trojki (f, s, v) gde je f osobina prisutna u stanju s sa sigurnošću $v \geq c$ za datu bazu d.

MACS podrazumeva da se identifikacija osobina u stanjima implementira zasebno za svaku konkretnu primenu. Sa druge strane, poseduje sistem za skladištenje i pretraživanje iskaza tipa „osobina f se pojavljuje u stanju c sa sigurnošću v" koji se može upotrebiti u svim primenama.

Sistem HERMES [39] predstavlja nadgradnju MACS sistema na takav način da obezbeđuje povezivanje MACS-a sa sistemima za upravljanje relacionim bazama podataka i odgovarajući prošireni upitni jezik. HERMES poseduje i mogućnost dodavanja programskih modula koji će biti pozivani nakon procesa pretrage. Ova arhitektura je iskorišćena za dodavanje modula koji se

bavi automatskim relaksiranjem upita u slučaju da pretraga nije vratila rezultate. Sistem za relaksaciju upita za MACS predstavljen je u [175].

Upotreba objektno-orijentisanih baza podataka za rukovanje multimedijalnim objektima tema je određenog broja radova [74, 75, 224, 301, 2]. Pregled potrebnih karakteristika ovakvih baza dat je u člancima [200, 5]. U [97] predstavljen je objektni model koji obuhvata različite tipove medija i omogućava kombinovanje objekata tokom prezentacije i sintezu novih. Sistem Jasmine [128] namenjen je rukovanju multimedijalnim objektima kao podrška kompleksnim CAD aplikacijama.

Članak [185] definiše klasifikaciju multimedijalnih aplikacija prema tipičnim operacijama u rukovanju multimedijalnim podacima. Skup multimedijalnih podataka može biti statički ili dinamički. Statički skup označava podatke koji su pre svega namenjeni za čitanje, dok se dinamički skup podataka često menja. Pored toga, skup podataka može biti pasivan ili aktivan. Pasivan skup očekuje komande korisnika koje vrše čitanje ili promenu, dok aktivni skupovi mogu da izazovu prikaz ili promenu nekog drugog skupa podataka. U skladu sa ovakvom klasifikacijom identifikovane su sledeće četiri kategorije multimedijalnih aplikacija:

- arhiviranje (statički + pasivni)
- učenje, reklamiranje, zabava (statički + aktivni)
- dizajn, kreiranje sadržaja, izdavaštvo (dinamički + pasivni)
- nadgledanje, revizije (dinamički + aktivni)

Rad [176] definiše klasifikaciju arhitektura sistema za rukovanje multimedijalnim bazama podataka. Model arhitekture koji je jedinstven i proširiv podrškom za različite tipove medija nazvan je *single DBMS architecture*, dok model koji poseduje primarni SUBP i poziva module sekundarnih SUBP specijalizovanih za pojedine tipove medija je nazvan *primary-secondary DBMS architecture*.

1.4.2 Pretraživanje strukturiranih multimedijalnih dokumenata

Članak [179] analizira osnovne karakteristike klasičnih (tekstualnih) IR sistema i definiše konceptualne zahteve koje multimedijalni IR sistemi moraju da ispune. Pojam zone konteksta (*context area*) definisan je kao deo dokumenta

na koga se može ograničiti pojedini upit za pretraživanje. Slično kao i IR sistemi za rad strukturiranim tekstualnim dokumentima, multimedijalni IR sistemi moraju obezbediti mogućnost definisanja zone konteksta od strane korisnika. Model dokumenata mora posedovati sledeće koncepte:

- *klasifikacija*, gde se objekti sa zajedničkim svojstvima grupišu u klase,
- *agregacija*, koja predstavlja veze sadržavanja između objekata i njihovih komponenti i omogućava gradnju hijerarhija sadržavanja i
- *generalizacija*, pomoću koje klase mogu sačinjavati hijerarhije nasleđivanja.

Upiti u multimedijalnim IR sistemima moraju, prema [179], zadovoljiti sledeće zahteve:

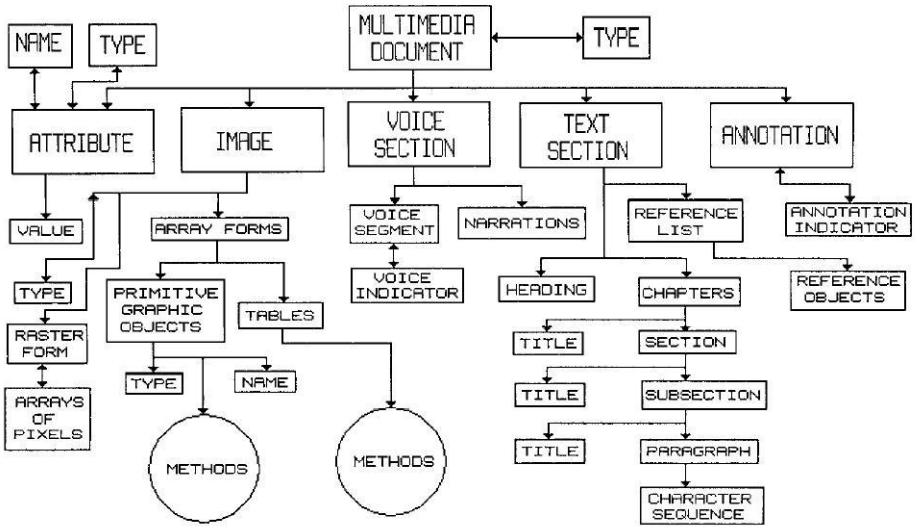
- jednak tretman objekata i klasa, u smislu da se nad svojstvima klasa mogu postavljati upiti na isti način kao i nad svojstvima objekata,
- jednak tretman definicionog (shema) i činjeničnog dela (podaci) dokumenta, tako da je moguće postavljanje upita nad definicijama dokumenata na isti način kao i nad samim dokumentima,
- postavljanje nepreciznih upita, sa nepotpunom specifikacijom strukture i sadržaja traženih dokumenata i
- postavljanje upita koji vraćaju svojstva umesto dokumenata, npr. upita koji ispituju relaciju između dva objekta.

Jedan od prvih sistema za rukovanje multimedijalnim dokumentima, MINOS [49, 50], poseduje module za prikaz dokumenta i kretanje kroz kolekciju dokumenata i editore za pojedine tipove medija. Pored toga, svaki dokument se može nalaziti u jednom od dva stanja: u toku izrade (*editing state*) i arhiviran (*archived state*). Arhivirani dokumenti su zaštićeni od daljih izmena i smešteni u repozitorijume kojima rukuje poseban modul sistema – arhiver. Ovakvim dokumentima je moguće pristupati i koristiti njihove elemente prilikom formiranja novih dokumenata. Oni su klasifikovani na nezavisne (*independent*) i zavisne (*dependent*) objekte. Nezavisni objekti mogu samostalno postojati u arhivi, dok zavisni mogu biti prisutni samo kao deo drugih objekata.

MINOS definiše i sopstveni model dokumenata koji se bazira na objektno-orijentisanom modelu podataka. Model dokumenata formira se sa dva aspekta,

kao *logički* i *fizički* model. Logički model definiše logičke komponente dokumenata. Fizički model definiše prezentaciju dokumenta na datom izlaznom uređaju. Mapiranje logičkog modela dokumenta na neki od fizičkih modela dato je *dokumentom mapiranja*.

Dokument u logičkom modelu dokumenata je instanca odgovarajuće klase dokumenata, ima jedinstveni identifikator i sačinjen je od atributa, slika, zvučnih snimaka, fragmenata teksta i anotacija (slika 1.1, preuzeta iz [49]). Atributi (*attribute*) imaju sebi pridružen naziv, tip i jedan ili više sadržaja. Fragment teksta (*text section*) može imati naslov, više *chapter* objekata i listu referenci. Zvučni snimak (*voice section*) može sadržati više segmenata (koji se reprodukuju samo po akciji korisnika) i naracija (koje se reprodukuju automatski prilikom pregleda određene stranice dokumenta). Anotacije (*annotations*) predstavljaju veze sa drugim dokumentima u kolekciji.



Slika 1.1: Logički model dokumenta u MINOS sistemu

Među pojedinim elementima dokumenata moguće je formirati veze (one nisu prikazane na slici 1.1). Interpretacija veza zavisi od konteksta upotrebe

dokumenta; veze se mogu upotrebiti za prezentaciju, ekstrakciju podataka, navigaciju kroz dokument ili adresiranje sadržaja.

Upiti u MINOS-u [50] mogu da sadrže sledeće komponente:

1. tip dokumenta,
2. konjunkcije vrednosti ili opsega vrednosti atributa,
3. konjunkcije disjunkcija reči ili koje se javljaju u tekstualnim elementima,
4. egzistencija zvučnog zapisa,
5. egzistencija slike,
6. približan položaj slike u okviru dokumenta,
7. konjunkcije reči koje se javljaju u tekstu asociranom slici,
8. egzistencija atributa ili vrednosti atributa za slike i
9. konjunkcije prethodnog.

Sistem MULTOS [28, 179], pridržavajući se ODA (*Office Document Architecture*) [126] standarda, takođe razlikuje logičku strukturu *logical structure*) i prezentacionu strukturu (*layout structure*) dokumenata. Međutim, kako logička struktura dokumenata ODA standarda omogućava definisanje sintaksnih koncepata (poglavlja, naslovi, pasusi, itd.), MULTOS pravi razliku između logičke i konceptualne strukture (*conceptual structure*) dokumenta. Dokumenti se, sa stanovišta konceptualne strukture, sastoje iz komponenti (*conceptual components*). Model dokumenata MULTOS-a, prikazan u [220], razvijen je korišćenjem semantičkog modela podataka [113]. Upitni jezik, razvijen namenski za MULTOS [28], koristi konceptualni model dokumenta.

Druga novina koju donosi MULTOS je pojam slabog tipa dokumenta *weak document type*). Definicija slabog tipa dokumenta sadrži konceptualne komponente koje intance dokumenta moraju posedovati, ali ne postavlja ograničenja na dodavanje novih opcionih komponenti u instancama. Ovaj pristup omogućava modelovanje dokumenata na različitim nivoima detalja.

Sa stanovišta podrške različitim tipovima medija, MULTOS klasifikuje konceptualne komponente na aktivne i pasivne. Aktivne čine strukturirani alfanumerički podaci i slobodan tekst, dok pasivne obuhvataju slike i zvučne zapise (video zapisi nisu podržani). Bitna razlika je u njihovoj upotrebljivosti prilikom pretraživanja: za aktivne komponente moguće je postavljati upite po njihovom sadržaju ili postojanju, dok je za pasivne moguće samo postavljanje upita o njihovom postojanju. Drugim rečima, sistem ne obuhvata pretraživanje

multimedijalnih tipova podataka po njihovom sadržaju, već samo po sadržaju njima pridruženih alfanumeričkih opisa.

IR model za strukturirane multimedijalne dokumente predstavljen u članku [180] rezultat je dužeg perioda istraživanja [181, 254, 178, 182]. Model definiše okvir koji obuhvata pretraživanje dokumenata po formi, sadržaju i strukturi (prema definicijama ovih pojmova datim u uvodnom delu ovog poglavlja). U članku [180] data je verzija modela koji omogućava rukovanje tekstom i statičnim slikama, ali se model može proširiti podrškom za druge tipove medija. Kao jezik za reprezentaciju znanja o dokumentima koristi se deskriptivna logika (*description logic*) $\mathcal{ALCO}$, kao kontrakcija predikatske logike.

$\mathcal{ALCO}$ poseduje *koncepte*, *uloge* i *individualne konstante* kao osnovne gradivne elemente. Koncepti predstavljaju skupove objekata; uloge definišu odnose između dva koncepta ili konstante; individualne konstante su nazivi za pojedine konkretne objekte. Elementarni iskazi koji se mogu formirati u $\mathcal{ALC}$ (za koncept C, ulogu R i konstantu a) pripadaju jednom od tri tipa:

1. a je instanca C, u oznaci $C(a)$; na primer, Muzičar(pera)
2. a_1 je povezan ulogom R sa a_2, u oznaci $R(a_1, a_2)$; na primer, Prijatelj(pera, mita)
3. koncept C_2 obuhvata C_1, u oznaci $C_1 \sqsubseteq C_2$; na primer, Pijanista $\sqsubseteq$ Muzičar.

Za potrebe izražavanja nepreciznosti vezanih za pronalaženje informacija $\mathcal{ALCO}$ je proširena *fuzzy* elementima na takav način da su elementarni iskazi oblika $\langle \alpha, n \rangle$, gde je α ne-*fuzzy* elementarni iskaz, a $n \in [0, 1]$.

Za ovako definisanu *fuzzy* $\mathcal{ALCO}$ deskriptivnu logiku formirani su modeli forme slike i teksta, jedinstveni model sadržaja dokumenta i model strukture dokumenta. Nad ovim modelima definisane su uloge koje omogućavaju izražavanje znanja o dokumentu ili njegovim elementima. Uloge namenjene opisivanju forme slika obuhvataju HAIR (*has atomic image region*), HIS (*has image region*), HS (*has shape*) i HC (*has color*); uloga vezana za formu teksta je ST (*similar text*); uloge vezane za strukturu dokumenta su HN (*has node*), Root, Leaf, HCh (*has child*), HP (*has parent*), HA (*has ascendant*) i HD (*has descendant*).

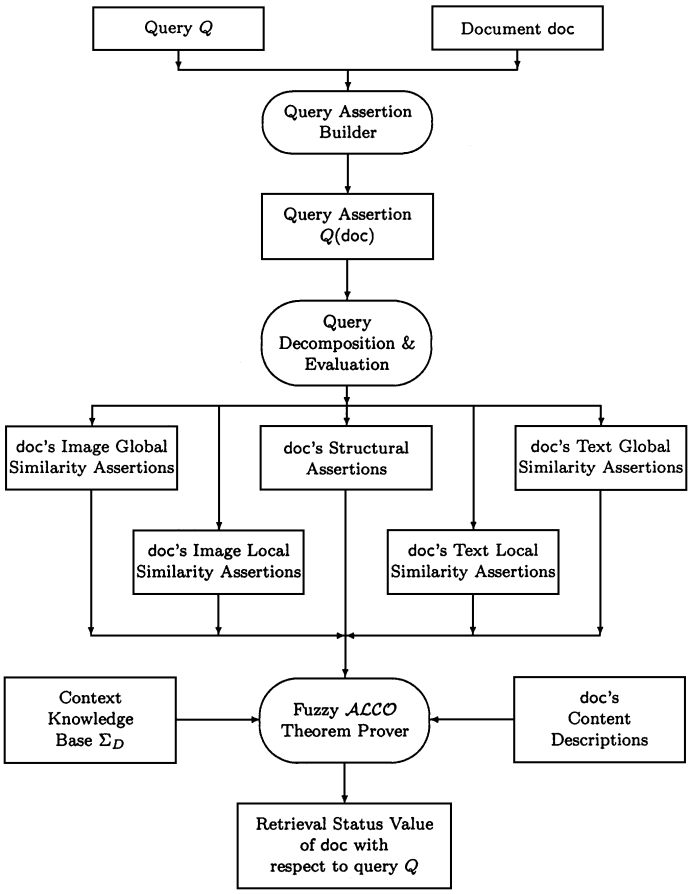
Upiti koji se postavljaju nad kolekcijama dokumenata su $\mathcal{ALCO}$ izrazi koji koriste prethodno pomenute uloge. Procedura izračunavanja upita za jedan

dokument prikazana je na slici 1.2 preuzetoj iz [180]. Procedura koristi funkcije modula za automatsko dokazivanje teorema (prikazanog na slici kao *Fuzzy $\mathcal{ALCO}$ Theorem Prover*). Iz prikazanog se vidi da se podrška za nove tipove medija u ovakav sistem može dodati formiranjem odgovarajućeg modela za dati tip medija i definisanjem uloga u okviru $\mathcal{ALCO}$ logike. Pretraživanje po svim tipovima medija odvija se uniformno u okviru modula za dokazivanje teorema.

Podrška za *relevance feedback* formirana je isključivo za pretraživanje po formi. Razvijeni su posebni RF modeli za slike i tekst. Na sličan način mogu se definisati i RF modeli za druge tipove medija.

Određeni broj istraživanja bavi se modeliranjem prezentacionog aspekta multimedijalnih dokumenata. Članak [152] definiše grafički jezik GVISUAL namenjen pretraživanju kolekcija multimedijalnih prezentacija. Pored toga, definisan je i jezik GCalculus/S, koji predstavlja formalnu osnovu za GVISUAL. Multimedijalna prezentacija se reprezentuje kao usmereni graf čiji čvorovi predstavljaju pojedine objekte (odnosno tokove) a grane predstavljaju sekvencijalno ili konkurentno prikazivanje pojedinih objekata tokom prezentacije. Za svaki čvor grafa moguće je vezati objekte koji reprezentuju sadržaj prikazan datim čvorom. Nad ovakvim modelom definisan je jezik GVISUAL koji omogućava pretraživanje grafova pomoću operatora *next*, *until* i *connected*. GVISUAL omogućava definisanje ograničenja nad objektima vezanim za čvorove samo u pogledu njihove egzistencije, ali ne i međusobnog odnosa. Implementacija ovog jezika sprovedena je u okviru sistema ViSiOn koji omogućava skladištenje, pretraživanje i prikaz multimedijalnih prezentacija.

Članak [47] definiše model multimedijalnih prezentacija pomoću ATN mreža (*augmented transition networks*) i tzv. ulaznih stringova (*input strings*). Model obuhvata prezentacije sačinjene od tekstualnih, audio, video zapisa i statičnih slika. Omogućava reprezentaciju redosleda prikazivanja pojedinih elemenata prezentacije i međusobnog prostorno-vremenskog odnosa elemenata u okviru slika i pojedinih frejmova video zapisa. Za kodiranje ovakvih informacija koristi se koncept multimedijalnih ulaznih stringova. Kao rezultat ovakvog pristupa kolekcija multimedijalnih prezentacija može se reprezentovati kolekcijom stringova. Pretraživanje ovakve kolekcije svodi se na pretraživanje stringova korišćenjem regularnih izraza.



Slika 1.2: Procedura izračunavanja $\mathcal{ALCO}$ upita

Poglavlje 2

Model proširivog sistema za pronalaženje multimedijalnih dokumenata

2.1 Osnovne karakteristike sistema

Prikaz modela proširivog sistema za pronalaženje multimedijalnih dokumenata nazvanog XMIRS (*extensible multimedia information retrieval system*) započinje pregledom njegovih najvažnijih karakteristika.

Struktura dokumenata zasnovana na XML jeziku. Kao i drugi modeli za pronalaženje multimedijalnih dokumenata, XMIRS model dokumenata polazi od stabla kao osnove strukture dokumenata. Upotreba XML jezika [37] za reprezentaciju XMIRS dokumenata u najvećoj meri određuje i mogućnosti pripadajućeg modela dokumenata. XML je jezik prevashodno namenjen opisivanju strukturiranih tekstualnih dokumenata ali poseduje i mogućnosti ugradnje elemenata koji pripadaju drugim tipovima medija. Ove mogućnosti su i iskorišćene u okviru XMIRS modela. Na taj način, svi tipovi medija od kojih se sastoji dokument su ravnopravni – XML jezik se koristi samo kao sredstvo izražavanja strukture i sadržaja dokumenata.

Reprezentacija dokumenata pomoću XML jezika donosi i osobine koje su vezane za XML jezik uopšte: prenosivost dokumenata na različite platforme

i aplikacije, široka rasprostranjenost u različitim domenima, i tehnološka podrška.

Jedinstvena globalna identifikacija dokumenata. Upotreba standardnih transportnih protokola, kao što su HTTP [86], FTP [216] ili SOAP [102, 103], omogućava široku dostupnost dokumenata iz različitih tipova klijenata. Standarni protokoli i URI standard [27] za dodelu jedinstvenih identifikatora resursima obezbeđuju aplikacijama nezavisnost od lokacije skladištenja i isporuke dokumenata. U sistemu za pronalaženje dokumenata to omogućava nezavisnost instalacija softverskih modula za pretraživanje od modula za skladištenje dokumenata. Drugim rečima, moguće je instalirati različite sisteme za pronalaženje dokumenata koji rukuju istim kolekcijama dokumenata. Pored toga, moguće je koristiti više kolekcija dokumenata skladištenih na različitim mestima u okviru jednog sistema za pronalaženje dokumenata.

Rukovanje različitim tipovima dokumenata i različitim kolekcijama dokumenata. Koncept klasifikacije, naveden u [179] kao jedan od obaveznih koncepata koje model multimedijalnih dokumenata mora da zadovolji, omogućava grupisanje dokumenata jednake strukture u klase (tipove). Prethodno poznavanje strukture dokumenata omogućava sistemu za pronalaženje da izgradi svoje interne strukture tako da može efikasno da podrži operacije nad dokumentima kao što su indeksiranje ili pretraživanje.

Pod kolekcijom dokumenata podrazumeva se podskup ukupnog skupa dokumenata. Skupovi dokumenata istog tipa mogu se posmatrati kao kolekcije. Međutim, to ne mora biti jedini skup kolekcija; one mogu sadržati i dokumente različitog tipa. Definisanje pojedinih kolekcija dokumenata (osim kolekcija određenih tipom dokumenata) obavlja korisnik. Osnovna namena kolekcija je sužavanje skupa pretraživanih dokumenata prilikom postavljanja upita.

Definicije tipova dokumenata u XMIRS sistemu koriste XML Schema [281, 30] standard. Ovaj standard predstavlja zamenu za koncept DTD-a koji sastavni deo osnovne XML specifikacije. Za razliku od DTD-a, XML Schema specifikacija strukture dokumenata ima na raspolaganju veliki broj ugrađenih tipova podataka i mogućnost definisanja novih tipova. XML Schema standard podržava koncepte klasifikacije, agregacije i generalizacije opisane u [179] (v. odeljak 1.4.2).

Jednak tretman dokumenata i specifikacija tipova dokumenata. Ako se specifikacije tipova dokumenata izražavaju istim sredstvima kao i sami

dokumenti, tada se same specifikacije mogu tretirati kao dokumenti u posebnoj kolekciji koji se mogu pretraživati. Na taj način moguće je skupove dokumenata pretraživati korišćenjem osobina njihovih tipova. XML Schema dokumenti koji predstavljaju definicije strukture tipova osnovnih XML dokumenata u kolekciji su takođe XML dokumenti, tako da se iste metode pretraživanja mogu primeniti kako na osnovne XML dokumente, tako i na pripadajuće XML Schema specifikacije tipova.

Proširivost sistema različitim modulima za pretraživanje. Dosadašnji modeli pronalaženja informacija u multimedijalnim dokumentima poseduju bar jedno od sledećih ograničenja:

1. rukuju dokumentima koji imaju unapred određenu, fiksnu strukturu,
2. omogućavaju rad sa ograničenim skupom tipova medija i
3. za sve podržane tipove medija nameću sopstveni model pretraživanja.

Stariji sistemi MINOS [49, 50] i MULTOS [28, 179] omogućavaju rad sa ograničenim skupom tipova medija. MINOS, pored toga, definiše i fiksnu strukturu dokumenata. Sa druge strane, sveobuhvatni multimedijalni IR model predstavljen u članku [180] omogućava rad sa proizvoljnim tipovima medija – uz definisanje odgovarajućih podmodela za svaki tip – ali nameće upotrebu $\mathcal{ALCO}$ deskriptivne logike za tu svrhu. Svi moduli ovog sistema moraju imati reprezentaciju forme i sadržaja izražene pomoću $\mathcal{ALCO}$ koncepata kako bi jedinstveni modul za pretraživanje, *fuzzy* $\mathcal{ALCO}$ *theorem prover*, mogao da ih koristi.

Osnovni cilj ove disertacije je razvoj IR modela koji ne nameće ovakva ograničenja, tj. omogućava integraciju podrške za različite tipove medija bez nametanja jedinstvenog modela reprezentacije forme ili sadržaja. Podršku za ovakvu integraciju potrebno je sprovesti sa dva aspekta: (1) u okviru modela pretraživanja sistema i (2) u okviru softverske arhitekture sistema. Razmatranje drugog aspekta sistema neophodno je kako bi se omogućila integracija postojećih softverskih sistema za pretraživanje pojedinih tipova medija u celoviti sistem.

Nepostojanje jedinstvenog modela pretraživanja nalaže da se podrška za pojedine tipove medija organizuje u programske module koji imaju funkciju kako kreiranja reprezentacije forme ili sadržaja (tj. indeksiranja) tako i pretraživanja. Jedan tipom medija može rukovati više ovakvih modula. Veza modula sa jezgrom sistema odvija se putem jedinstvenog interfejsa.

Proširivost sistema različitim modelima pretraživanja dokumenata. Za klasične IR sisteme definisan je veći broj modela pretraživanja. Analize performansi klasičnih IR sistema objavljivane su u više navrata [267, 59, 241, 23]. Modeli multimedijalnih IR sistema koji nisu ograničeni na određeni broj tipova medija (npr. [180, 188]) su i dalje malobrojni pa ovakvih analiza do sada nije bilo. Cilj XMIRS sistema je da omogući upotrebu različitih modela pretraživanja nad istim kolekcijama dokumenata.

Skladištenje i indeksiranje dokumenata. Sistem XMIRS omogućava upotrebu različitih strategija skladištenja dokumenata, kao i povezivanje sa drugim sistemima koji ne moraju nužno da rukuju XML dokumentima. Indeksiranje dokumenata u okviru XMIRS-a podrazumeva upotrebu različitih modula za indeksiranje. Mogućnost konfigurisanja XMIRS-a tako da se pojedini moduli izvršavaju na različitim, putem računarske mreže povezanim, platformama obezbeđuje potrebnu skalabilnost sistema [188].

2.2 Dokumenti, kolekcije, indeksi i upiti

2.2.1 Struktura dokumenata

U prethodnom odeljku naglašena je upoteba XML jezika za reprezentaciju dokumenata. Ta činjenica bi mogla da znači da je model XML dokumenata ujedno i model dokumenata kojima rukuje XMIRS. Međutim, jedinstveni model XML dokumenata ne postoji. U okviru različitih standarda vezanih za XML jezik definisana su četiri različita modela dokumenata: XML Information Set model [58], XPath model [54], DOM model [123] i XQuery 1.0 i XPath 2.0 model [85]. Za potrebe XMIRS sistema definisan je model zasnovan na XML Infoset modelu iz koga su izdvojeni samo koncepti koji su od interesa za XMIRS.

XMIRS rukuje isključivo dokumentima čiji tip je definisan odgovarajućim XML Schema dokumentom. Prilikom interpretiranja sadržaja dokumenta podrazumeva se da je dokument zadovoljio proces validacije koga obavlja XML parser. Ovakav model se u XQuery 1.0 modelu naziva PSVI (*post schema validation infoset*).

Model dokumenata za XMIRS posmatra dokument kao strukturu stabla čiji čvorovi pripadaju jednom od sledeća četiri tipa:

- *čvor dokumenta*,
- *čvor elementa*,
- *čvor atributa* i
- *čvor teksta*.

Čvor dokumenta predstavlja koren stabla dokumenta. Postoji tačno jedan čvor dokumenta za svaki dokument. Sadrži sledeća svojstva:

- ⟨element⟩ – čvor elementa koji predstavlja korenski element i
- ⟨docid⟩ – atribut koji sadrži jedinstveni identifikator dokumenta.

Čvor elementa predstavlja jedan element u okviru XML dokumenta. Sadrži sledeća svojstva:

- ⟨namespace⟩ – naziv prostora imena kome pripada odgovarajući element,
- ⟨local name⟩ – lokalni deo naziva odgovarajućeg elementa,
- ⟨children⟩ – lista čvorova-naslednika u stablu; lista može da sadrži samo čvorove elementa ili čvorove teksta,
- ⟨attributes⟩ – skup čvorova atributa koji odgovaraju atributima datog elementa u dokumentu i
- ⟨parent⟩ – čvor elementa koji sadrži ovaj čvor u svojoj listi ⟨children⟩ ili čvor dokumenta, ako njegovo svojstvo ⟨element⟩ sadrži ovaj čvor.

Čvor atributa predstavlja atribut jednog elementa u dokumentu. Sadrži sledeća svojstva:

- ⟨namespace⟩ – naziv prostora imena kome pripada odgovarajući atribut,
- ⟨local name⟩ – lokalni deo naziva odgovarajućeg atributa,
- ⟨value⟩ – sadržaj odgovarajućeg atributa i
- ⟨owner⟩ – čvor elementa koji sadrži ovaj čvor u svom skupu ⟨attributes⟩.

Čvor teksta predstavlja tekstualni sadržaj elemenata u dokumentu. Tekstualni sadržaj može biti naveden direktno u okviru sadržaja elementa, i može biti rezultat interpretiranja tekstualne reference ili CDATA sekcije. Sadrži sledeća svojstva:

- ⟨value⟩ – tekstualni sadržaj naveden u okviru dokumenta i
- ⟨parent⟩ – čvor elementa koji sadrži ovaj čvor u svojoj listi ⟨children⟩.

Za čvorove teksta postoji i dodatno ograničenje da se ne mogu pojaviti dva čvora teksta na susednim mestima u ⟨children⟩ listi čvora elementa.

Objekti koji pripadaju drugim tipovima medija (osim teksta) se, u okviru XMIRS dokumenata, mogu nalaziti kao ugrađeni ili eksterni. Ugrađeni objekti se smeštaju u čvorove teksta. Tip ovih čvorova je, sa stanovišta XML Schema tipova podataka [30], *base64Binary* ili *hexBinary*. Binarni zapis ovih objekata potrebno je kodirati na odgovarajući način prilikom smeštanja u XML zapis dokumenta. Eksterni objekti smešteni su izvan XML zapisa dokumenta. Dostupni su sistemu putem nekog od standardnih komunikacionih protokola. Čvor teksta koji predstavlja njihov položaj u dokumentu tada sadrži samo njihovu URI adresu.

Tipovi čvorova koji se javljaju u drugim modelima XML dokumenata, a u XMIRS modelu su izostavljeni, predstavljaju određene delove XML dokumenta koje XMIRS ignoriše, ali ih ne uklanja iz originalnog sadržaja dokumenta. U pitanju su sledeći tipovi čvorova: čvor instrukcije za procesiranje (*processing instruction node*), čvor komentara (*comment node*), čvor DTD deklaracije (*DTD declaration node*), čvor neparsiranog eksternog entiteta (*unparsed external entity node*) i čvor notacije (*notation node*).

Prikazani model predstavlja dokumente kao strukture tipa stabla. Definisanje drugih veza među čvorovima dokumenta (ili među čvorovima različitih dokumenata) moguće je korišćenjem npr. ID/IDREF atributa [37] ili XLink [65] linkovima, ali je interpretacija takvih veza isključivo u domenu aplikacija koje koriste date dokumente ili specijalizovanih modula za pretraživanje.

2.2.2 Identifikacija dokumenata

Za dokumente koji se skladište u okviru XMIRS-a mora postojati i njihova XML Schema specifikacija tipa. Specifikacije tipova dokumenata kojima rukuje XMIRS dostupni su putem URI identifikatora koji imaju sledeći šablon:

```
http://<server>[:<port>]/schemas/<schema-name>
```

gde je *schema-name* naziv datog tipa dokumenta.

Dokumenti skladišteni u okviru XMIRS-a dostupni su putem URI identifikatora koji imaju sledeći šablon:

`http://<server>[:<port>]/docs/<schema-name>/<doc-id>[#<xpointer>]`

gde je *schema-name* naziv tipa kome dati dokument pripada, *doc-id* numerički identifikator dokumenta koga samostalno dodeljuje XMIRS, a *xpointer* je opcioni XPointer [64] izraz kojim se referencira pojedini fragment dokumenta.

2.2.3 Kolekcije, moduli i indeksi

Dokumenti kojima rukuje XMIRS su validni XML dokumenti čiji tip je dat XML Schema dokumentima. Nad ukupnim skupom dokumenata kojima rukuje XMIRS može se definisati više particija. Pojam particije skupa posmatra se na isti način kao i u teoriji skupova. Elementi particije nazivaju se *kolekcije*. Particija koju čine kolekcije dokumenata istog tipa je uvek definisana. Druge particije ukupnog skupa dokumenata definiše korisnik.

Sistem XMIRS predstavlja okvir za pronalaženje dokumenata namenjen integraciji različitih modula. Svaki *modul* poseduje dve osnovne funkcije: (1) indeksiranje određenog tipa medija i (2) pretraživanje određenog tipa medija. Indeksiranje kolekcija dokumenata određeno je skupom parametara koji se naziva konfiguracija indeksa. Proces pretraživanja kolekcija koristi strukture podataka generisane tokom indeksiranja.

Svaki modul poseduje sopstveni upitni jezik kojim se izražava kriterijum pretrage. Upitni jezik modula, sa stanovišta XMIRS sistema, definiše se XML Schema dokumentom. Upiti upućeni modulu su, na taj način, XML dokumenti koji predstavljaju instance odgovarajućeg tipa.

Indeks predstavlja jednu konfiguraciju modula namenjenu pretraživanju jedne kolekcije dokumenata. Nad datom kolekcijom dokumenata može se definisati više *indeksa*. Indeksi se mogu posmatrati kao instance modula namenjeni pretraživanju kolekcije po pojedinačnim aspektima.

2.2.4 Elementarni i složeni upiti

Upit po jednom datom indeksu naziva se *elementarni upit*. Elementarni upit se formuliše kao validan XML dokument određen XML Schema definicijom vezanom za odgovarajući modul. Rezultat elementarnog upita je podskup kolekcije dokumenata koji zadovoljavaju dati upit. Stepen slaganja svakog dokumenta sa datim elementarnim upitom izražava se brojem u intervalu $[0, 1]$.

Složeni upiti su sastavljeni iz više elementarnih upita. Semantika kombinovanja elementarnih upita i izračunavanje ukupnog rezultata vezani su za konkretan model pronalaženja dokumenata. Sistem XMIRS omogućava korišćenje različitih modela pronalaženja koji se mogu dodavati sistemu kao posebne softverske komponente.

2.3 Modeli pronalaženja dokumenata

Ovaj odeljak definiše tri modela pronalaženja koji se mogu upotrebiti za prethodno opisani model dokumenata. Na početku su date definicije pojmova koje su zajedničke svim modelima.

Definicija 2.1 *Kolekcija dokumenata koji učestvuju u pretraživanju data je skupom* $D = \{d_i | i = 1, \ldots, N\}$. *Skup definisanih indeksa nad ovom kolekcijom je* $I = \{m_i | i = 1, \ldots, M\}$.

Definicija 2.2 *Elementarni upit* q_i *je onaj upit za koga se može izračunati rezultat korišćenjem tačno jednog indeksa* m_i. *Svaki indeks* m_i *izražava sličnost* j*-tog dokumenta* $d_j \in D$ *sa datim elementarnim upitom* q_i *pomoću vrednosti* $s_{ji} \in [0, 1]$. *Uređeni par* $h_{ji} = (d_j, s_{ji})$ *naziva se elementarni pogodak pretrage. Rezultat pretrage po indeksu* m_i *za elementarni upit* q_i *je skup elementarnih pogodaka* H_i.

Definicija 2.3 *Funkcija* doc *preslikava skup elementarnih pogodaka na skup dokumenata tako da je* $\mathrm{doc}(h_{ji}) = d_j$. *Skup pronađenih dokumenata za elementarni upit* q_i *je tada* $D_i = \{d_j | \, \mathrm{doc}(h_{ji}) = d_j \wedge h_{ji} \in H_i\}$.

2.3.1 Modifikacija vektorskog modela

Model prikazan u ovom odeljku nastao je na osnovu klasičnog vektorskog modela [241, 23] namenjenog pronalaženju nestrukturiranih tekstualnih dokumenata. Modifikacija klasičnog modela takođe tretira dokumente kao tačke vektorskog prostora ali su dimenzije vektorskog prostora drugačije prirode. U nastavku je data formalna definicija ovog modela.

Definicija 2.4 *Složeni upit* Q *predstavlja uređenu trojku* $Q = (q, f, k)$ *gde je*

- $q = (q_1, q_2, \ldots, q_n)$ *uređena n-torka elementarnih upita po indeksima* $m_1, m_2, \ldots, m_n$,
- *funkcija* $f : R^n \to R^+ \cup \{0\}$ *metrika u prostoru* R^n *i*
- k *maksimalan broj pronađenih dokumenata koje će rezultat upita sadržati.*

Definicija 2.5 *Pogodak pretrage za složeni upit Q je uređeni par (d_j, s_j) gde je $d_j \in D$ pronađeni dokument, a $s_j = (s_{j1}, s_{j2}, \ldots, s_{jn})$ uređena n-torka čije komponente s_{ji} predstavljaju sličnost dokumenta d_j sa elementarnim upitom q_i, tj. $h_{ji} = (d_j, s_{ji})$.*

Rezultat pretraživanja za složeni upit Q je skup pogodaka H_Q. Ukoliko je broj pronađenih pogodaka veći od k, u rezultat ulazi prvih k pogodaka koji su najbliži upitu, pri čemu je rastojanje dato metrikom f.

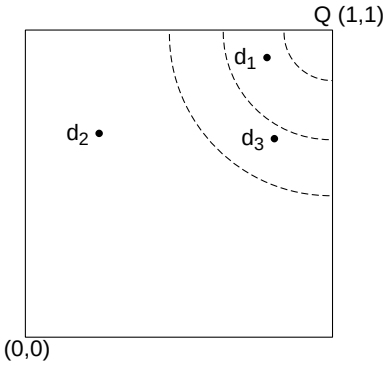
Složeni upit Q tretira se kao dokument-uzorak sa kojim se porede svi dokumenti u kolekciji. Sličnost dokumenta $d_j \in D$ sa složenim upitom Q izražava se pomoću rastojanja tačke $s_j = (s_{j1}, s_{j2}, \ldots, s_{jn}) \in [0,1]^n$ koja reprezentuje dokument d_j u vektorskom prostoru R^n i tačke koja odgovara upitu. Pretpostavka je da je sličnost dokumenta-uzorka sa samim sobom maksimalna, pa se složeni upit uvek predstavlja tačkom $Q(1, 1, \ldots, 1)$ u okviru prostora R^n. Kako za sve mere sličnosti sa elementarnim upitima važi $s_{ji} \in [0, 1]$ sledi da se sve posmatrane tačke nalaze u jediničnoj kocki prostora R^n, pri čemu je zadatak izračunavanja upita određivanje k najbližih dokumenata upitu Q. Slika 2.1 prikazuje ovu situaciju za prostor R^2, odnosno situaciju kada je postavljen složen upit koji se sastoji iz dva elementarna upita.

Izbor pogodne metrike f u prostoru R^n, za primene koju su ovde od interesa, najčešće se svodi na izbor neke od L_p vektorskih normi ($1 < p < \infty$):

$$L_p(x, y) = \left(\sum_{i=1}^{n} (x_i - y_i)^p \right)^{\frac{1}{p}} \tag{2.1}$$

Najčešće korišćene metrike dobijaju se za vrednosti parametra $p = 1$, $p = 2$ i $p = \infty$.

$$L_1(x, y) \quad = \quad \sum_{i=1}^{n} |x_i - y_i| \tag{2.2}$$

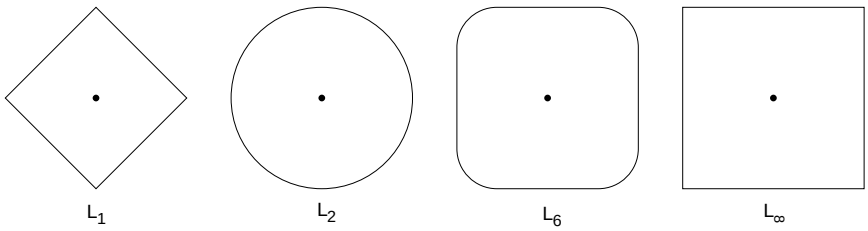


Slika 2.1: Sličnost dokumenata sa upitom u prostoru R^2

$$L_2(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.3}$$

$$L_\infty(x,y) = \max_i\{|x_i - y_i|\} \tag{2.4}$$

Neke od L_p normi, primenjene na prostor R^2, ilustrovane su na slici 2.2. Osnovna mana ove familije metrika, tendencija da dimenzija sa najvećim opsegom vrednosti dominira nad ostalima [130], ovde nije prisutna jer je izvršena normalizacija svih dimenzija, tj. sve posmatrane tačke nalaze se u jediničnoj kocki.



Slika 2.2: Skup tačaka na istom rastojanju od centralne tačke za različite L_p norme

Sa stanovišta pretraživanja, tj. izračunavanja ukupnog skupa pogodaka H_Q, potrebno je uzeti u obzir i računsku složenost pojedinih metrika. Princip postepenog sužavanja skupa pogodaka primenjen u sistemu GEMINI [6, 80, 79] zasniva se na sukcesivnoj upotrebi različitih metrika. Problem izbacivanja iz

skupa pogodaka tačke koja bi, upotrebom sledeće metrike, bila svrstana u skup pogodaka (*false dismissal*) izbegava se biranjem metrika koje zadovoljavaju tzv. *lower bounding* uslov (v. odeljak 1.4.1). Ovakav pristup omogućava korišćenje metrike manje računske složenošti za eliminaciju najvećeg broja tačaka, dok se preostali skup tačaka filtrira metrikom koja ima veću računsku složenost ali i veću preciznost odabira tačaka. Jedna moguća kombinacija ovakvih metrika su L_∞ (kao prva upotrebljena) i L_2 (kao druga).

Važnu razliku između ovde prikazanog modela i klasičnog vektorskog modela čine: (1) izbor vektorskog prostora u kome se vrši pretraživanje i (2) izbor funkcije za izračunavanje sličnosti dokumenata sa upitom. Dimenzija vektorskog prostora u klasičnom vektorskom modelu određena je brojem izraza indeksa i samim tim je fiksna (v. odeljak 1.1.1). Za velike kolekcije dokumenata različite tematike, broj izraza indeksa može dostići red veličine 10.000. Sa druge strane, dimenzionalnost prostora u modifikovanom modelu zavisi od broja elementarnih upita iz kojih se sastoji dati složeni upit. S obzirom da upite formuliše korisnik, može se očekivati da dimenzionalnost prostora bude reda veličine 10 (imajući u vidu perceptivna ograničenja čoveka). Pored toga, pojedine dimenzije vektorskog prostora modifikovanog modela zavise od elemenata konkretnog postavljenog upita, pa je prethodna izgradnja SAM struktura za efikasnije izračunavanje rezultata (v. odeljak 1.4.1) nepraktična, jer bi se morale pokriti sve kombinacije upita po definisanim indeksima.

U klasičnom modelu upit je predstavljen vektorom čija vrednost direktno zavisi od sadržaja upita, dok je vektorski prostor nepromenljiv. Sličnost dokumenata u kolekciji i upita izračunava se prema jednačini 1.2. U modifikovanom modelu upit se smatra idealnim pogotkom predstavljenim tačkom $Q(1, 1, \ldots, 1)$. Sličnost dokumenata sa upitom izačunava se pomoću neke od metrika definisanim nad prostorom R^n. Pored toga, moguća je i sukcesivna upotreba više metrika radi povećanja efikasnosti procesa pretraživanja.

Relevance feedback

Varijanta modifikovanog vektorskog modela koja omogućava *relevance feedback* može se dobiti izmenom korišćenih metrika tako da se svakom elementarnom upitu q_i dodeljuje težina $w_i \in [0, 1]$ koja predstavlja nivo relevantnosti i-tog elementarnog upita u ukupnom rezultatu. Metrike iz familije p-normi u ovoj varijanti imaju oblik:

$$L_p(x, y) = \left(\sum_{i=1}^{n} w_i (x_i - y_i) \right)^{\frac{1}{p}} \tag{2.5}$$

Inicijalna vrednost svih težina w_i je jednaka 1. Nakon postavljenog upita korisnik ima priliku da pregleda rezultat upita koga čine najbližih k pogodaka. Za svaki od k pogodaka korisnik označava da li je relevantan ili nije. Iterativnom promenom vrednosti težina w_i korišćene metrike koriguju se rezultati pretraživanja. U tom slučaju posmatrajmo sledeće veličine:

- D_r: skup relevantnih dokumenata koji je odredio korisnik među *pronađenim* dokumentima,
- D_n: skup ne-relevantnih dokumenata među *pronađenim* dokumentima,
- $|D_r|$ i $|D_n|$: broj elemenata u skupovima D_r i D_n, tim redosledom,
- α, β, γ: konstante za podešavanje.

Na osnovu definisanih skupova D_r i D_n težine w_i se mogu korigovati na sledeći način:

$$w_i' = \alpha w_i + \frac{\beta}{|D_r|} \sum_{\forall d_j \in D_r} (1 - s_{ji}) + \frac{\gamma}{|D_n|} \sum_{\forall d_j \in D_n} (1 - s_{ji}) \tag{2.6}$$

Jednačina 2.6 zasniva se na jednačinama iz [232, 125, 23] za klasičan vektorski model. Inicijalna vrednost konstante α je 1. Uobičajeno je da se pretpostavlja da su informacije sadržane u relevantnim dokumentima važnije od informacija sadržanih u ne-relevantnim, pa je vrednost konstante γ manja po apsolutnoj vrednosti od konstante β. Za $\gamma = 0$ uzimaju se u obzir samo relevantni dokumenti. Formulacija kriterijuma optimalnosti za ovakav RF proces ne postoji (v. odeljak 1.1.4), ali su eksperimenti primenjeni na sisteme sa nestrukturiranim tekstualnim dokumentima pokazali dobre rezultate [23]. Nakon izračunavanja novih vrednosti težina w_i vrši se njihova normalizacija za interval $[0, 1]$. Razlika u odnosu na klasični vektorski model prilikom implementacije RF ciklusa je u tome što se u klasičnom modelu vrši reformulacija upita, tj. tačka koja predstavlja upit se pomera u fiksnom prostoru dokumenata, dok se ovde ne modifikuje upit (koji uvek ostaje u temenu kocke) nego upotrebljena metrika.

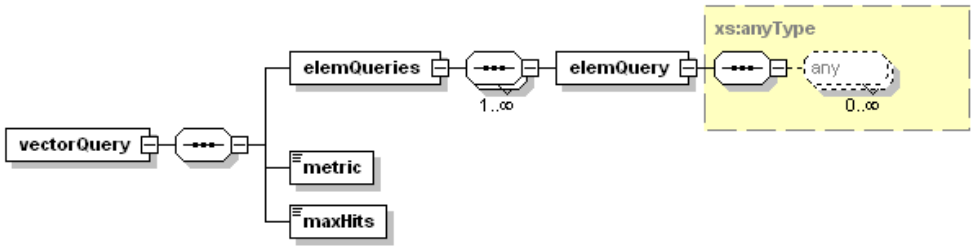
XML Schema dokument kojim se definiše format upita dat je listingom 2.1. Slika 2.3 ilustruje strukturu ovog dokumenta. XML Schema dokument kojim se definiše format rezultata dat je listingom 2.2.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="vectorQuery">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="elemQueries">
          <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
              <xs:element name="elemQuery">
                <xs:complexType>
                  <xs:complexContent>
                    <xs:extension base="xs:anyType">
                      <xs:attribute name="index" type="xs:token" use="required"/>
                    </xs:extension>
                  </xs:complexContent>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="metric" type="xs:token"/>
        <xs:element name="maxHits" type="xs:integer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.1: Šema upita za modifikovani vektorski model

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="vectorResult">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="hit">
          <xs:complexType>
            <xs:attribute name="docid" type="xs:anyURI" use="required"/>
            <xs:attribute name="distance" type="xs:double" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.2: Šema rezultata za modifikovani vektorski model



Slika 2.3: XML Schema za upite u modifikovanom vektorskom modelu

XML Schema dokument kojim se definiše format poruke korisnika o relevantnosti pronađenih dokumenata u toku jednog *relevance feedback* ciklusa dat je listingom 2.3.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="relevanceFeedback">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="hit">
          <xs:complexType>
            <xs:attribute name="docid" type="xs:anyURI" use="required"/>
            <xs:attribute name="relevant" use="required">
              <xs:simpleType>
                <xs:restriction base="xs:token">
                  <xs:pattern value="yes|no"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.3: Šema poruke o relevantnosti dokumenata

2.3.2 Modifikacija proširenog bulovskog modela

Prethodno definisani model pronalaženja dokumenata omogućava postavljanje složenih upita koji se sastoje iz niza elementarnih upita. Međusobna

veza elementarnih upita u takvom modelu implicitno sadrži semantiku konjunkcije, tj. pretraživanje u takvom modelu odnosi se na dokumente koji u dovoljnoj meri zadovoljavaju *sve* zadate elementarne upite.

Klasični bulovski model definisan za nestrukturirane tekstualne dokumente (v. odeljak 1.1.1) podrazumeva binarne težine $w_{ij} \in \{0, 1\}$ dodeljene dokumentu d_i koje označavaju prisustvo izraza indeksa k_j u njemu. Primena ovakvog modela u slučaju multimedijalnih dokumenata nije izvodljiva jer se slaganje dokumenta sa upitom ne može izraziti binarnim rezultatom. Prošireni bulovski model [243, 23] dozvoljava korišćenje težina iz intervala $[0, 1]$ i iskorišćen je kao osnova za definisanje novog modela.

Definicija 2.6 *Upit Q u modifikovanom bulovskom modelu ima sledeće osobine:*

- *elementarni upit q je upit,*
- *konjunkcija dva upita, u oznaci $Q_1 \,\text{AND}^{(k)}\, Q_2$, je upit,*
- *disjunkcija dva upita, u oznaci $Q_1 \,\text{OR}^{(k)}\, Q_2$, je upit i*
- *razlika dva upita, u oznaci $Q_1 \,\text{NOT}^{(k)}\, Q_2$, je upit.*

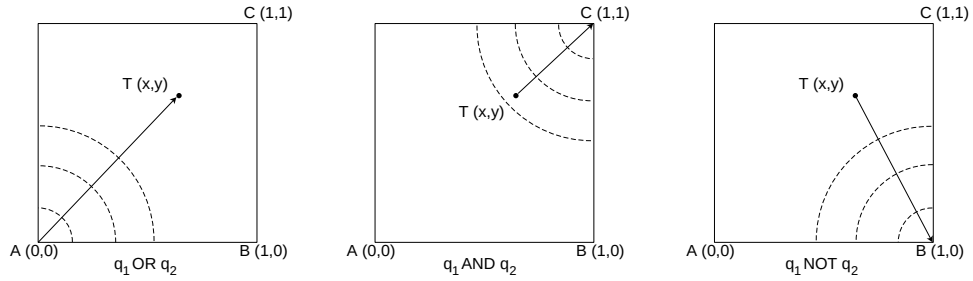
Parametar k predstavlja maksimalan broj najboljih pogodaka koji će se naći u rezultujućem skupu.

Potrebno je istaći da je u prethodnoj definiciji operator $\text{NOT}^{(k)}$ definisan kao binarni operator sa značenjem razlike skupova, što se razlikuje od njegove uobičajene definicije gde ima značenje komplementa skupa.

Interpretacija elementarnih upita je u nadležnosti odgovarajućeg indeksa, odnosno korišćenog modula za pretraživanje. Interpretacija upita sa operatorima, tipa $q_1 \,\text{AND}^{(k)}\, q_2$, $q_1 \,\text{OR}^{(k)}\, q_2$ ili $q_1 \,\text{NOT}^{(k)}\, q_2$ ilustrovana je na slici 2.4. Za upit tipa $q_1 \,\text{AND}^{(k)}\, q_2$ najpoželjnija je situacija kada je sličnost posmatranog dokumenta sa oba elementarna upita q_1 i q_2 maksimalna, tj. izražena vrednošću $s_1 = s_2 = 1$. Posmatrano u prostoru R^2, najpoželjnija tačka za dati dokument je tačka $C(1, 1)$. Mera sličnosti dokumenta sa ovakvim upitom izražava se kao komplement rastojanja tačke $T(x, y)$ pridružene dokumentu od tačke C.

Analogno prethodnom slučaju mogu se definisati mere sličnosti za preostala dva tipa upita. Upit tipa $q_1 \,\text{OR}^{(k)}\, q_2$ kao najnepoželjniju tačku ima tačku $A(0, 0)$ pa se sličnost dokumenta sa upitom u ovom slučaju izražava kao rastojanje tačke $T(x, y)$ pridružene dokumentu od tačke A. Upit tipa $q_1 \,\text{NOT}^{(k)}\, q_2$

kao najpoželjniju tačku ima tačku $B(1,0)$ pa se sličnost dokumenta sa upitom izražava kao komplement rastojanja tačaka T i B.



Slika 2.4: Proširena bulovska logika posmatrajući upit sa dva elementarna upita

Prethodno opisano izračunavanje upita može se uopštiti za slučaj n-D prostora R^n. Pri tome karakteristične tačke A, B i C imaju odgovarajuće vrednosti pobrojane u tabeli 2.1.

Izraz	Karakteristična tačka
q_1 AND$^{(k)}$ q_2 AND$^{(k)}$ $\ldots$ q_n	C(1,1,. . . ,1)
q_1 OR$^{(k)}$ q_2 OR$^{(k)}$ $\ldots$ q_n	A(0,0,. . . ,0)
q_1 NOT$^{(k)}$ q_2 NOT$^{(k)}$ $\ldots$ q_n	B(1,0,. . . ,0)

Tabela 2.1: Vrste upita i karakteristične tačke

Za određivanje rastojanja može se upotrebiti metrika iz L_p familije, i to njena normalizovana varijanta u kojoj maksimalno rastojanje dve tačke u kocki iznosi 1. Ova varijanta data je jednačinom 2.7.

$$L_p(x,y) = \left(\frac{\sum_{i=1}^{n} (x_i - y_i)^p}{n} \right)^{\frac{1}{p}} \tag{2.7}$$

Izrazi za izračunavanje sličnosti će, za upotrebljenu metriku L_p, biti definisani sledećim jednačinama.

$$sim(q_{and}, d) = 1 - L_p(T, C) \tag{2.8}$$

$$sim(q_{or}, d) = L_p(T, A) \tag{2.9}$$

$$sim(q_{not}, d) = 1 - L_p(T, B) \tag{2.10}$$

Sva tri operatora imaju jednak prioritet prilikom interpretiranja upita. Tako je upit $q_1 \text{ OR}^{(k_1)} q_2 \text{ AND}^{(k_2)} q_3$ ekvivalentan upitu $(q_1 \text{ OR}^{(k_1)} q_2) \text{ AND}^{(k_2)} q_3$. Sada se može dati i formalna definicija rezultata izvršavanja upita.

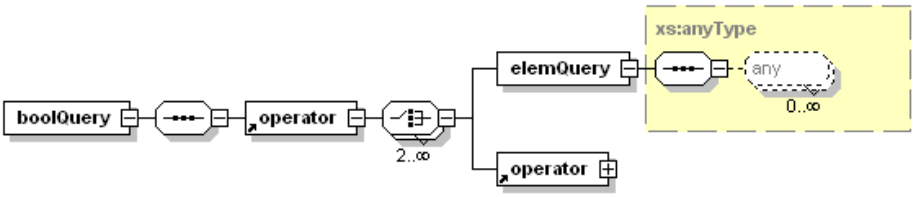
Definicija 2.7 *Rezultat pretraživanja za upit Q je skup pogodaka $H_Q = \{(d, s) | d \in \bigcup_{i=1,\ldots,n}, s \in [0, 1]\}$, gde je*

- *d pronađeni dokument i*
- *s sličnost pronađenog dokumenta sa upitom.*

U trivijalnom slučaju kada upit Q sadrži samo jedan elementarni upit rezultujući skup pogodaka generiše odgovarajući indeks. U slučaju kada Q sadrži i operatore, interpretacija upita obuhvata i sukcesivno izvršavanje (za svaki operator) operacije izbora najbližih k pogodaka u odnosu na karakterističnu tačku.

Kao i kod modifikovanog vektorskog modela, i ovde dimenzionalnost prostora zavisi od broja elementarnih upita koji se nalaze u ukupnom upitu. Slično prethodnom modelu, i ovde je dimenzionalnost prostora relativno mala. Pored toga, i u ovom modelu je moguće sukcesivno korišćenje više metrika prilikom izračunavanja skupova pogodaka za pojedine operatore radi povećanja efikasnosti izračunavanja. Ovaj model omogućava i korišćenje različitih metrika za pojedine operatore u istom upitu. Upit $q_1 \text{ AND}_2^{(k_1)} q_2 \text{ OR}_\infty^{(k_2)} q3$ predstavlja primer takvog upita u kome prvi operator koristi metriku L_2, a drugi metriku L_∞. Za sada nije poznato da li ova mogućnost ima praktičnog značaja.

XML Schema dokument kojim se definiše format upita dat je listingom 2.4. Slika 2.5 ilustruje strukturu ovog dokumenta. Dokument kojim se predstavlja rezultat ima identičnu strukturu kao i za modifikovani vektorski model.



Slika 2.5: Šema upita za modifikovani bulovski model

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="operator">
    <xs:complexType>
      <xs:choice minOccurs="2" maxOccurs="unbounded">
        <xs:element name="elemQuery">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="xs:anyType">
                <xs:attribute name="index" type="xs:token" use="required"/>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
        <xs:element ref="operator"/>
      </xs:choice>
      <xs:attribute name="type" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="AND"/>
            <xs:enumeration value="OR"/>
            <xs:enumeration value="NOT"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="metric" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:token">
            <xs:enumeration value="L_1"/>
            <xs:enumeration value="L_2"/>
            <xs:enumeration value="L_inf"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="maxHits" type="xs:integer" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="boolQuery">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="operator"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.4: Šema upita za modifikovani bulovski model

2.3.3 Model sličnih klastera

Prethodno prikazani modeli predstavljaju rezultat prilagođavanja postojećih modela pronalaženja ambijentu multimedijalnih dokumenata. Za njih važe sledeće osobine:

1. prostor u kome se vrši pretraživanje zavisi od postavljenog upita,
2. reformulacija upita kroz RF interakciju sa korisnikom ne obuhvata proširenje upita novim elementima (*query expansion*), već samo promenu težina dodeljenih postojećim elementima (*reweighting*) i
3. informacije o pronađenim dokumentima iz prethodnih sesija se ne koriste u toku pretraživanja.

Model definisan u ovom odeljku omogućava proširivanje rezultata pretrage pogocima iz ranijih sesija pretraživanja. Iako se postavljeni upit formalno ne modifikuje, proširivanje rezultata se može shvatiti kao implicitna modifikacija upita. Modifikacija upita u tom slučaju obuhvata i promenu težina i proširivanje upita. Prilikom formiranja modela usvojene su dve pretpostavke.

Klaster hipoteza. Istraživanja u oblasti IR definišu klaster hipotezu kao „težnju međusobno sličnih dokumenata da budu relevatni za iste upite“ [289, 91, 130]. Posledica ove pretpostavke je da će dokumenti koji su najbolje rangirani za dati upit formirati poseban klaster imajući u vidu njihov položaj u posmatranom vektorskom prostoru. Ovom prilikom ćemo proširiti ovaj rezon i na dokumente koji se prilikom pretraživanja nisu našli među najbolje rangiranim. Takvi dokumenti mogu biti među najbolje rangiranim za neki drugi upit i tako opet formirati klaster. Klaster slabije rangiranih dokumenata za jedan upit na taj način može biti vrlo sličan klasteru najbolje rangiranih dokumenata za drugi upit. Poređenjem sadržaja ovih klastera može se jedan klaster proširiti elementima drugog klastera radi proširivanja rezultata pretrage.

Sferni oblik klastera. U IR primenama najčešće se pretpostavlja sferni oblik klastera [83, 130, 153]. Ova pretpostavka nameće upotrebu algoritama za klasterovanje koji favorizuju sferni oblik klastera kao što su *k-means* ili hijerarhijski *complete-link* algoritam. Ispravnost ove pretpostavke potvrđivana je eksperimentalno (npr. [153, 120]) za kolekcije tekstualnih dokumenata.

Navedene pretpostavke će poslužiti kao osnova za formiranje modela koji omogućava proširivanje skupa pogodaka podeljenog na klastere dokumentima koji se nalaze u sličnim klasterima određenim tokom ranijih sesija pretraživanja.

Definicija 2.8 *Složeni upit Q je uređena n-torka $Q = (q_1, q_2, \ldots, q_n)$ gde je q_i elementarni upit po indeksu m_i.*

Složeni upit dat definicijom 2.8 nalik je složenom upitu definisanom u okviru modifikovanog vektorskog modela (v. odeljak 2.3.1). Na isti način kao i u modifikovanom vektorskom modelu, i u ovom slučaju elementarni upiti koji čine složeni upit određuju karakteristike vektorskog prostora u kome se vrši pretraživanje. Za razliku od prethodnog modela, ovde se uvodi i pojam razlike dva složena upita.

Definicija 2.9 *Razlika dva složena upita, u oznaci $r(Q_1, Q_2)$, je minimalan broj elementarnih operacija kojima se skup korišćenih indeksa jednog upita može transformisati u skup korišćenih indeksa drugog upita. Elementarne operacije nad skupovima korišćenih indeksa su*

- *dodavanje jednog elementa u skup,*
- *uklanjanje jednog elementa iz skupa i*
- *zamena jednog elementa skupa drugim elementom koji ne pripada datom skupu.*

Uvođenje pojma razlike složenih upita omogućava njihovo međusobno poređenje. Mera sličnosti dva upita data je njihovom razlikom.

Definicija 2.10 *Pogodak pretrage za složeni upit Q je uređeni par $h_j = (d_j, s_j)$ gde je $d_j \in D$ pronađeni dokument, a $s_j = (s_{j1}, s_{j2}, \ldots, s_{jn})$ uređena n-torka čije komponente s_{ji} predstavljaju sličnost dokumenta d_j sa elementarnim upitom q_i, tj. $h_{ji} = (d_j, s_{ji})$.*

Rezultat pretraživanja za složeni upit Q je skup pogodaka H_Q i particija ovog skupa $P_{H_Q} = \{C_1, C_2, \ldots, C_k\}$ koju čine klasteri dokumenata.

Klasteri dokumenata nad skupom pogodaka H_Q određuju se pomoću hijerarhijskog aglomerativnog *complete-link* algoritma [130, 153, 83]. Međusobno rastojanje dva klastera, u oznaci $c(C_1, C_2)$, dato je sledećom jednačinom.

$$c(C_1, C_2) = \begin{cases} L_2(h_1, h_2), & |C_1| = 1 \wedge |C_2| = 1 \\ \max\limits_{i,j}\{L_2(h_{1i}, h_{2j})\}, & |C_1| > 1 \vee |C_2| > 1 \end{cases} \tag{2.11}$$

U jednačini 2.11 L_2 je metrika koja predstavlja euklidsko rastojanje, h_1 i h_{1i} su pogoci u klasteru C_1, a h_2 i h_{2j} su pogoci u klasteru C_2. Izabrani algoritam formira klastere nad skupom H_Q na sledeći način:

1. Inicijalni skup klastera čine klasteri koji sadrže tačno po jedan element skupa.
2. Od dva najbliža klastera formira se unija – klaster koji obuhvata sve elemente oba klastera.
3. Ako se svi klasteri nalaze na međusobnom rastojanju većem od unapred date vrednosti ε postupak se završava; u suprotnom se prelazi na korak 2.

Rezultat pretraživanja za složeni upit se trajno čuva u sistemu zajedno sa specifikacijom upita. Proširivanje skupa pogodaka rezultatima upita iz prethodnih sesija je iterativan postupak. U i-toj iteraciji posmatraju se tekući rezultat pretrage (za upit Q) i skup prethodno postavljanih upita $P_i = \{Q_{ij} | r(Q, Q_{ij}) = i\}$ (početna iteracija ima indeks 0). U svakoj iteraciji korisnik posmatra rezultat tekućeg upita i može ga modifikovati na dva načina: (1) dopunjavanjem pogocima prethodnih upita ili (2) izbacivanjem nekog od postojećih pogodaka. Ove operacije podrazumevaju da je korisnik u mogućnosti da pregleda sadržaj željenog dokumenta u rezultatu kako bi utvrdio njegovu relevantnost. Za svaki dokument d_i u okviru klastera C_j tekućeg rezultata korisnik može da zatraži dopunjavanje klastera C_j sadržajima klastera za upite iz skupa P_{ij} koji sadrže dokument d_i.

Kada korisnik proceni da je formirao optimalan rezultat pretrage prekida se iterativan postupak i rezultat upita Q koji sadrži modifikovane klastere se trajno čuva u okviru sistema. Pored toga, neki od klastera iz prethodnih sesija se može ukloniti čime se on proglašava jednakim tekućem klasteru.

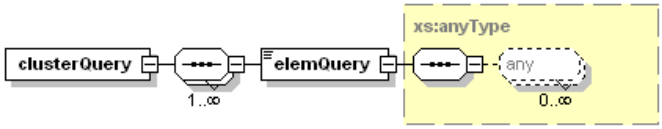
Osnovni cilj ovog postupka je da tokom vremena omogući formiranje klastera koji obuhvataju slične dokumente, pri čemu je međusobna sličnost dokumenata rezultat subjektivne procene korisnika. Na ovaj način moguće je kombinovanje automatizovane pretrage dokumenata u datom vektorskom prostoru (inicijalno formiranje rezultata) sa subjektivnim informacijama o relevantnosti dokumenata. Princip proširivanja skupa pogodaka rezultatima ranijih sesija pronalaženja ne zavisi od načina formiranja klastera tako da se može izabrati onaj koji eksperimentalno pokaže najbolje rezultate.

XML Schema dokument kojim se definiše format upita dat je listingom 2.5. Slika 2.6 ilustruje strukturu ovog dokumenta.

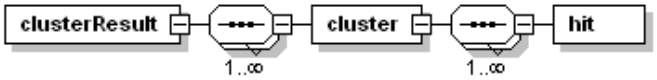
```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="clusterQuery">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="elemQuery">
          <xs:complexType>
            <xs:complexContent>
              <xs:extension base="xs:anyType">
                <xs:attribute name="index" type="xs:token" use="required"/>
              </xs:extension>
            </xs:complexContent>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.5: Šema upita za model sličnih klastera

XML Schema dokument kojim se definiše format rezultata dat je listingom 2.6. Slika 2.7 ilustruje strukturu ovog dokumenta.



Slika 2.6: XML Schema za upite u modelu sličnih klastera



Slika 2.7: XML Schema za rezultat u modelu sličnih klastera

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="clusterResult">
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="cluster">
          <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
              <xs:element name="hit">
                <xs:complexType>
                  <xs:attribute name="docid" type="xs:anyURI" use="required"/>
                  <xs:attribute name="distance" type="xs:double" use="required"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="id" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.6: Šema rezultata za model sličnih klastera

XML Schema dokument kojim se definiše format zahteva korisnika za dobijanjem sadržaja svih klastera koji se od posmatranog klastera razlikuju za datu vrednost dat je na listingom 2.7.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="clusterHistory">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="getSimilar">
          <xs:complexType>
            <xs:attribute name="clusterID" type="xs:integer" use="required"/>
            <xs:attribute name="level" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
```

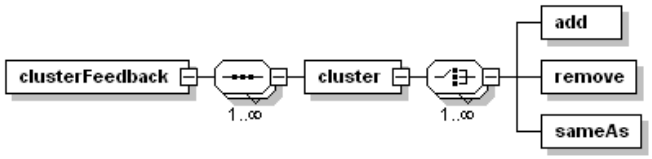
```
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.7: Šema zahteva za dobijanje sadržaja sličnih klastera

XML Schema dokument kojim se definiše format poruke korisnika o modifikacijama dobijenih klastera u toku jedne iteracije dat je listingom 2.8. Slika 2.8 ilustruje strukturu ovog dokumenta.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="clusterFeedback">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="cluster">
          <xs:complexType>
            <xs:choice maxOccurs="unbounded">
              <xs:element name="add">
                <xs:complexType>
                  <xs:attribute name="docid" type="xs:anyURI" use="required"/>
                </xs:complexType>
              </xs:element>
              <xs:element name="remove">
                <xs:complexType>
                  <xs:attribute name="docid" type="xs:anyURI" use="required"/>
                </xs:complexType>
              </xs:element>
              <xs:element name="sameAs">
                <xs:complexType>
                  <xs:attribute name="clusterID" type="xs:integer" use="required"/>
                </xs:complexType>
              </xs:element>
            </xs:choice>
            <xs:attribute name="id" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.8: Šema poruke o modifikacijama klastera



Slika 2.8: Šema poruke o modifikacijama klastera

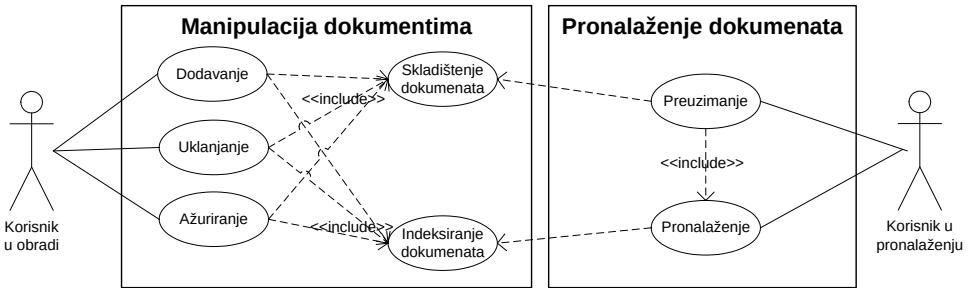
2.4 Softverska arhitektura

Sistem XMIRS sastoji se od nekoliko osnovnih elemenata za realizaciju četiri osnovne funkcije sistema: skladištenje, rukovanje, indeksiranje i pretraživanje dokumenata. Na osnovu toga celokupan sistem je dekomponovan na sledeće podsisteme:

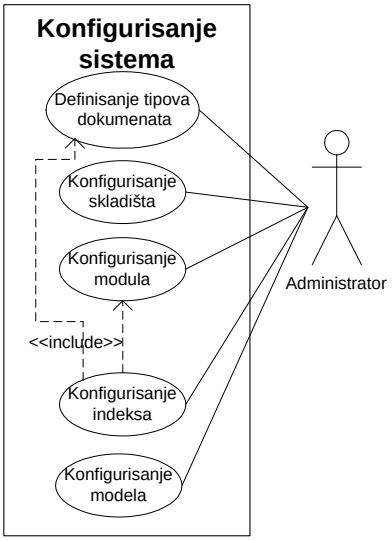
1. *skladištenje dokumenata*: obezbeđuje skladištenje celokupnih dokumenata ili pojedinih njihovih elemenata,
2. *rukovanje dokumentima*: implementira operacije dodavanja, uklanjanja i modifikacije dokumenata i njihovih šema; oslanja se na podsistem za skladištenje dokumenata,
3. *indeksiranje dokumenata*: implementira proces indeksiranja dokumenata, uz oslonac na instalirane module za indeksiranje i
4. *pronalaženje dokumenata*: omogućava pronalaženje dokumenata, koristeći različite modele pretraživanja i instalirane module.

Sistem XMIRS razlikuje tri tipa učesnika: korisnike u pretraživanju, korisnike u obradi dokumenata i administratore. Korisnici u obradi imaju mogućnost manipulacije kolekcijama dokumenata, korisnici u pretraživanju imaju mogućnost pronalaženja dokumenata, dok je zadatak administratora konfigurisanje sistema. Slika 2.9 predstavlja dijagram slučajeva korišćenja za korisnike u obradi i korisnike u pronalaženju dokumenata.

Manipulacija dokumentima, sa stanovišta korisnika u obradi, obuhvata funkcije *Dodavanje*, *Uklanjanje* i *Ažuriranje* koje se odvijaju nad skupom dokumenata kojima raspolaže sistem. Ove tri funkcije oslanjaju se na funkcije *Skladištenje dokumenata* i *Indeksiranje dokumenata*. Njihova implementacija obavezno obuhvata i upotrebu ovih funkcija.



Slika 2.9: Funkcije sistema namenjene korisnicima u obradi i pronalaženju



Slika 2.10: Funkcije sistema namenjene administratorima

Funkcija *Preuzimanja* dokumenata obezbeđuje korisniku pristup dokumentima skladištenim u okviru XMIRS sistema. *Preuzimanje* dokumenata zahteva poznavanje identifikatora dokumenata, pa je prethodno potrebno odrediti date identifikatore kao rezultat funkcije *Pronalaženja* dokumenata.

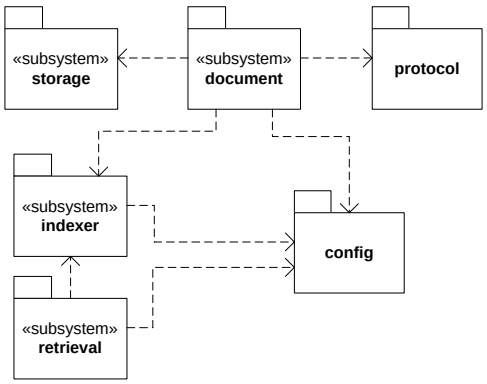
Zadatak administratora predstavlja konfigurisanje XMIRS sistema. Konfigurisanje se odnosi na formiranje funkcionalnog sistema za pronalaženje dokumenata prilagođenog konkretnoj primeni. Ovakav sistem sačinjen je od XMIRS jezgra i modula za pretraživanje. Konfigurisanje sistema sastoji se iz funkcija prikazanih dijagramom slučajeva korišćenja na slici 2.10.

Definisanje tipova dokumenata (datih XML Schema dokumentima) sastoji se u registrovanju datih tipova i skladištenju njihovih specifikacija. *Konfigurisanje skladišta* obuhvata definisanje parametara podsistema za skladištenje dokumenata. *Konfigurisanje modula* obuhvata instalaciju modula sistema i definisanje njihovih parametara. *Konfiguracija indeksa* sadrži definicije indeksa koji će biti formirani od strane pojedinih modula nad odgovarajućim delovima dokumenata. Modeli pronalaženja dokumenata mogu imati sopstvene parametre koji se određuju u okviru *Konfigurisanja modela*.

Međusobne veze zavisnosti između pojedinih XMIRS podsistema i njihovih paketa prikazani su na slici 2.11. Podsistemi skladištenja, rukovanja dokumentima, indeksiranja i pronalaženja nalaze se u okviru paketa *storage*, *document*, *indexer* i *retrieval*, tim redosledom. Paket *config* sadrži klase koje predstavljaju opis tekuće konfiguracije sistema i koristi se u okviru ostalih paketa. Paket *protocol* sadrži klase koje omogućavaju isporuku dokumenata putem različitih transportnih protokola. Specifikacija navedenih podsistema data je u narednim odeljcima.

2.4.1 Konfigurisanje sistema

Proširivost sistema različitim modulima za indeksiranje i pretraživanje, kao i različitim modelima pronalaženja dokumenata zahteva njegovo konfigurisanje prilikom instalacije. Konfiguracija sistema obuhvata konfigurisanje njegovih podsistema predstavljenih u prethodnom odeljku. Proces konfigurisanja sastoji se iz više koraka, navedenih u prethodnom odeljku u okviru slučajeva korišćenja za administratore sistema.

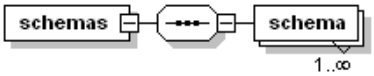


Slika 2.11: XMIRS podsistemi

Konfigurisanje tipova dokumenata

Kako je sistem sposoban da rukuje sa više tipova dokumenata datih svojim šemama, to je potrebno registrovati sve tipove dokumenata. Katalog upotrebljenih tipova dokumenata je XML dokument čija šema je data na listingu 2.9, odnosno slici 2.12. Svaki tip dokumenta definiše se elementom *schema*. Atributi ovog elementa su *name*, koji predstavlja naziv datog tipa koji se koristi interno u okviru XMIRS sistema, i *url*, URL identifikator odgovarajućeg XML Schema dokumenta. Sadržaj definicije tipa dokumenta koristi se prilikom validacije dokumenata nad kojima se obavljaju operacije dodavanja i ažuriranja.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="schemas">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="schema" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="name" type="xs:token" use="required"/>
            <xs:attribute name="url" type="xs:anyURI" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```
Listing 2.9: Šema dokumenta za konfigurisanje tipova dokumenata



Slika 2.12: Šema dokumenta za konfigurisanje tipova dokumenata

Konfigurisanje modula

Moduli se sastoje od dve celine: *indeksera* i *pretraživača*. Moduli se konfigurišu u okviru XML dokumenta čija šema je data na listingu 2.10, odnosno slici 2.13. Modul se definiše elementom *module*. Interni naziv modula u okviru XMIRS sistema dat je njegovim atributom *name*. Atributom *content* određuje se da li se modulu prilikom indeksiranja upućuje samo sadržaj elementa koji se indeksira (vrednost *inner*) ili definicija kompletnog elementa (vrednost *outer*). Parametri modula definišu se elementom *module/param*, čiji atributi su *name*, koji predstavlja naziv parametra, i *value*, koji predstavlja vrednost datog parametra. Ovako definisani parametri modula odnose se kako na indekser, tako i na pretraživač.

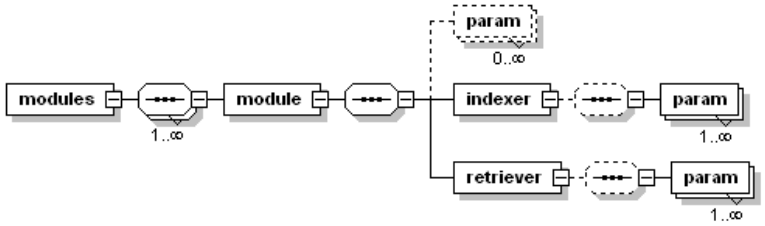
Indekser predstavlja implementaciju odgovarajućeg programskog interfejsa (v. odeljak 2.4.4). U okviru konfiguracije on se navodi elementom *indexer*, čiji parametar *class* predstavlja naziv klase koja implementira potrebni interfejs. Parametri koji se odnose samo na indeksiranje navode se u okviru elementa *indexer/param* analogno prethodnom slučaju.

Pretraživač takođe predstavlja implementaciju odgovarajućeg programskog interfejsa (v. odeljak 2.4.5). On se konfiguriše elementom *retriever* analogno konfigurisanju indeksera.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="modules">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="module">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="param" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="name" type="xs:string" use="required"/>
                  <xs:attribute name="value" type="xs:string" use="required"/>
                </xs:complexType>
              </xs:element>
```

```
            <xs:element name="indexer">
              <xs:complexType>
                <xs:sequence minOccurs="0">
                  <xs:element name="param" maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:attribute name="name" type="xs:string" use="required"/>
                      <xs:attribute name="value" type="xs:string" use="required"/>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
                <xs:attribute name="class" type="xs:string" use="required"/>
              </xs:complexType>
            </xs:element>
            <xs:element name="retriever">
              <xs:complexType>
                <xs:sequence minOccurs="0">
                  <xs:element name="param" maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:attribute name="name" type="xs:string" use="required"/>
                      <xs:attribute name="value" type="xs:string" use="required"/>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
                <xs:attribute name="class" type="xs:string" use="required"/>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="name" type="xs:token" use="required"/>
          <xs:attribute name="content" use="required">
            <xs:simpleType>
              <xs:restriction base="xs:token">
                <xs:enumeration value="inner"/>
                <xs:enumeration value="outer"/>
              </xs:restriction>
            </xs:simpleType>
          </xs:attribute>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
 </xs:element>
</xs:schema>
```

Listing 2.10: Šema dokumenta za konfigurisanje modula



Slika 2.13: Šema dokumenta za konfigurisanje modula

Konfigurisanje indeksa

Indeksi koji se kreiraju nad pojedinim elementima tipova dokumenata definišu se dokumentom čija šema je data na listingu 2.11, odnosno slici 2.14. Za dati tip dokumenta može se formirati više indeksa. Indeks predstavlja strukturu koju generiše jedan modul nad datim tipom dokumenta. Indeks može da obuhvata jedan ili više elemenata datog tipa dokumenta. Ukoliko izabrani element nije list stabla koje predstavlja strukturu tipa dokumenta, sadržaj elementa koji se prosleđuje indeksu obuhvata i sve podređene čvorove stabla. Pri tome se sadržaj elementa tretira kao serijalizovani oblik datog podstabla. Pod serijalizacijom podrazumeva se reprezentacija podstabla dokumenta pomoću XML jezika.

Indeks se definiše elementom *index* tipa *TIndex*. Atribut *name* predstavlja naziv indeksa. Atribut *module* sadrži naziv modula koji se koristi za indeksiranje. Atributom *schema* definiše se tip dokumenata nad kojim se formira indeks. Atribut *invocation* predstavlja jedan od tri načina pozivanja modula: sinhroni, asinhroni i asinhroni putem reda sa porukama. O načinima pozivanja modula biće više reči u odeljku 2.4.4.

Svaki indeks može imati svoje parametre, koje interpretira odgovarajući modul. Parametre indeksa treba razlikovati od parametara modula. Parametri indeksa definišu se elementom *param*, na isti način kao i u prethodnim situacijama.

Elementi koji ulaze u indeks navode se elementima *elements/element*. Jedini njihov atribut je *xpath*, koji predstavlja apsolutnu XPath putanju do elemenata koji ulaze u indeks.

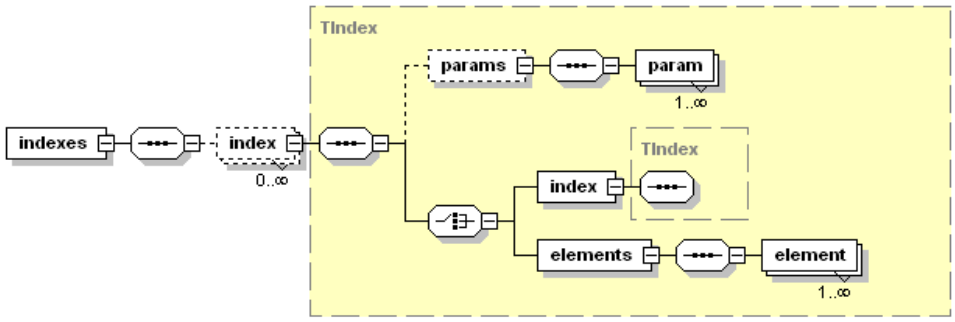
Umesto elemenata koji ulaze u sastav indeksa može se, elementom *index/index*, navesti drugi indeks koji će predstavljati izvor sadržaja za tekući

indeks. Na ovaj način omogućava se preusmeravanje izlaza koga generiše jedan indekser na ulaz drugog indeksera. Za formiranu listu povezanih indeksera pretraživanje se vrši upotrebom koncepata onog modula čiji indekser je poslednji u listi. Pri tome poslednji indekser ne mora biti u stanju da interpretira originalni sadržaj dela dokumenta za čije indeksiranje je zadužen, već se za to koristi prvi indekser u listi. Prilikom pretraživanja, koriste se samo rezultati indeksiranja poslednjeg indeksera.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="indexes">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="index" type="TIndex" minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="TIndex">
    <xs:sequence>
      <xs:element name="params" minOccurs="0">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="param" maxOccurs="unbounded">
              <xs:complexType>
                <xs:attribute name="name" type="xs:string" use="required"/>
                <xs:attribute name="value" type="xs:string" use="required"/>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:choice>
        <xs:element name="index" type="TIndex"/>
        <xs:element name="elements">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="element" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="xpath" type="xs:string"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:choice>
    </xs:sequence>
    <xs:attribute name="name" type="xs:token" use="required"/>
    <xs:attribute name="module" type="xs:token" use="required"/>
```

```
    <xs:attribute name="invocation" use="required">
      <xs:simpleType>
        <xs:restriction base="xs:token">
          <xs:enumeration value="sync"/>
          <xs:enumeration value="async"/>
          <xs:enumeration value="jms"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="schema" type="xs:token" use="required"/>
  </xs:complexType>
</xs:schema>
```

Listing 2.11: Šema dokumenta za konfigurisanje indeksa



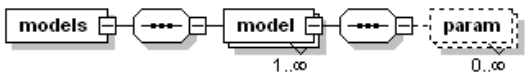
Slika 2.14: Šema dokumenta za konfigurisanje indeksa

Konfigurisanje modela

Modeli pronalaženja dokumenata namenjeni ugrađivanju u XMIRS konfigurišu se dokumentom čija struktura je data listingom 2.12, odnosno na slici 2.15. Model pronalaženja dokumenata predstavlja implementaciju odgovarajućeg programskog interfejsa. Model se, u okviru konfiguracionog dokumenta, definiše elementom *model*. Njegov atribut *name* predstavlja interni naziv modela, dok atribut *class* predstavlja naziv klase koja implementira potrebni programski interfejs. Parametri datog modela pronalaženja navode se elementom *param* na isti način kao i u prethodnim situacijama.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="models">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="model" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="param" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:attribute name="name" type="xs:string" use="required"/>
                  <xs:attribute name="value" type="xs:string" use="required"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="name" type="xs:token" use="required"/>
            <xs:attribute name="class" type="xs:string" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.12: Šema dokumenta za konfigurisanje modela pretraživanja

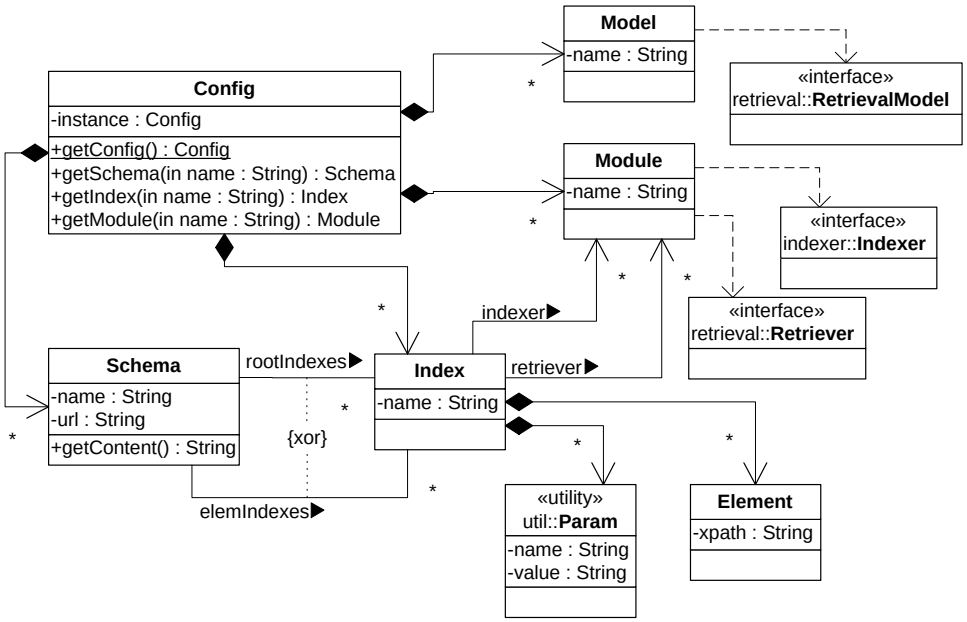


Slika 2.15: Šema dokumenta za konfigurisanje modela pretraživanja

Konfigurisanje podsistema za skladištenje dokumenata je u tesnoj vezi sa konceptima koji čine dizajn ovog podsistema. Opis ovog procesa biće dat u odeljku 2.4.2 koji opisuje podsistem za skladištenje.

Interna reprezentacija konfiguracije

Na osnovu konfiguracije date XML dokumentima gore navedene strukture, XMIRS sistem formira svoju internu reprezentaciju konfiguracije. Sistem podrazumeva da je konfiguracija nepromenljiva tokom rada. Dijagram klasa paketa *config*, prikazan na slici 2.16, predstavlja internu reprezentaciju konfiguracije. Klasa *Model* predstavlja model pronalaženja dokumenata. Pored internog



Slika 2.16: Reprezentacija XMIRS konfiguracije

naziva modela, ova klasa poseduje i vezu sa implementacijom konkretnog modela pronalaženja, predstavljenog interfejsom *RetrievalModel*. Ovaj interfejs pripada paketu *retrieval*, i biće razmatran u odeljku 2.4.5. Klasa *Module* predstavlja modul sistema. Pored toga što kao atribut poseduje svoj interni naziv, povezana je sa odgovarajućim interfejsima analogno klasi *Model*. Interfejs *Indexer* predstavlja indekser modula (v. odeljak 2.4.4), a interfejs *Retriever* predstavlja pretraživač modula (v. odeljak 2.4.5). Klasa *Schema* predstavlja tip dokumenta. Nad datim tipom dokumenta može biti definisano više indeksa, pri čemu su ovde ti indeksi podeljeni na indekse nad celim dokumentom (odnosno nad korenskim elementom dokumenta), predstavljene asocijacijom *rootIndexes* i indekse nad fragmentom dokumenta, predstavljene asocijacijom *elemIndexes*. Klasa *Index* predstavlja indeks definisan nad datim tipom dokumenta. Elementi koji pripadaju indeksu definisani su vezanim *Element* objektima. Indekser koji će se upotrebiti za indeksiranje originalnog sadržaja (prvi indekser u listi) predstavljen je asocijacijom *indexer*. Pretraživač koji će biti upotrebljen za pretraživanje (odgovara modulu čiji indekser je poslednji u listi) predstavljen je asocijacijom *retriever*.

Klasa kojoj pristupaju korisnici ovog paketa je klasa *Config*. U okviru nje implementirana je evidencija o svim registrovanim tipovima dokumenata, modulima, indeksima i modelima pronalaženja. Klasa koristi *singleton* šablon [95] da obezbedi korišćenje tačno jednog svog objekta svim korisnicima. To dalje znači da se i svi objekti kreirani tokom inicijalizacije *Config* objekta kreiraju tačno jednom, što se odnosi i na instanciranje i inicijalizaciju indeksera, pretraživača i modela za pronalaženje.

Prikazani XML Schema dokumenti definišu strukturu konfiguracionih dokumenata XMIRS sistema. Sadržaj konfiguracionih dokumenata koristi se tokom formiranja interne reprezentacije konfiguracije. Prikazana struktura dokumenata u najvećoj meri definiše potrebna ograničenja na njihov sadržaj. Izuzetak su delovi dokumenata u kojima se navode nazivi klasa koji implementiraju odgovarajuće interfejse. Kako se provera postojanja odgovarajućih klasa i uslova da implementiraju potreban interfejs ne može izraziti u okviru XML Schema jezika, to je na administratoru sistema da obezbedi zadovoljenje i ovih ograničenja.

2.4.2 Podsistem za skladištenje dokumenata

Podsistem za skladištenje dokumenata omogućava smeštanje celih dokumenata ili njihovih pojedinih elemenata u masovnu memoriju. Centralni pojam ovog podsistema je *skladište*, objekat koji omogućava skladištenje i učitavanje dokumenata na osnovu njihovog jedinstvenog identifikatora. Operacije koje poseduje skladište su dodavanje, uklanjanje, ažuriranje i učitavanje dokumenta.

Za jedan tip dokumenata (dat njihovom shemom) definiše se jedna *strategija* skladištenja. Strategija obuhvata jedno ili više skladišta. Upotreba pojedinih pripadajućih skladišta zavisi od konkretne strategije. Primer najjednostavnije strategije obuhvata tačno jedno skladište i koristi ga za skladištenje svih dokumenata istog tipa. Složeniji primer strategije predstavlja strategiju koja rukuje sa više skladišta, pri čemu se pojedina skladišta koriste po *round robin* principu. Pored prikazanih strategija, na isti način moguće je dodati i druge strategije.

Mapiranje pojedinih tipova dokumenata na strategije skladištenja definisano je konfiguracijom podsistema za skladištenje.

Skladišta se mogu upotrebiti, osim smeštanja celih dokumenata, i za smeštanje pojedinih njihovih elemenata. Ova mogućnost je prisutna iz sledeća dva razloga.

1. Multimedijalni elementi dokumenata sa specifičnim ograničenjima vezanim za skladištenje moraju imati mogućnost upotrebe posebno konfigurisanog skladišta. Pored toga, elementi za koje postoji specifičan način isporuke njihovog sadržaja klijentima moraju imati posebno skladište koje je dostupno datom sistemu za isporuku sadržaja. Primer ovakvog elementa je video zapis u RealVideo formatu [228] čiji sadržaj se klijentima isporučuje putem posebnog servera koji implementira RTSP protokol [251]. Dati server mora biti u mogućnosti da pristupi sadržajima odgovarajućih elemenata.
2. Pojedini moduli XMIRS sistema u okviru svog dela za indeksiranje mogu imati potrebu za perzistentnim smeštanjem dodeljenih elemenata dokumenta, pri čemu takvu mogućnost nisu sami u stanju da obezbede.

Za označavanje odvojeno skladištenih elemenata koriste se apsolutne XPath putanje u skraćenoj sintaksi [54]. Celokupni dokumenti se mogu takođe oz-

načiti na ovakav način, kao koren stabla dokumenta. Iako se pomoću XPath jezika mogu referencirati i drugi tipovi čvorova XML dokumenta, s obzirom na model dokumenata usvojen za XMIRS, nije dozvoljeno referenciranje čvorova dokumenta koji nisu elementi.

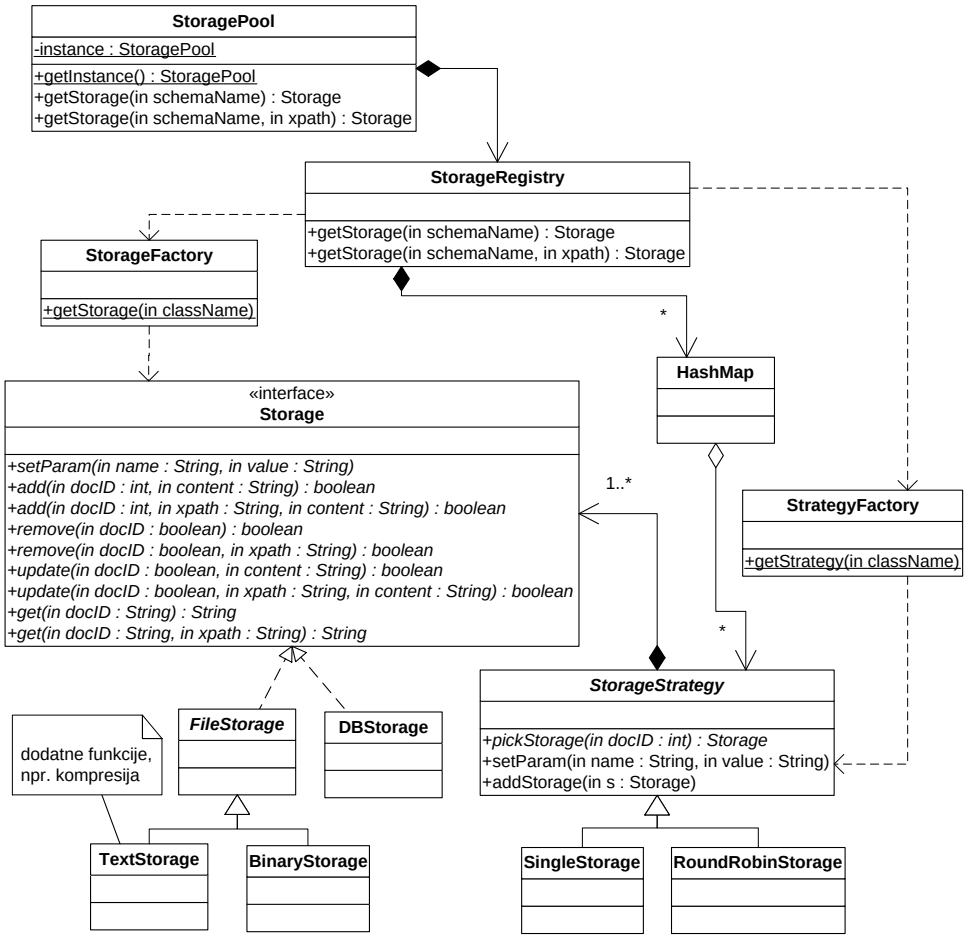
Slika 2.17 predstavlja dijagram klasa podsistema za skladištenje dokumenata. Skladište je predstavljeno interfejsom *Storage*, sa operacijama za dodavanje, uklanjanje, ažuriranje i učitavanje dokumenata i operacijom za definisanje parametara skladišta. Konkretno skladište implementira interfejs *Storage*. Kao primeri konkretnih skladišta, na dijagramu su prikazane klase *TextStorage*, *BinaryStorage* i *DBStorage*, koje implementiraju funkcionalnost skladišta prilagođenu smeštanju tekstualnih fajlova, binarnih fajlova i smeštanju u okviru baze podataka, tim redosledom. Kreiranje *Storage* objekata namena je klase *StorageFactory*, koja predstavlja primenu šablona *parameterized factory method* [95].

Strategija skladištenja definisana je apstraktnom klasom *StorageStrategy*. Jedna strategija poseduje jedno ili više skladišta, što je prikazano vezom kompozicije sa interfejsom *Storage*. Kao primer konkretnih strategija ovde su navedene klase *SingleStorage* i *RoundRobinStorage*. Klasa *SingleStorage* predstavlja implementaciju strategije koja rukuje tačno jednim skladištem, dok klasa *RoundRobinStorage* implementira strategiju koja rukuje nizom skladišta po *round robin* principu. Za instanciranje objekata klase *StorageStrategy* namenjena je klasa *StrategyFactory*, analogno klasi *StorageFactory*.

Mapiranje tipova dokumenata ili pojedinih elemenata u okviru datih tipova na strategije skladištenja implementira klasa *StorageRegistry*. Sa stanovišta ove klase, mapiranje se vrši između naziva tipova dokumenata (eventualno uz dodatu XPath putanju do pojedinačnog elementa) i instanciranih strategija skladištenja. Metode klase *StorageRegistry* omogućavaju pristup konkretnom skladištu koje određuje upotrebljena strategija za dati tip dokumenta (ili elementa) i dati dokument.

Jedinstvenost mapiranja tipova na strategije, koje je implementirano klasom *StorageRegistry*, obezbeđuje klasa *StoragePool* upotrebom šablona *singleton* [95]. Ova klasa predstavlja jedinu tačku pristupa podsistemu za skladištenje.

Konfiguracija podsistema za skladištenje definiše se XML dokumentom čija struktura je data XML Schema dokumentom sa listinga 2.13, odnosno



Slika 2.17: Dijagram klasa podsistema za skladištenje

slike 2.18. Dokument ima dva glavna dela: element *registry/storages* sadrži definicije skladišta, dok element *registry/strategies* sadrži definicije strategija skladištenja.

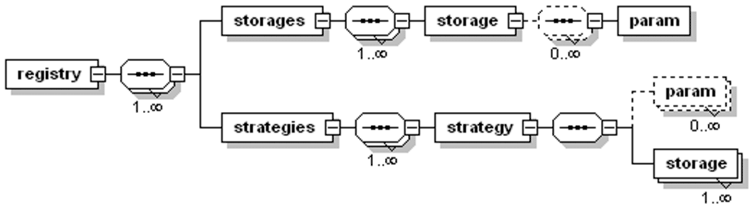
Element *storage* namenjen je definisanju skladišta. Njegov atribut *name* predstavlja interni naziv skladišta, dok atribut *class* predstavlja naziv klase koja implementira skladište. Parametri skladišta definišu se elementom *param* slično kao i u prethodnim situacijama.

Strategija skladištenja definiše se elementom *strategy*. Njegov atribut *name* predstavlja interni naziv strategije. Atributom *class* navodi se naziv klase koja implementira strategiju. Atribut *schema* predstavlja naziv tipa dokumenta za koga se definiše skladište. Opcionim atributom *element* navodi se, pomoću XPath izraza, element dokumenta koji će se smestiti u okviru skladišta. Ukoliko se ovaj atribut ne navede, podrazumeva se skladištenje kompletnog dokumenta. Parametri strategije definišu se odgovarajućim elementom *param* slično kao i u prethodnim situacijama. Skladišta kojima će rukovati data strategija navode se elementom *strategy/storage* čiji atribut *name* predstavlja naziv skladišta.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="registry">
    <xs:complexType>
      <xs:sequence maxOccurs="unbounded">
        <xs:element name="storages">
          <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
              <xs:element name="storage">
                <xs:complexType>
                  <xs:sequence minOccurs="0" maxOccurs="unbounded">
                    <xs:element name="param">
                      <xs:complexType>
                        <xs:attribute name="name" type="xs:string" use="required"/>
                        <xs:attribute name="value" type="xs:string" use="required"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="name" type="xs:token" use="required"/>
                  <xs:attribute name="class" type="xs:string" use="required"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="strategies">
```

```
          <xs:complexType>
            <xs:sequence maxOccurs="unbounded">
              <xs:element name="strategy">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="param" minOccurs="0" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:attribute name="name" type="xs:string" use="required"/>
                        <xs:attribute name="value" type="xs:string" use="required"/>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="storage" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:attribute name="name" type="xs:token" use="required"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                  <xs:attribute name="name" type="xs:token" use="required"/>
                  <xs:attribute name="class" type="xs:string" use="required"/>
                  <xs:attribute name="schema" type="xs:string" use="required"/>
                  <xs:attribute name="element" type="xs:string" use="optional"/>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 2.13: Šema dokumenta za konfigurisanje podsistema za skladištenje



Slika 2.18: Šema dokumenta za konfigurisanje podsistema za skladištenje

Rad sa različitim skladištima omogućava, između ostalog, i povezivanje XMIRS sistema sa drugim sistemima putem XML dokumenata. Mapiranje XML dokumenata na koncepte kojima rukuju drugi sistemi implementira se

u okviru skladišta. Na primer, veza sa relacionim bazama podataka može se realizovati pomoću skladišta koje implementira mapiranje XML dokumenata na datu relacionu šemu.

2.4.3 Podsistem za rukovanje dokumentima

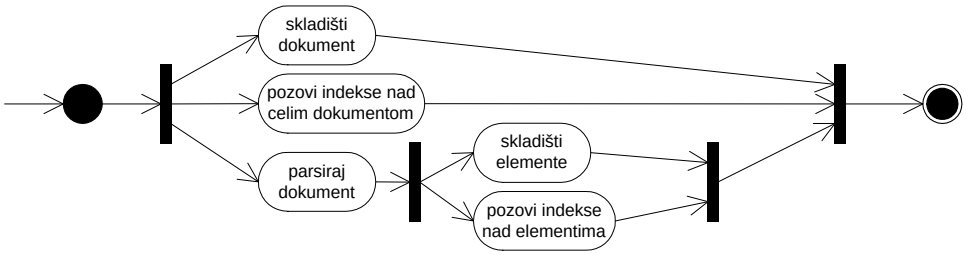
Rukovanje dokumentima podrazumeva operacije dodavanja, uklanjanja, ažuriranja i učitavanja dokumenta. Implementacije ovih operacija oslanjaju se na podsistem za skladištenje. Pored toga, operacije dodavanja, uklanjanja i ažuriranja obuhvataju i interakciju sa podsistemom za indeksiranje dokumenata.

Operacija dodavanja dokumenta ilustrovana je dijagramom aktivnosti na slici 2.19. Operacija se odvija u više paralelnih niti kako bi se iskoristio paralelizam u radu pojedinih delova računarskog sistema. Prilikom dodavanja dokumenta pokreću se tri niti:

1. nit za skladištenje dokumenta, u okviru koje se pozivaju funkcije podsistema za skladištenje,
2. nit za pozivanje svih modula za indeksiranje koji su registrovani za ceo dokument,
3. nit za parsiranje dokumenta, u kojoj se, pored procesa parsiranja, pokreću sledeće niti:
   (a) nit za skladištenje elemenata, koji su u okviru konfiguracije skladišta određeni za odvojeno skladištenje i
   (b) nit za pozivanje svih modula za indeksiranje koji su registrovani za pojedine elemente dokumenta.

Nit za skladištenje dokumenta pronalazi skladište namenjeno datom tipu dokumenata u okviru konfiguracije skladišta. Potom pokreće odgovarajuće operacije za izabrano skladište.

Indeksima koji su, u okviru konfiguracije sistema, navedeni kao indeksi nad celim dokumentom (tj. nad njegovim korenskim elementom) može biti prosleđen celokupan sadržaj dokumenta odmah po njegovom prispeću u sistem. Dalja analiza sadržaja dokumenta je u nadležnosti modula za indeksiranje. Nit za pozivanje modula za indeksiranje vrši notifikaciju svih modula koji su registrovani za indeksiranje celokupnog dokumenta.

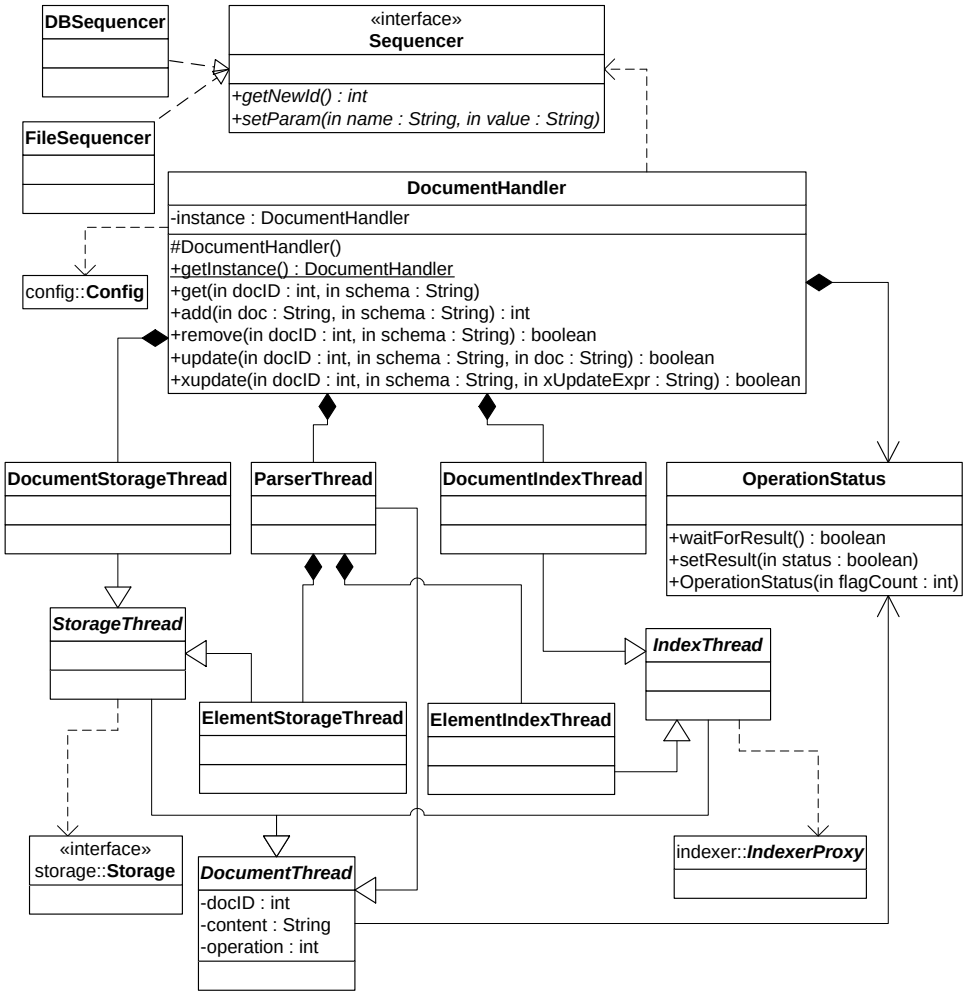


Slika 2.19: Proces dodavanja dokumenta

Elementi koji imaju zasebno skladištenje moraju biti izdvojeni iz sadržaja dokumenta i poslati u odgovarajuće skladište. Izdvajanje takvih elemenata podrazumeva parsiranje pristiglog XML dokumenta. Pored toga, indeksi nad pojedinim (ne-korenskim) elementima dokumenta kao ulaz prihvataju samo sadržaj datog elementa. Izdvajanje elemenata namenjenih za indeksiranje je drugi zadatak koji se obavlja u toku parsiranja dokumenta. Proces parsiranja dokumenta i izdvajanja elemenata namenjenih za posebno skladištenje i elemenata namenjenih za indeksiranje implementira nit za parsiranje dokumenta. Za svaki element namenjen posebnom skladištenju pokreće se nit za skladištenje elementa. Za svaki element namenjen indeksiranju pokreće se nit pozivanja modula za indeksiranje.

Ovakva organizacija programskih niti u toku procesa dodavanja dokumenta obezbeđuje da se parsiranje dokumenta za potrebe izdvajanja pojedinih elemenata, kao vremenski zahtevna operacija, izvršava najviše jednom. Sa druge strane, indeksi nad celokupnim dokumentom nisu obuhvaćeni ovim parsiranjem. Analiza sadržaja dokumenta za ovakve indekse je potpuno u nadležnosti odgovarajućeg modula za indeksiranje.

Dijagram klasa sa slike 2.20 predstavlja klase podsistema za rukovanje dokumentima. Centralna klasa ovog sistema je *DocumentHandler* koja implementira operacije rukovanja dokumentima: dodavanje, uklanjanje, ažuriranje i učitavanje. Prilikom operacije dodavanja dokumenata potrebno je formirati jedinstveni identifikator dokumenta. Interfejs *Sequencer* predstavlja apstrakciju različitih implementacija za generisanje jedinstvenih identifikatora dokumenata. Na slici su, ilustracije radi, prikazane dve različite implementacije: *FileSequencer*, koji koristi fajl-sistem za perzistenciju podataka o generisanim identifikatorima, i *DBSequencer*, koji za ovu namenu koristi relacionu bazu

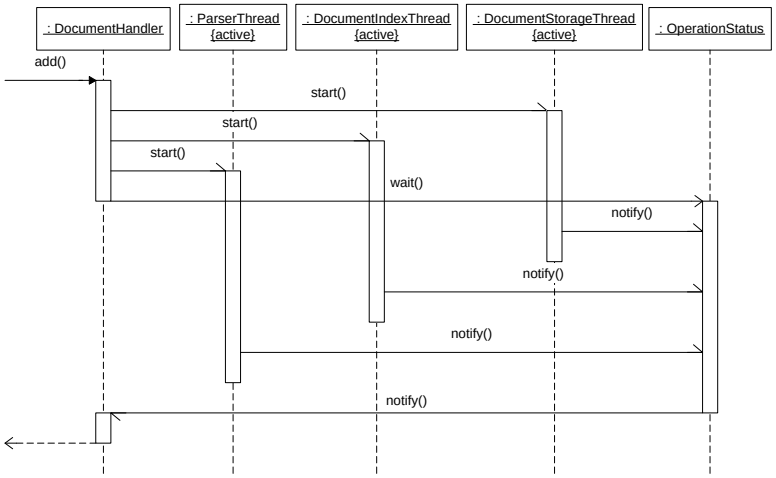


Slika 2.20: Dijagram klasa podsistema za rukovanje dokumentima

podataka. Ažuriranje postojećeg dokumenta može se izvršiti na dva načina: (1) navođenjem kompletnog sadržaja nove verzije dokumenta, čemu je namenjena metoda *update*, i (2) navođenjem XUpdate [148] izraza čija interpretacija rezultuje novim sadržajem dokumenta (metoda *xupdate*). Upotreba XUpdate jezika pokazuje se posebno korisnom kada je odnos veličine promenjenih delova dokumenta i veličine celog dokumenta mali.

Nit za skladištenje dokumenta implementirana je klasom *DocumentStorageThread*. Nit za skladištenje pojedinih elemenata implementirana je klasom *ElementStorageThread*. Veza prema skladištu (tj. interfejsu *Storage*) izdvojena je u roditeljsku apstraktnu klasu *StorageThread*.

Nit za parsiranje dokumenta definisana je klasom *ParserThread*. Niti za pozivanje modula za indeksiranje, date klasama *DocumentIndexThread* i *ElementIndexThread*, imaju kao roditelja apstraktnu klasu *IndexThread* u koju je izdvojena veza sa podsistemom za indeksiranje. Apstraktna klasa *DocumentThread* predstavlja zajedničkog pretka svim klasama koje predstavljaju niti korišćene u toku procesa dodavanja dokumenta. Klasa *OperationStatus* predstavlja objekte za sinhronizaciju prethodno opisanih niti namenjene za čuvanje tekućeg statusa izvršavane operacije.



Slika 2.21: Operacija dodavanja dokumenta

Operacije uklanjanja i ažuriranja dokumenata definisane su analogno operaciji dodavanja. Organizacija programskih niti ovih operacija je jednaka. Razlika se nalazi u okviru samih niti, gde se pozivaju odgovarajuće operacije skladišta i modula za indeksiranje, zavisno od operacije koja se sprovodi za dati dokument. Dijagram sekvenci na slici 2.21 prikazuje operaciju dodavanja dokumenta koja se sprovodi u više programskih niti. Dodavanje dokumenta implementira se u okviru metode *add* klase *DocumentHandler*. U okviru ove metode pokreću se tri programske niti: nit za skladištenje dokumenta, nit za pozivanje indeksera za indekse nad celim dokumentom i nit za parsiranje dokumenta. Iako se ove tri niti izvršavaju nezavisno jedna od druge, celokupna operacija (implementirana kao metoda *add*) se smatra završenom kada sve tri niti završe sa radom. Označavanje završetka rada pokrenutih niti obavlja se putem asinhronih poruka upućenih objektu klase *OperationStatus*.

2.4.4 Podsistem za indeksiranje dokumenata

Podsistem za indeksiranje ima namenu da obezbedi pozivanje odgovarajućih modula, odnosno njihovih indeksera tokom izvršavanja operacija rukovanja dokumentima. Važna karakteristika ovog podsistema je da omogućava deklarativan izbor načina pozivanja funkcija indeksera, koji se navodi u okviru konfiguracije indeksa. Na raspolaganju su sledeća tri načina pozivanja indeksera.

- *Sinhroni.* Operacija formiranja indeksa za dati dokument izvršava se kao sastavni deo programske niti koja ju je pokrenula. U ovom slučaju u pitanju je nit za indeksiranje kompletnog dokumenta, ili nit za parsiranje dokumenta. Ovakav način pozivanja najviše odgovara situacijama kada je proces indeksiranja kratkog trajanja, pa upotreba posebne programske niti samo za ovaj proces predstavlja dugotrajniju operaciju.

- *Asinhroni, u istom adresnom prostoru.* Pokretanje posebne programske niti u okviru XMIRS serverskog programa koja izvršava proces indeksiranja poželjno je u situacijama kada se proces indeksiranja može efikasno paralelizovati sa drugim aktivnostima sistema.

- *Asinhroni, u drugom adresnom prostoru.* Implementacija procesa indeksiranja u okviru zasebnog serverskog programa koji je dostupan putem

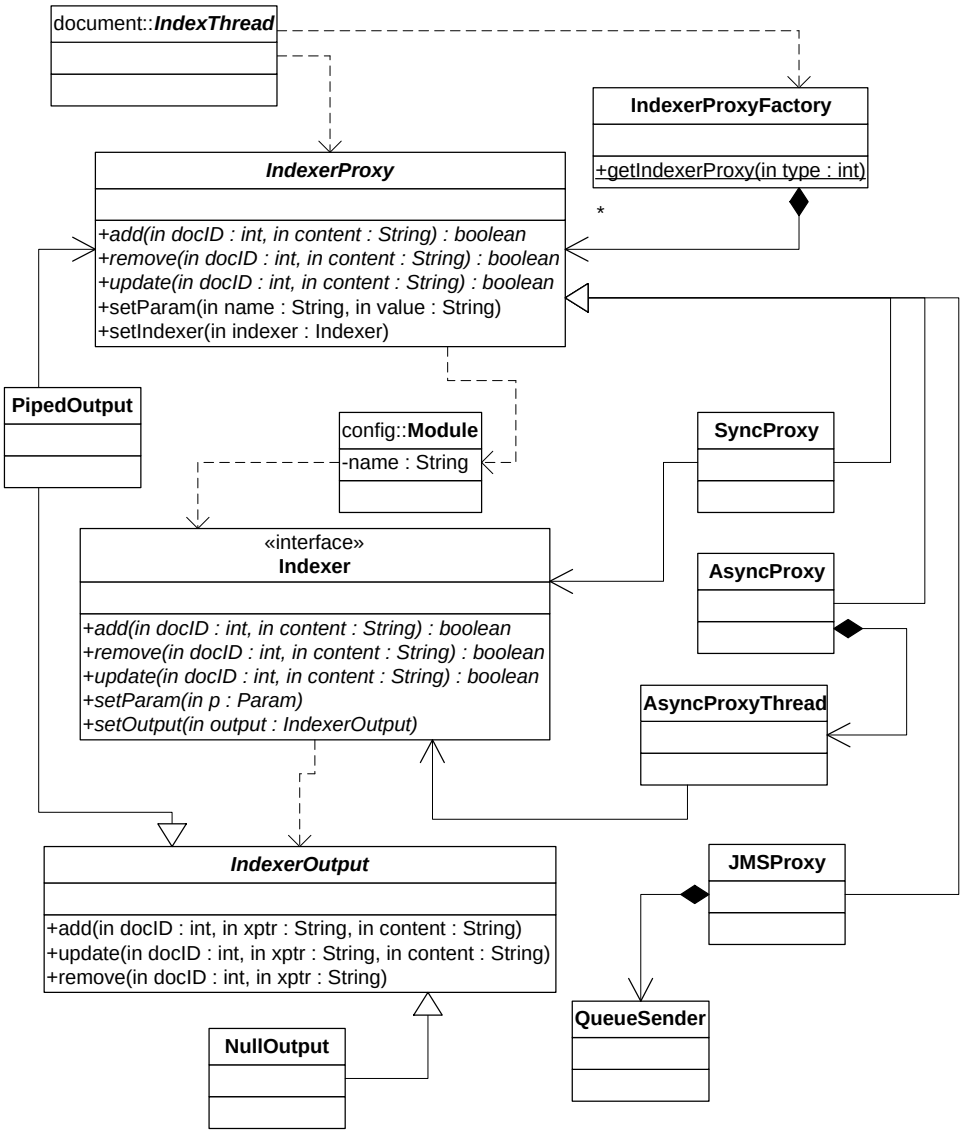
računarske mreže omogućava formiranje distribuiranog sistema za indeksiranje dokumenata. Komunikacija osnovnog XMIRS serverskog programa sa ovako konfigurisanim indekserima odvija se putem asinhrone razmene poruka kroz redove poruka.

Slika 2.22 predstavlja dijagram klasa podsistema za indeksiranje. Indekser, kao sastavni deo modula sistema, predstavljen je interfejsom *Indexer*. Apstraktna klasa *IndexerProxy* predstavlja vezu programskih niti koje se izvršavaju u okviru operacija nad dokumentima sa konkretnim indekserom. Razdvajanje konkretnog indeksera od niti koja ga poziva omogućava izbor načina pozivanja njegovih metoda. Naslednici *IndexerProxy* klase implementiraju jednu od mogućih tehnika pozivanja indeksera. Klasa *SyncProxy* implementira sinhrono pozivanje indeksera. Klasa *AsyncProxy* implementira asihnrono pozivanje u istom adresnom prostoru koristeći instancu klase *AsyncProxyThread* kao novu programsku nit. Klasa *JMSProxy* predstavlja jednu mogućnost implementacije asinhronog pozivanja u zasebnom adresnom prostoru koja koristi *Java Message Service* (JMS) tehnologiju kao implementaciju međuprocesne komunikacije zasnovane na redovima poruka. Klasa *QueueSender* i interfejs *MessageListener* su sastavni deo JMS biblioteke.

Rezultat procesa indeksiranja, pored formirane strukture indeksa, može biti i sadržaj koji se prosleđuje drugim indekserima, omogućavajući tako sekvencijalno nizanje više indeksa. Apstraktna klasa *IndexerOutput* predstavlja odredište ovakvih sadržaja. Postoje dve konkretne implementacije ove klase. Upotreba klase *NullOutput* predstavlja ignorisanje sadržaja koga je proizveo dati indekser. Klasa *PipedOutput* implementira vezu dobijenog sadržaja prema sledećem indekseru u nizu. Pozivanje sledećeg indeksera koristi isti mehanizam definisan *IndexerProxy* klasom.

Dijagram sekvenci na slici 2.23 prikazuje situaciju sinhronog pozivanja indeksera prilikom operacije dodavanja dokumenta za indeks definisan nad celim dokumentom. Operacija dodavanja dokumenta pokreće programsku nit za poziv svih indeksera nad celim dokumentom (objekat klase *DocumentIndexThread*). Dijagram prikazuje poziv jednog indeksera, ali ovih poziva može biti i više u okviru jedne programske niti. Poziv metode objekta klase *SyncProxy* odvija se u istoj niti, kao i direktni poziv potrebnog indeksera.

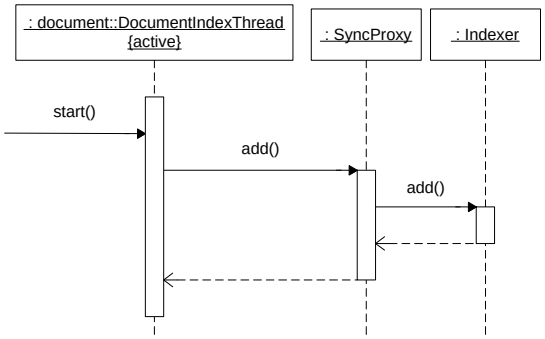
Slika 2.24 prikazuje situaciju asihronog poziva indeksera u okviru istog adresnog prostora, ponovo za slučaj operacije dodavanja dokumenta za in-



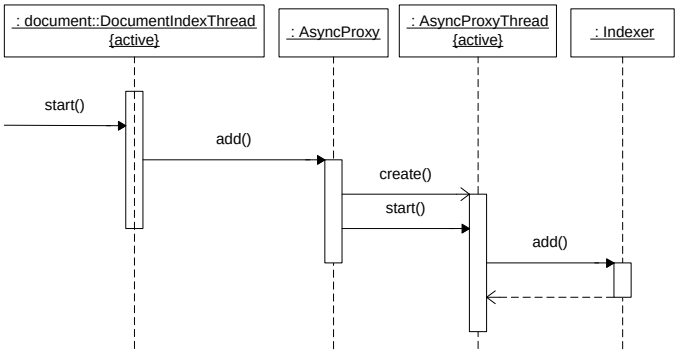
Slika 2.22: Dijagram klasa podsistema za indeksiranje dokumenata

deks definisan nad celim dokumentom. Ovaj put poziva se metoda objekta klase *AsyncProxy*, koja pokreće novu programsku nit i u okviru nje direktno poziva potrebni indekser. Završetak rada operacije dodavanja dokumenta u ovom slučaju ne zavisi od završetka rada indeksera.

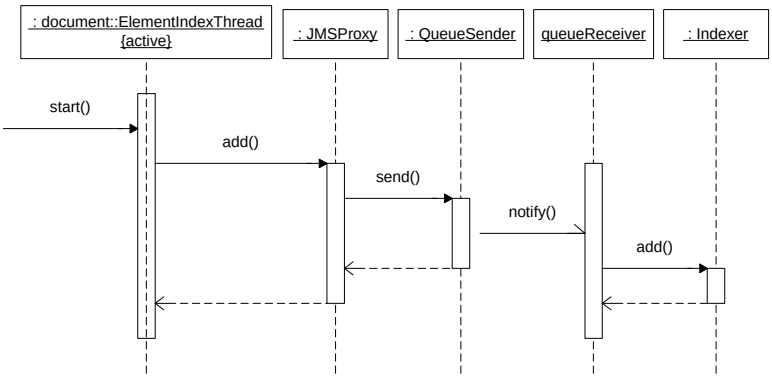
Na slici 2.25 prikazan je slučaj pozivanja indeksera putem JMS servisa. Objekat klase *JMSProxy* putem *QueueSender* objekta šalje poruku drugom serverskom programu koji ima funkciju indeksera. Na strani indeksera poruku će primiti odgovarajući objekat koji će potom direktno pozvati indekser.



Slika 2.23: Sinhrono pozivanje indeksera



Slika 2.24: Asinhrono pozivanje indeksera u istom adresnom prostoru



Slika 2.25: Asinhrono pozivanje indeksera u drugom adresnom prostoru

2.4.5 Podsistem za pronalaženje dokumenata

Osnovni zadatak podsistema za pronalaženje dokumenata je da obezbedi sledeće funkcije:

- pozivanje pojedinačnih modula, odnosno njihovih pretraživača,
- preuzimanje rezultata pretrage pojedinačnih modula,
- formiranje ukupnog skupa pronađenih dokumenata i
- pozivanje izabranog modela pronalaženja dokumenata radi formiranja konačnog skupa rezultata.

Pod *pretraživačem* se podrazumeva deo modula sistema koji ima funkciju pretraživanja dokumenata. Model pronalaženja dokumenata se, sa aspekta ovog podsistema, tretira kao programski modul koji implementira funkcije formiranja konačnog rezultata upita.

Komunikacija korisnika sa podsistemom za pronalaženje dokumenata odvija se putem XML dokumenata koji predstavljaju kako same upite, tako i rezultate pronalaženja. Struktura ovih dokumenata zavisi od izabranog modela pretraživanja. Primer ovakvih dokumenata dat je u odeljku 2.3. Pored toga, komunikacija izabranog modela pronalaženja sa pojedinačnim modulima u pogledu slanja elementarnih upita takođe se odvija putem XML dokumenata.

Slika 2.26 predstavlja dijagram klasa podsistema za pronalaženje dokumenata. Klasa *RetrievalEngine* predstavlja vezu prema korisniku. Jedino njene

metode su direktno dostupne korisniku podsistema. Klasa *UserProfile* sadrži opšte podatke o tekućem korisniku i njegovim prethodnim sesijama pronalaženja. Sa druge strane, klasa *UserSession* predstavlja tekuću sesiju pronalaženja. Klasa *QueryContext* implementira proces pozivanja odgovarajućih pretraživača za potrebe interpretacije složenih upita. Različiti modeli pronalaženja dokumenata predstavljeni su interfejsom *RetrievalModel*.

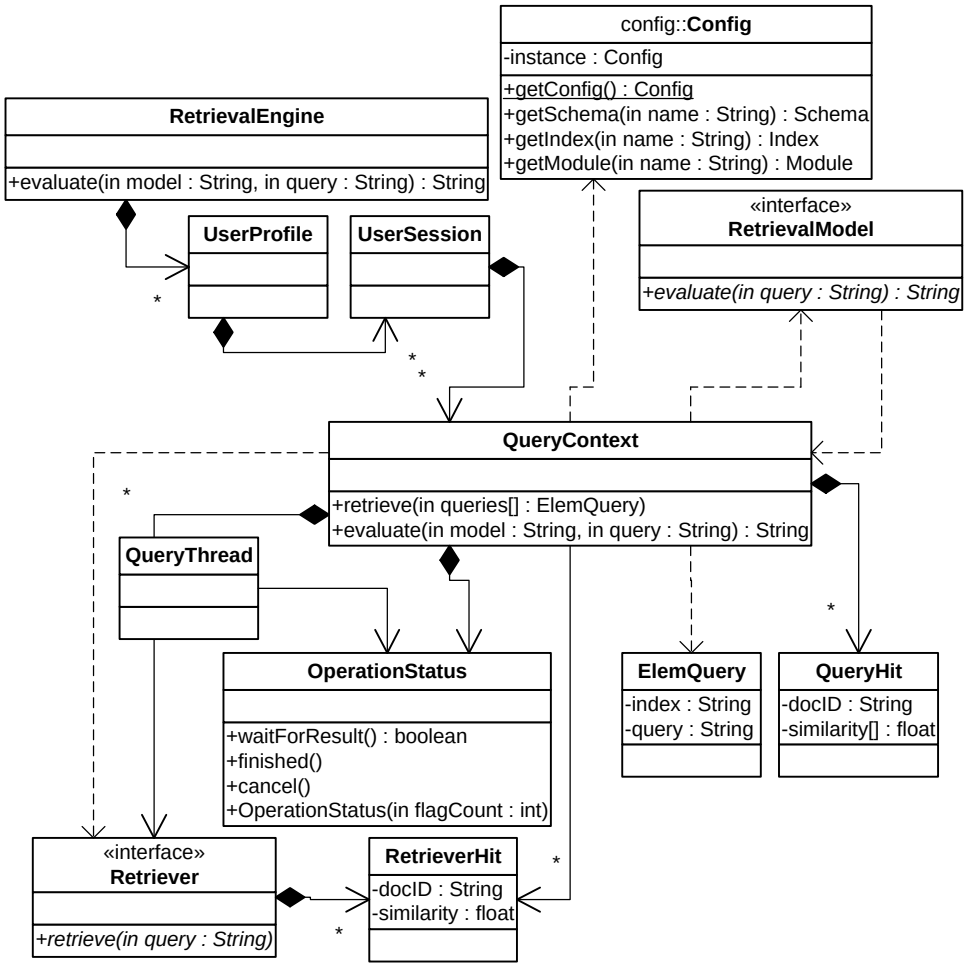
Pretraživači, kao sastavni deo modula sistema, predstavljeni su interfejsom *Retriever*. Pozivanje svih pretraživača uvek je asinhrono i odvija se u okviru programske niti definisane klasom *QueryThread*. Označavanje završetka rada pokrenutih niti obavlja se putem asinhronih poruka upućenih objektu klase *OperationStatus*. Elementarni pogodak, nastao kao rezultat rada pretraživača odnosno interpretacije elementarnog upita, predstavljen je klasom *RetrieverHit*. Atributi ove klase su identifikator pronađenog dokumenta i mera njegove sličnosti sa upitom. Klasa *QueryHit* predstavlja pronađeni dokument za koji su vezani svi koeficijenti sličnosti formirani u okviru interpretacije elementarnih upita. Njeni atributi su, dakle, identifikator dokumenta i niz odgovarajućih koeficijenata sličnosti.

Implementacija modela za pronalaženje dokumenata će, za potrebe interpretiranja elementarnih upita, pozvati metodu *retrieve* klase *QueryContext*. Parametar ove metode je lista *ElemQuery* objekata, koji predstavljaju elementarni upit. Atributi klase *ElemQuery* su naziv indeksa nad kojim se vrši pretraživanje i sadržaj elementarnog upita.

Proces interpretacije složenog upita za izabrani model pronalaženja prikazan je dijagramom sekvenci na slici 2.27. Proces počinje pozivom metode *evaluate* klase *QueryContext*. Na osnovu njenog parametra *model*, određuje se potrebna implementacija modela pronalaženja. Datoj implementaciji prosleđuje se dobijeni složeni upit pozivom njene metode *evaluate*.

Prvi zadatak izabrane implementacije modela pretraživanja je parsiranje dobijenog složenog upita (na dijagramu je to predstavljeno pozivom metode *parse* istog objekta). Rezultat parsiranja, je, između ostalog, lista elementarnih upita koja se prosleđuje početnom *QueryContext* objektu.

Primljene elementarne upite *QueryContext* objekat upućuje na interpretiranje odgovarajućim modulima, odnosno njihovim pretraživačima. Na dijagramu je, radi preglednosti, prikazana situacija kada se poziva samo jedan pretraživač. Pokretanje pretraživača je asinhrono i odvija se u okviru pro-



Slika 2.26: Dijagram klasa podsistema za pronalaženje dokumenata

gramskih niti predstavljenih objektima klase *QueryThread*. Rezultat metode *retrieve* pretraživača je lista objekata klase *RetrieverHit*. Inicijalna programska nit u kojoj se odvija interpretacija upita je blokirana dok sve niti u kojima se pozivaju pretraživači ne završe sa radom (što označavaju pozivom metode *finished* objekta klase *OperationStatus*). Nakon toga, izvršavanje inicijalne niti se nastavlja i obuhvata formiranje objekata klase *QueryHit* koji predstavljaju zbirne rezultate pretrage svih pretraživača.

Kontrola se potom vraća objektu koji predstavlja implementaciju modela pronalaženja gde se formira konačan rezultat procesa pronalaženja i prosleđuje kao rezultat poziva metode *evaluate*. Ovako dobijeni rezultat direktno se dalje prosleđuje korisniku podsistema.

Ukoliko jedan pretraživač tokom izračunavanja upita dobije takav rezultat da rezultati ostalih pretraživača nisu od značaja, može pozivom metode *cancel* (umesto *finished*) objekta klase *OperationStatus* prekinuti dalje čekanje *QueryContext* objekta na završetak rada preostalih niti.

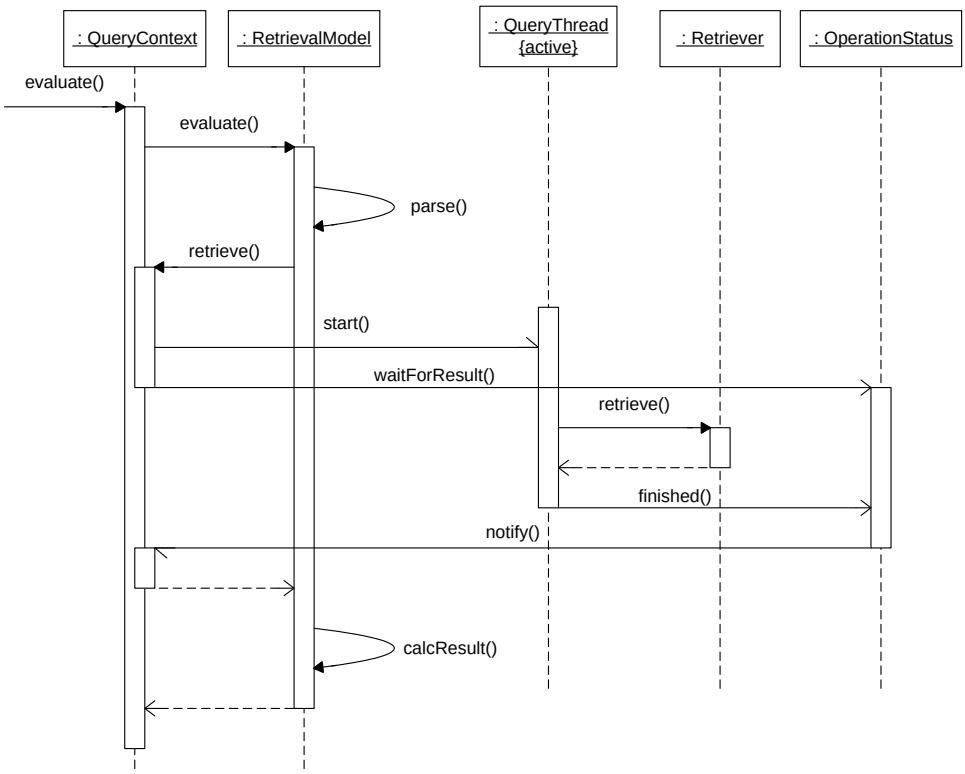
2.5 Okruženje implementacije

Implementacija jezgra XMIRS sistema je izvedena upotrebom programskog jezika Java. Prototip je implementiran prema specifikaciji datoj u prethodnom odeljku. Ovaj odeljak prikazuje osnovnu strukturu i najvažnije karakteristike implementacije prototipa. Paketi iz kojih se sastoji prototip su navedeni u tabeli 2.2. Svi navedeni paketi se nalaze u okviru korenskog paketa *com.gint.app.xmirs*.

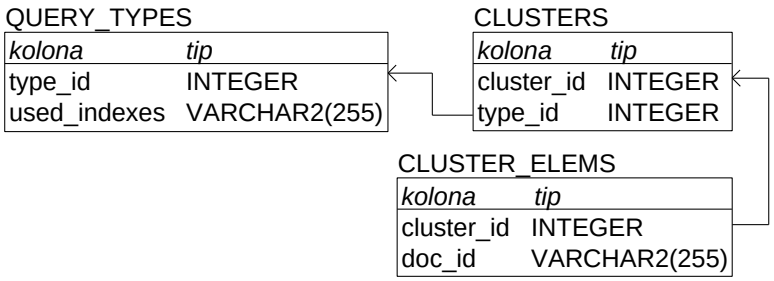
Instalacija prototipa se sastoji iz odgovarajuće strukture direktorijuma. Direktorijumi su pobrojani u tabeli 2.3. Prikazani direktorijumi čine hijerarhiju sa zajedničkim korenom, koji predstavlja osnovni direktorijum instalacije.

Podsistem za skladištenje dokumenata obuhvata implementaciju skladišta koja se oslanja na fajl-sistem. Implementirana strategija skladištenja rukuje tačno jednim skladištem. Generisanje jedinstvenih identifikatora dokumenata koristi fajl-sistem za perzistenciju podataka.

Prototip jezgra XMIRS-a koristi sistem za upravljanje relacionim bazama podataka u okviru implementacije modela sličnih klastera. Slika 2.28 predstavlja relacionu šemu koju koristi ova implementacija.



Slika 2.27: Proces izračunavanja upita



Slika 2.28: Šema baze podataka implementacije modela sličnih klastera

Naziv	**Opis**
config	Konfiguracija sistema
document	Podsistem za rukovanje dokumentima
indexer	Podsistem za indeksiranje
models.cluster	Implementacija modela sličnih klastera
models.extbool	Implementacija bulovskog modela
models.vector	Implementacija vektorskog modela
retrieval	Podsistem za pronalaženje dokumenata
storage	Podsistem za skladištenje dokumenata
storage.file	Implementacija skladišta pomoću fajl-sistema
storage.simple	Implementacija strategije skladištenja
util	Pomoćne klase sistema

Tabela 2.2: Paketi iz kojih se sastoji prototip jezgra XMIRS-a

Naziv	**Opis**
lib	Programski kôd prototipa
conf	Dokumenti sa konfiguracijom XMIRS-a
conf/schemas	Tipovi dokumenata kojima rukuje XMIRS
data	
data/docs	Skladištenje dokumenata

Tabela 2.3: Direktorijumi koje koristi prototip jezgra XMIRS-a

Relacija *QUERY_TYPES* predstavlja tipove upita koji su do tekućeg trenutka postavljani sistemu. Tipovi upita se međusobno razlikuju po skupu indeksa koji je korišćen u upitu. Relacija *CLUSTERS* predstavlja klastere koji čine rezultat pojedinog upita. Relacija *CLUSTER_ELEMS* predstavlja elemente pojedinog klastera.

Poglavlje 3

Verifikacija prototipa sistema na digitalnoj biblioteci

Kolekcije dokumenata čuvane u klasičnim bibliotekama obrađuju se radi formiranja njihovog kataloga. Ovakvi katalozi se, u računarski podržanim informacionim sistemima, zasnivaju na odgovarajućoj strukturi (formatu) za reprezentaciju dokumenata. Međunarodni standard UNIMARC [287] definiše ovakav format. Informacioni sistemi zasnovani na UNIMARC-u, poput Bibliotečkog informacionog sistema BISIS [275], rukuju opisima dokumenata biblioteke (zapisima) u UNIMARC formatu. Sadržaj samih dokumenata ostaje izvan UNIMARC zapisa. Sa druge strane, digitalna biblioteka podrazumeva rukovanje digitalnim/digitalizovanim dokumentima. Informacioni sistem digitalne biblioteke rukuje kako opisima dokumenata, tako i samim dokumentima u digitalnom obliku.

3.1 Digitalna biblioteka teza i disertacija

Projekat mreže digitalnih biblioteka doktorskih i magistarskih teza (*Networked Digital Library of Theses and Dissertations*, NDLTD) [195] započet je na univerzitetu Virginia Tech, SAD. Glavni cilj NDLTD projekta je jednostavna i efikasna centralizovana katalogizacija doktorskih i magistarskih teza odbranjenih u okviru institucija-članova.

Sistem se sastoji od čvorova u mreži koji predstavljaju autonomna skladišta dokumenata. Originalni, kompletni dokumenti se čuvaju na odgovarajućem čvoru u nekom od pogodnih formata digitalnih dokumenata (najčešće PDF [206]). Čvorovi mreže omogućavaju pronalaženje dokumenata koje sami skladište. Proces pronalaženja dokumenata koristi metapodatke o dokumentima koji čine sastavni deo opisa dokumenata. U formiranju metapodataka za dokument učestvuju autor dokumenta i nadležni bibliotekari. U smislu klasifikacije sistema za indeksiranje date u [283, 289] (v. poglavlje 1), NDLTD predstavlja primer sistema sa ručno generisanim indeksom, pri čemu neki elementi indeksa pripadaju kontrolisanom rečniku. Prilikom pregleda rezultata pronalaženja, moguće je preuzeti i originalni oblik traženog dokumenta. NDLTD mreža ne nameće posebne zahteve pred funkcionalnost pronalaženja dokumenata implementirane u okviru pojedinih čvorova.

Centralni sistem za pronalaženje dokumenata [196] postoji kao poseban čvor mreže koji omogućava pronalaženje dokumenata skladištenih na svim ostalim čvorovima. Pronalaženje koristi metapodatke formirane na originalnim čvorovima. Centralni sistem ne skladišti originalne dokumente, već samo URL adrese do originalnog čvora na kome su skladišteni.

U ovom poglavlju predstavljena je primena XMIRS sistema za pronalaženje dokumenata u okviru jednog čvora mreže. Pored toga, dat je predlog proširenja osnovnog opisa dokumenata tako da će on, pored metapodataka i originalnog dokumenta, sadržati i statične slike i video zapise koji se mogu pretraživati po sadržaju. U daljem tekstu se pod NDLTD dokumentima podrazumevaju dokumenti ove proširene strukture.

3.2 Dokumenti digitalne biblioteke

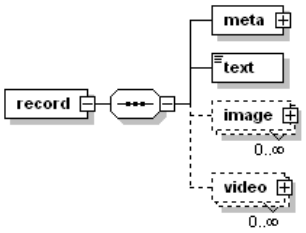
NDLTD sistem rukuje jednim tipom dokumenata. U ovom odeljku predstavljena je struktura ovih dokumenata i definisani su zahtevi u pogledu potrebne funkcionalnosti pronalaženja dokumenata.

3.2.1 Struktura dokumenata

Dokument kojim rukuje NDLTD sistem je XML dokument čija struktura je predstavljena na listingu 3.1. Dokumenti ovog tipa sadrže sledeće elemente:

- digitalnu verziju originalnog dokumenta u PDF formatu,
- tekstualne metapodatke predstavljene kao ključna dokumentacijska informacija,
- bitmapirane slike koje, prema autoru dokumenta, mogu biti korisne za pretraživanje po sadržaju i
- video zapise čija namena je jednaka nameni slika.

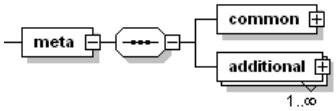
Struktura dokumenta ovde je ilustrovana u nekoliko koraka. Slika 3.1 predstavlja koren hijerarhijske strukture dokumenta. Element *record* predstavlja korenski element dokumenta. Njegovi neposredni naslednici predstavljaju prethodno opisana četiri osnovna dela dokumenta. Element *meta* sadrži strukturirane tekstualne metapodatke koji opisuju originalni dokument. Element *text* sadrži binarni zapis originalnog dokumenta u PDF formatu koji je, prilikom smeštanja u XML strukturu, kodiran Base64 [32] algoritmom. Element *image* je ponovljiv i predstavlja mesto za skladištenje bitmapiranih slika. Element *video*, analogno prethodnom elementu, predstavlja mesto za skladištenje video zapisa.



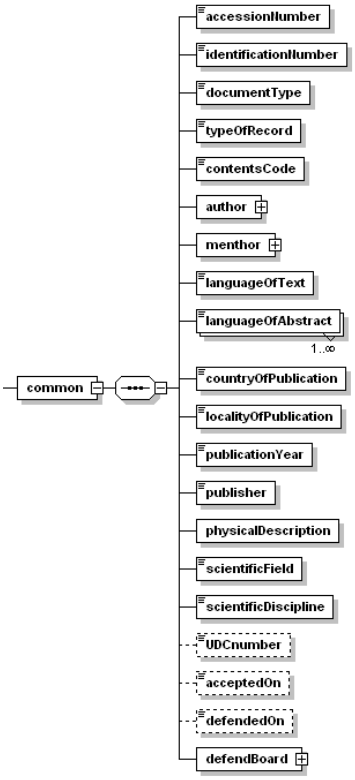
Slika 3.1: Shema NDLTD dokumenta: osnova hijerarhije

Metapodaci su podeljeni na dva segmenta: prvi čine metapodaci čiji sadržaj je numeričkog tipa ili pripada kontrolisanom rečniku, dok drugi deo čine oni čiji je sadržaj tekstualni i može se navesti više puta, na više različitih (prirodnih) jezika. Na slici 3.2 prikazani su element *common*, namenjen prvoj grupi metapodataka, i ponovljivi element *additional*, predviđen za drugu grupu metapodataka.

Slika 3.3 predstavlja podstablo dokumenta čiji koren je element *common*. Elementi ovog podstabla su pobrojani i opisani u tabeli 3.1.



Slika 3.2: Shema NDLTD dokumenta: metapodaci

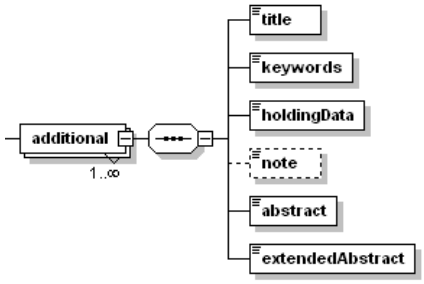


Slika 3.3: Shema NDLTD dokumenta: prva grupa metapodataka

Naziv	**Opis**
accessionNumber	redni broj dokumenta
identificationNumber	identifikacioni broj
documentType	tip dokumentacije
typeOfRecord	tip zapisa
contentsCode	vrsta rada
author	podaci o autoru
menthor	podaci o mentoru
languageOfText	jezik publikacije
languageOfAbstract	jezik apstrakta
countryOfPublication	zemlja publikovanja
localityOfPublication	uže geografsko područje
publicationYear	godina objavljivanja
publicationPlace	mesto izdavanja
publisher	izdavač
physicalDescription	fizički opis rada
scientificField	naučna oblast
scientificDiscipline	uža naučna oblast
UDCnumber	UDK broj
acceptedOn	datum prihvatanja teme
defendedOn	datum odbrane
defendBoard	članovi komisije

Tabela 3.1: Elementi iz prve grupe metapodataka

Slika 3.4 predstavlja podstablo dokumenta za drugu grupu metapodataka sa elementom *additional* kao korenom. Elementi ovog podstabla su pobrojani i opisani u tabeli 3.2.

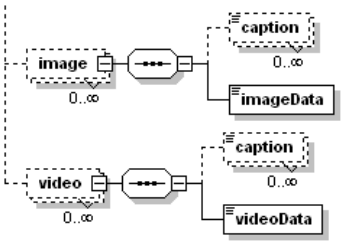


Slika 3.4: Shema NDLTD dokumenta: druga grupa metapodataka

Naziv	**Opis**
title	naslov dokumenta
keywords	ključne reči
holdingData	institucija koja čuva dokument
note	napomena
abstract	apstrakt
extendedAbstract	prošireni apstrakt

Tabela 3.2: Elementi iz druge grupe metapodataka

Slike i video zapisi smeštaju se u podstabla čiji koreni su elementi *image* i *video*, tim redosledom. Slika 3.5 predstavlja strukturu ova dva podstabla. Elementi *caption* sadrže tekstualne opise vezane za sadržaj slika, odnosno video zapisa. Element *imageData* sadrži sliku koja se prilikom smeštanja u XML dokument kodira Base64 algoritmom [32]. Video zapis se smešta u element *videoData* na isti način.



Slika 3.5: Shema NDLTD dokumenta: slike i video zapisi

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="record">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="meta">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="common">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="accessionNumber" type="xs:string"/>
                    <xs:element name="identificationNumber" type="xs:string"/>
                    <xs:element name="documentType" type="xs:string"/>
                    <xs:element name="typeOfRecord" type="xs:string"/>
                    <xs:element name="contentsCode" type="xs:string"/>
                    <xs:element name="author" type="person"/>
                    <xs:element name="menthor" type="person"/>
                    <xs:element name="languageOfText" type="xs:string"/>
                    <xs:element name="languageOfAbstract" type="xs:string"
                      maxOccurs="unbounded"/>
                    <xs:element name="countryOfPublication" type="xs:string"/>
                    <xs:element name="localityOfPublication" type="xs:string"/>
                    <xs:element name="publicationYear" type="xs:gYear"/>
                    <xs:element name="publisher" type="xs:string"/>
                    <xs:element name="physicalDescription">
                      <xs:complexType>
                        <xs:attribute name="pages" type="xs:integer" use="required"/>
                        <xs:attribute name="chapters" type="xs:integer" use="required"/>
                        <xs:attribute name="figures" type="xs:integer" use="required"/>
                        <xs:attribute name="citations" type="xs:integer" use="required"/>
                        <xs:attribute name="tables" type="xs:integer" use="required"/>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="scientificField" type="xs:string"/>
                    <xs:element name="scientificDiscipline" type="xs:string"/>
```

```
            <xs:element name="UDCnumber" type="xs:string" minOccurs="0"/>
            <xs:element name="acceptedOn" type="xs:date" minOccurs="0"/>
            <xs:element name="defendedOn" type="xs:date" minOccurs="0"/>
            <xs:element name="defendBoard">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="president" type="person"/>
                  <xs:element name="member" type="person" maxOccurs="unbounded"/>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="additional" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="title" type="xs:string"/>
            <xs:element name="keywords" type="xs:string"/>
            <xs:element name="holdingData" type="xs:string"/>
            <xs:element name="note" type="xs:string" minOccurs="0"/>
            <xs:element name="abstract" type="xs:string"/>
            <xs:element name="extendedAbstract" type="xs:string"/>
          </xs:sequence>
          <xs:attribute name="language" type="xs:token" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="text">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:base64Binary">
        <xs:attribute name="mimeType" type="xs:string" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
<xs:element name="image" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="caption" type="xs:string" minOccurs="0"
        maxOccurs="unbounded"/>
      <xs:element name="imageData" type="xs:base64Binary"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="video" minOccurs="0" maxOccurs="unbounded">
  <xs:complexType>
```

```
            <xs:sequence>
              <xs:element name="caption" type="xs:string" minOccurs="0"
                maxOccurs="unbounded"/>
              <xs:element name="videoData" type="xs:base64Binary"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="person">
    <xs:sequence>
      <xs:element name="firstName" type="xs:string"/>
      <xs:element name="middleName" type="xs:string" minOccurs="0"/>
      <xs:element name="lastName" type="xs:string"/>
      <xs:element name="degree" type="xs:string"/>
      <xs:element name="affiliation" type="xs:string"/>
      <xs:element name="title" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Listing 3.1: Shema dokumenata NDLTD kataloga

3.2.2 Funkcionalnost pronalaženja dokumenata

Nad dokumentima prethodno definisane strukture potrebno je specificirati funkcionalnost pronalaženja dokumenata. Osnovno polazište za ovu specifikaciju je omogućavanje pronalaženja po sadržaju za svaki tip medija prisutan u dokumentima.

Pretraživanje prve grupe metapodataka sadržanih u dokumentu svodi se na egzaktno poređenje sadržaja sa kriterijumom pretrage. Ovo pretraživanje ima karakter pronalaženja podataka (*data retrieval*), za razliku od pronalaženja informacija (*information retrieval*), prema definiciji ovih termina na početku prvog poglavlja.

XPath 1.0 je „jezik za adresiranje delova XML dokumenata“ [54]. Rezultat XPath izraza je skup podstabala dokumenta koja zadovoljavaju dati uslov. Iako je XPath prevashodno namenjen pretraživanju tekstualnih podataka u okviru jednog dokumenta, moguće je upotrebiti ga i za pretraživanje nad kolekcijom dokumenata. XPath jezik omogućava pretraživanje po sadržaju i po strukturi dokumenta. Moguće je formiranje aritmetičkih, logičkih i string

izraza. U kontekstu pretraživanja prve grupe metapodataka NDLTD dokumenata, XPath odgovara potrebama pretraživanja ovih elemenata.

Druga grupa metapodataka (naslov, ključne reči, apstrakt, prošireni apstrakt i napomena) predstavljaju tekstualne sadržaje koji mogu biti znatno duži i složeniji od jednostavnih sadržaja prve grupe metapodataka. Pojedinačni elementi iz ovog skupa imaju nestrukturirani tekstualni sadržaj, tako da pretraživanje po strukturi ovde nije od značaja. Sa druge strane, pretraživanje ovog tipa podataka po sadržaju zahteva veće mogućnosti od prostog egzaktnog poređenja sadržaja. Standardne mogućnosti postojećih sistema za pretraživanje nestrukturiranog teksta (tzv. *full-text* pretraživanje) obuhvataju sledeće:

- pretraživanje po rečima kao osnovnim jedinicama pretrage,
- upotreba džoker-znakova,
- pretraživanje po korenu reči (*stemming*),
- *fuzzy* pretraživanje,
- logičke, blizinske i težinske operatore i
- rangiranje pogodaka prema učestanosti ponavljanja traženog izraza u tekstu.

Originalni tekst dokumenta, iako formalno nije sastavni deo metapodataka, sa stanovišta pretraživanja može se tretirati jednako kao i druga grupa metapodataka. Isti koncepti pretraživanja primenjeni na drugu grupu metapodataka mogu se primeniti i u ovom slučaju.

Pretraživanje metapodataka NDLTD dokumenata, onako kako je do sada opisano, podrazumeva poznavanje strukture samih dokumenata. Ova struktura je vezana za dati sistem i prosečan korisnik ne mora biti upoznat sa njom. Upotreba nekog od raširenih standarda za pretraživanje tekstualnih dokumenata omogućava dostupnost NDLTD sistema širem krugu korisnika. Dialog sistemi za pronalaženje dokumenata [68] predstavljaju rašireni standard koji definiše upitni jezik i sopstvenu reprezentaciju sadržaja dokumenata. Upotreba Dialog upitnog jezika za pretraživanje metapodataka može omogućiti upotrebu sistema širem krugu korisnika. Sa stanovišta ovog upitnog jezika dokument se reprezentuje pomoću skupa *prefiksa*, određenih svojim nazivom. Svaki prefiks može imati jedan ili više sadržaja. Na primer, prefiks AU (autor

dokumenta) može imati više sadržaja i tako opisati situaciju kada dati dokument ima više autora. Pretraživanje sadržaja prefiksa u upitnom jeziku Dialog poseduje sledeće mogućnosti:

- pretraživanje po rečima kao osnovnim jedinicama pretrage,
- upotreba džoker-znakova,
- logičke i blizinske operatore.

Pretraživanje slika sadržanih u NDLTD dokumentu potrebno je izvesti kao pretraživanje po sadržaju na osnovu datog uzorka (*sample*). Prilikom formiranja kriterijuma pretrage potrebno je da postoji mogućnost definisanja važnosti pojedinih karakteristika koje se koriste u poređenju (npr. osnovne osobine slike, ili prostorni odnosi elemenata slike; v. odeljke 1.2.1 i 1.2.2).

Kako je za slike sadržane u NDLTD dokumentima vezan i propratni opisni tekst, ovaj tekst se može upotrebiti kao izvor informacija o semantici sadržaja datog slikom. Pretraživanje teksta vezanog za sliku, na način kao kod pretraživanja druge grupe metapodataka, u ovom slučaju može na posredan način obezbediti pretraživanje slika po semantici. Ovakav način pretraživanja slika po semantici razmatran je i u okviru drugih sistema (v. odeljak 1.2.3).

Pretraživanje video zapisa može se obaviti putem pretraživanja njihovih ključnih frejmova (v. odeljak 1.3). Reprezentacija sadržaja video zapisa pomoću ključnih frejmova predstavljenih statičnim slikama obrađivana je npr. u [324, 9]. Pretraživanje ovako dobijenih statičnih slika može se sprovesti na isti način kao i u prethodnom slučaju, dakle pomoću osnovnih osobina slike ili prostornih odnosa elemenata slike.

Za video zapise u NDLTD dokumentu je, analogno slikama, vezan i propratni tekst koji ima istu svrhu. Ovaj tekst može pretraživati na isti način kao i u slučaju slika.

Prethodno opisanu funkcionalnost pretraživanja pojedinih elemenata dokumenta potrebno je kombinovati i tako formirati složene kriterijume pronalaženja dokumenata. Kombinovanje pojedinačnih funkcionalnosti moguće je sprovesti u okviru odgovarajućeg modela pronalaženja dokumenata. Za potrebe NDLTD sistema mogu se upotrebiti sva tri modela pronalaženja dokumenata definisani u odeljku 2.3. Modifikovani vektorski model (v. odeljak 2.3.1) omogućava jednostavno kombinovanje pojedinačnih pretraga po raznorodnim elementima, uz upotrebu *relevance feedback* petlji za rafinaciju

rezultata. Međusobna veza pojedinačnih elementarnih upita uvek ima semantiku konjunkcije. Modifikacija proširenog bulovskog modela (v. odeljak 2.3.2) omogućava povezivanje elementarnih upita logičkim operatorima i tako na određeni način prevazilazi ograničenje vektorskog modela. Model sličnih klastera (v. odeljak 2.3.3), za razliku od prethodnih modela, omogućava praćenje prethodnih sesija pretraživanja i korekciju rezultata pretrage pomoću tih informacija tokom svojih *relevance feedback* ciklusa.

3.3 Moduli sistema

3.3.1 Apache Xindice

Pretraživanje kolekcija XML dokumenata po sadržaju pomoću XPath jezika omogućava server Apache Xindice [309]. Xindice je rezultat razvojnog projekta čiji cilj je implementacija servera za upravljanje XML bazama podataka. Pri tome se implementacija ne oslanja na sisteme za rad sa bazama podataka koji pripadaju nekom drugom modelu podataka (npr. relacionom), već se radi o tzv. *native* implementaciji. Osnovno stanovište autora ovog sistema je da postojeći modeli podataka koji se koriste u drugim sistemima za upravljanje bazama podataka ne odgovaraju u potpunosti zahtevima sistema namenjenog radu sa XML dokumentima.

Xindice sistem omogućava skladištenje, indeksiranje i pretraživanje XML dokumenata. Dokumenti se u okviru Xindice sistema smeštaju u hijerarhijski organizovane kolekcije. Pretraživanje kolekcija dokumenata obavlja se pomoću XPath jezika. Rezultat pretrage čine oni dokumenti za koje dati XPath izraz vraća neprazan skup čvorova dokumenta. Svi dokumenti su pretraživi odmah po svom skladištenju. Pretraživanje dokumenata može (ali i ne mora) da koristi indeksne struktura za povećanje efikasnosti ovog procesa.

Sistem Xindice predstavlja serversku aplikaciju koja implementira sve pomenute funkcije. Pristup klijenata ovom sistemu moguć je putem računarske mreže uz upotrebu odgovarajuće biblioteke koja implementira klijentsku stranu komunikacionog protokola.

Modul sistema XMIRS koji donosi funkcionalnost pretraživanja dokumenata pomoću XPath jezika predstavlja vezu XMIRS-a sa Xindice serverom. Funkcije Xindice modula direktno su implementirane u okviru Xindice servera.

Modul se formira kao implementacija dva potrebna interfejsa, *Indexer* (v. odeljak 2.4.4) i *Retriever* (v. odeljak 2.4.5). Konfiguracioni parametri Xindice modula prikazani su u tabeli 3.3. Parametri indeksa koje koristi Xindice modul prikazani su u tabeli 3.4.

Naziv	**Tip**	**Opis**
host	string	Adresa hosta na kome je pokrenut Xindice server
port	integer	TCP port koji je zauzeo Xindice server
rootcoll	string	naziv korenske kolekcije na serveru

Tabela 3.3: Parametri Xindice modula

Naziv	**Tip**	**Opis**
collection	string	naziv kolekcije na serveru koja se koristi za dati indeks

Tabela 3.4: Parametri indeksa koje koristi Xindice modul

Struktura upitnog dokumenta Xindice modula data je listingom 3.2. Upitni dokument sadrži samo korenski element, *xindice*, čiji sadržaj je tekst XPath upita.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <xs:element name="xindice" type="xs:string"/>
</xs:schema>
```

Listing 3.2: Struktura upitnog dokumenta Xindice modula

Rezultat pretraživanja Xindice modula je skup dokumenata koji zadovoljavaju dati XPath upit. Kako XPath jezik podrazumeva egzaktno poređenje tekstualnog sadržaja, sličnost dokumenta sa upitom je binarna vrednost. Dokumenti pronađeni tokom pretrage pomoću Xindice modula biće predstavljeni elementarnim pogocima $h_{ji} = (d_i, s_{ji})$ gde je $s_{ji} \in \{0, 1\}$.

3.3.2 Apache Lucene

Programska biblioteka Apache Lucene [166] omogućava *full-text* pretraživanje sa rangiranjem rezultata. U pitanju je samostalna biblioteka namenjena ugradnji u sisteme kojima je potrebna ovakva funkcionalnost. Za potrebe

efikasnog pretraživanja Lucene koristi indeks u obliku invertovanih datoteka [143].

Dokumenti kojima rukuje Lucene sastoje se iz više polja, gde svako polje poseduje jedan sadržaj. Sadržaj je predstavljen nestrukturiranim tekstom. Za potrebe formiranja Lucene modula za XMIRS, dokumenti se predstavljaju pomoću dva polja: (1) poljem *docid*, koje sadrži identifikator originalnog XMIRS dokumenta i ne ulazi u indeks i (2) poljem *text*, koje predstavlja sadržaj koji se indeksira.

Implementacija Lucene modula za XMIRS svodi se na implementaciju odgovarajućih programskih interfejsa. Implementacija metoda ovih interfejsa sadrži pozive funkcija Lucene biblioteke.

Lucene modul ne definiše nijedan sopstveni parametar modula. Sa druge strane, parametri indeksa koje koristi Lucene modul dati su u tabeli 3.5.

Naziv	**Tip**	**Opis**
rootdir	string	direktorijum u koji se smešta indeks
analyzer	string	naziv klase za leksičku analizu
format	string	*ascii*: tekst je u ASCII formatu *pdf*: tekst je u PDF formatu

Tabela 3.5: Parametri indeksa koje koristi Lucene modul

Struktura upitnog dokumenta Lucene modula data je listingom 3.3. Upitni dokument sadrži samo korenski element, *lucene*, čiji sadržaj je tekst upita prema Lucene sintaksi.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="lucene" type="xs:string"/>
</xs:schema>
```

Listing 3.3: Struktura upitnog dokumenta Lucene modula

Rezultat pretraživanja Lucene modula je skup dokumenata koji zadovoljavaju dati upit. Za razliku od pretraživanja u slučaju Xindice modula, ovde se rangiranje elementarnih pogodaka vrši prema učestanosti pojavljivanja traženog izraza u tekstu. Na taj način, dokumenti pronađeni tokom pretrage pomoću Lucene modula biće predstavljeni elementarnim pogocima $h_{ji} = (d_i, s_{ji})$ gde je $s_{ji} \in [0, 1]$.

3.3.3 BISIS

Neposredna upotreba Dialog jezika za pretraživanje metapodataka sadržanih u NDLTD dokumentima nije moguća. Potrebno je prethodno definisati mapiranje strukture NDLTD metapodataka na prefikse Dialog jezika. Implementacija Dialog jezika postoji u okviru tekst servera Bibliotečkog informacionog sistema BISIS [275]. Tekst server BISIS-a [187] je specijalizovani sistem za pretraživanje dokumenata strukturiranih po UNIMARC standardu [287]. Kao sastavni deo implementacije Dialog jezika u okviru BISIS-a definisano je mapiranje UNIMARC formata na Dialog prefikse. Tekst server BISIS-a može se upotrebiti kao modul XMIRS sistema ako se prethodno definiše mapiranje strukture NDLTD metapodataka na UNIMARC format. UNIMARC zapisi dobijeni kao rezultat ovog mapiranja mogu se direktno proslediti tekst serveru BISIS-a koji omogućava pretraživanje dokumenata putem Dialoga.

Tekst server BISIS-a implementiran je, za potrebe NDLTD sistema, u obliku XMIRS modula. Osnovne funkcije tekst servera obuhvataju manipulaciju i pretraživanje UNIMARC dokumenata. Modul XMIRS-a, pored toga, implementira potrebne interfejse direktnim pozivanjem odgovarajućih operacija BISIS-a, uz prethodnu konverziju NDLTD metapodataka iz XML strukture u UNIMARC format.

Dialog jezik podrazumeva egzaktno poređenje tekstualnih sadržaja prilikom pretraživanja. Na osnovu toga, elementarni rezultati pretrage $h_{ji} = (d_i, s_{ji})$ imaju binarne vrednosti za sličnost dokumenta sa upitom, odnosno $s_{ji} \in \{0, 1\}$.

Parametri indeksa koje koristi BISIS modul dati su u tabeli 3.6.

Naziv	**Tip**	**Opis**
dbtype	string	tip baze podataka; *oracle* ili *sapdb*
driver	string	naziv JDBC drajvera za bazu podataka
url	string	JDBC putanja do baze podataka
username	string	korisničko ime u okviru baze podataka
password	string	lozinka u okviru baze podataka
converter	string	naziv klase koja vrši konverziju iz XML u UNIMARC

Tabela 3.6: Parametri indeksa koje koristi BISIS modul

Struktura upitnog dokumenta BISIS modula data je listingom 3.4. Upitni dokument sadrži samo korenski element, *bisis*, čiji sadržaj je tekst upita prema Dialog sintaksi.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bisis" type="xs:string"/>
</xs:schema>
```

Listing 3.4: Struktura upitnog dokumenta BISIS modula

3.3.4 Oracle9i interMedia

Oracle9*i* [203] je sistem za upravljanje relacionim bazama podataka. Pored toga što poseduje funkcionalnost uobičajenu za ovakve sisteme, Oracle9*i* sadrži i *inter*Media, dodatak namenjen radu sa različitim tipovima medija. U okviru ovog dodatka postoji podrška za rad sa bitmapiranim slikama i audio i video zapisima. *inter*Media se oslanja na jezgro Oracle9*i* sistema koje rukuje podacima u okviru relacione baze podataka, tako da je moguće formirati baze podataka koje kombinuju klasične strukturirane i multimedijalne tipove podataka. Mogućnosti *inter*Media dodatka obuhvataju i pronalaženje slika po sadržaju.

Za potrebe pronalaženja po sadržaju, *inter*Media reprezentuje sadržaj slika pomoću vektora osobina. Upotreba vektora osobina je i najčešći pristup rešavanju ovog problema (v. odeljak 1.2.1). Karakteristike slike koje se analiziraju u okviru *inter*Media sistema su: (1) boja, (2) tekstura, (3) oblik i (4) položaj. Položaj se, kao karakteristika slike, ne može koristiti samostalno tokom pretraživanja već samo u kombinaciji sa jednom od prve tri karakteristike. Za svaku sliku skladištenu u Oracle9*i* bazu podataka sa *inter*Media dodatkom generiše se vektor osobina kao rezultat analize sadržaja slike. Analiza sadržaja počinje segmentacijom slike, tj. uočavanjem oblika popunjenih uniformnom bojom, a zatim se određuju veličine koje predstavljaju četiri pomenute karakteristike slike. Dobijeni vektor osobina sadrži između 3000 i 4000 elemenata.

Pronalaženje slika podrazumeva postojanje slike-uzorka sa kojom se porede slike sadržane u bazi podataka. Za dati uzorak se izračunava vektor osobina na isti način kao i za slike u bazi podataka. Poređenje slika vrši se poređenjem

njihovih vektora osobina. Pretraživanje baze podataka sa slikama može da koristi specijalan indeks čime se ovaj proces ubrzava.

Poređenje slika po osnovne četiri karakteristike svodi se na pojedinačno poređenje karakteristika i potom izračunavanje ukupnog rezultata. Sličnost dvaju slika po datoj karakteristici izražava se vrednošću u intervalu $[0, 100]$. Kako ova vrednost izražava rastojanje u vektorskom prostoru za datu karakteristiku, maksimalna sličnost izražena je međusobnim rastojanjem 0. Označimo sličnost po boji sa sim_{col}, sličnost po teksturi sa sim_{tex} i sličnost po obliku sa sim_{shp}.

Kriterijum pretrage, pored slike-uzorka, poseduje i težine dodeljene svakoj od četiri karakteristike slike koje definišu relativnu važnost odgovarajuće karakteristike u poređenju slika i konačnom rangiranju rezultata. Označimo težine sa w_{col}, w_{tex}, w_{shp} i w_{loc}. Dozvoljena vrednost ovih težina je u intervalu $[0, 1]$. Bar jedna od težina dodeljenih prvim trima karakteristikama mora biti veća od nule. Tokom pretrage vrednosti težina se normalizuju tako da je njihov zbir jednak 1. Ukoliko je težina dodeljena položaju $w_{loc} > 0$, izračunavanje vrednosti za sim_{col}, sim_{tex} i sim_{shp} uzimaće u obzir i položaj objekata u slici. Ukupna sličnost dvaju slika A i B izračunava se pomoću jednačine 3.1. Dobijeni rezultat nalazi se u intervalu $[0, 100]$, gde 0 označava maksimalnu sličnost.

$$sim(A, B) = w_{col} \cdot sim_{col}(A, B) + w_{tex} \cdot sim_{tex}(A, B) + w_{shp} \cdot sim_{shp}(A, B) \quad (3.1)$$

Implementacija modula XMIRS-a za pristup funkcijama *inter*Media sistema podrazumeva upotrebu *inter*Media klijentske biblioteke. Funkcije modula XMIRS-a implementiraju se pozivima odgovarajućih funkcija ove biblioteke. Pri tome se, za potrebe indeksiranja i pretraživanja pomoću *inter*Media, slike skladište u okviru Oracle9*i* relacione baze sa odgovarajućom šemom. Parametri indeksa koje koristi *inter*Media modul za XMIRS dati su u tabeli 3.7.

Naziv	**Tip**	**Opis**
type	string	MIME tip formata slike
url	string	JDBC putanja do baze podataka
username	string	korisničko ime u okviru baze
password	string	lozinka

Tabela 3.7: Parametri indeksa koje koristi *inter*Media modul

Struktura upitnog dokumenta *inter*Media modula data je listingom 3.5. Korenski element dokumenta, *intermedia*, sadrži dva podelementa: *image* i *weights*. Element *image* sadrži sliku-uzorak. Njegov atribut *contentType* sadrži oznaku MIME tipa [32] kome pripada data slika. Elementom *weights*, odnosno njegovim atributima *color*, *texture*, *shape* i *location* definišu se vrednosti težina za odgovarajuće karakteristike slike.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="intermedia">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="image">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:base64Binary">
                <xs:attribute name="contentType" type="xs:string" use="required"/>
              </xs:extension>
            </xs:simpleContent>
          </xs:complexType>
        </xs:element>
        <xs:element name="weights">
          <xs:complexType>
            <xs:attribute name="color" type="xs:float" use="required"/>
            <xs:attribute name="texture" type="xs:float" use="required"/>
            <xs:attribute name="shape" type="xs:float" use="required"/>
            <xs:attribute name="location" type="xs:float" use="optional" default="0.0"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Listing 3.5: Struktura upitnog dokumenta *inter*Media modula

3.3.5 IBM CueVideo

IBM CueVideo [62] je softverski paket namenjen analizi sadržaja video zapisa u cilju njegovog pretraživanja. Analiza video zapisa podrazumeva detekciju kadrova (što obuhvata više tipova prelaza između susednih kadrova) i prepoznavanje govora u okviru audio kanala. Podržani formati video zapisa su MPEG-1 [191] i QuickTime [219]. CueVideo poseduje i funkcionalnost

potrebnu za pretraživanje prepoznatog govora, međutim nema mogućnost pretraživanja detektovanih kadrova u video zapisima.

Rezultat detekcije kadrova nad jednim video zapisom je skup ključnih frejmova – slika u JPEG formatu [134] koje reprezentuju vizuelni sadržaj kadra (v. odeljak 1.3). Za svaki kadar generiše se po jedna slika. Dobijena slika se može upotrebiti za pretraživanje po sadržaju upotrebom nekog drugog softverskog paketa koji ima te mogućnosti. Oracle9*i* *inter*Media, prikazan u prethodnom odeljku, može poslužiti ovoj nameni.

CueVideo se, za potrebe analize video zapisa, može koristiti kao programska biblioteka namenjena ugrađivanju u druge sisteme. Modul XMIRS-a koji implementira CueVideo funkcionalnost koristi datu biblioteku. Parametri CueVideo modula dati su u tabeli 3.8. Parametri indeksa koje koristi CueVideo modul dati su u tabeli 3.9.

Naziv	**Tip**	**Opis**
programpath	string	putanja do programa za parsiranje videa

Tabela 3.8: Parametri CueVideo modula

Naziv	**Tip**	**Opis**
outputdir	string	direktorijum za smeštanje slika
policy	string	način izbora ključnog frejma: *first*: prvi frejm u kadru *middle*: srednji frejm u kadru *last*: poslednji frejm u kadru *two*: kombinacija prvog i poslednjeg frejma *three*: kombinacija prvog, srednjeg i poslednjeg

Tabela 3.9: Parametri indeksa koje koristi CueVideo modul

Kako CueVideo biblioteka ne omogućava pretraživanje generisanih ključnih frejmova, upiti upućeni ovom modulu nemaju smisla. Struktura upitnog dokumenta CueVideo modula, prikazana listingom 3.6, sadrži samo jedan element čiji je sadržaj prazan i nema atributa. Kako XMIRS modul mora da implementira funkcionalnost pretraživanja (interfejs *Retriever*, v. odeljak 2.4.5), rezultat pretraživanja CueVideo modula je definisan, ali uvek predstavlja prazan skup.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="cuevideo"/>
</xs:schema>
```

Listing 3.6: Struktura upitnog dokumenta CueVideo modula

3.4 Konfiguracija sistema

Prethodno prikazani dokumenti NDLTD sistema, potrebna funkcionalnost u pronalaženju dokumenata i moduli XMIRS-a daju osnovu za opis konfiguracije XMIRS-a koja odgovara NDLTD sistemu. Konfigurisanju, koje sprovodi administrator sistema, prethodi konstrukcija modula XMIRS-a koji predstavljaju vezu jezgra sistema sa dodatnim softverskim paketima. Konfigurisanje XMIRS-a odvija se na način opisan u odeljku 2.4.1. U narednim odeljcima data je specifikacija konfiguracije XMIRS-a za NDLTD sistem.

3.4.1 Definicije tipova dokumenata

XMIRS se, za potrebe NDLTD sistema, koristi kao sistem za pronalaženje dokumenata u okviru jednog čvora NDLTD mreže. Struktura dokumenata kojima rukuje pojedinačni čvor mreže nije nametnuta posebnom specifikacijom; svaki čvor može da koristi sopstvenu strukturu dokumenata za interne potrebe, ali se razmena dokumenata sprovodi korišćenjem unapred određene zajedničke strukture. Struktura dokumenata predstavljena u odeljku 3.2.1 namenjena je internoj upotrebi unutar jednog čvora NDLTD mreže. Prikazana struktura dokumenata ujedno definiše i jedini tip dokumenata kojima rukuje XMIRS sistem.

Definisanje tipova dokumenata kojima rukuje XMIRS se u ovom slučaju svodi na definiciju jednog tipa dokumenta. Njegova struktura data je XML Schema dokumentom sa listinga 3.1. Konfiguracioni dokument XMIRS-a kojim se definiše upotreba jedinog tipa dokumenta dat je na listingu 3.7. Interni naziv tipa dokumenata je *ndltd*, a njegova šema data je odgovarajućom URL adresom.

```
<?xml version="1.0" encoding="UTF-8"?>
<schemas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="schemas.xsd">
  <schema name="ndltd" url="file://G:/xmirs/conf/schemas/ndltd.xsd"/>
</schemas>
```

Listing 3.7: Definicija tipova dokumenata kojima rukuje XMIRS

3.4.2 Konfiguracija skladišta

Skladištenje dokumenata za potrebe XMIRS-a može biti sprovedeno na više načina. Jedan način je skladištenje kompletnih dokumenata u izvornom obliku u okviru fajl-sistema XMIRS servera. Eksterno smeštanje pojedinih elemenata dokumenata, imajući u vidu karakteristike upotrebljenih modula, nije potrebno. Ovakav način skladištenja dokumenata je jednostavan za implementaciju i omogućava jednostavno rukovanje kolekcijom dokumenata.

Sa druge strane, NDLTD sistem poseduje, pored funkcije pronalaženja dokumenata, i druge funkcije – administraciju različitih tipova korisnika i institucija-članova, kolaboraciju različitih korisnika u radu na formiranju konačnog dokumenta i drugo. Ove funkcije mogu biti implementirane u sistemu koji koristi klasičnu relacionu bazu podataka. Sastavni deo šeme ove baze podataka su i relacije koje opisuju sadržaj metapodataka u NDLTD dokumentima. Povezivanje baze podataka sa kolekcijom dokumenata kojima rukuje XMIRS moguće je sprovesti konstrukcijom specijalizovanog skladišta koje implementira mapiranje XML dokumenata kojima rukuje XMIRS na relacionu šemu koju koristi ostatak sistema. Problem mapiranja strukture XML dokumenata na relacione šeme razmatran je u literaturi i rešavan na više načina (v. odeljak 1.1.3). U ovom slučaju radi se o mapiranju konkretne šeme XML dokumenata na konkretnu relacionu šemu.

Izbor načina skladištenja dokumenata funkcionalno ne utiče na ostatak XMIRS sistema. Konfigurisanje ostatka sistema ne zavisi od izabrane implementacije skladišta. Za ovu priliku prikazana je konfiguracija skladišta koja koristi fajl-sistem, uz kompresiju sadržaja dokumenata. Konfiguracija skladišta data je dokumentom sa listinga 3.8. Skladište je implementirano klasom *TextStorage*. Upotrebljena je i trivijalna implementacija strategije skladištenja koja radi sa tačno jednim skladištem, data klasom *SimpleStrategy*.

```
<?xml version="1.0" encoding="UTF-8"?>
<registry xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="storage.xsd">
  <storages>
    <storage name="Files" class="com.gint.app.xmirs.storage.file.TextStorage">
      <param name="rootdir" value="G:\xmirs\data\docs"/>
      <param name="compress" value="yes"/>
    </storage>
  </storages>
  <strategies>
    <strategy name="Simple"
        class="com.gint.app.xmirs.storage.simple.SimpleStrategy"
        schemaName="ndltd">
      <storage name="Files"/>
    </strategy>
  </strategies>
</registry>
```

Listing 3.8: Konfiguracija skladišta

3.4.3 Konfiguracija modula

XMIRS je za potrebe NDLTD sistema dopunjen modulima koji koriste funkcije sledećih softverskih paketa: Apache Xindice, Apache Lucene, BISIS, Oracle9*i* *inter*Media i IBM CueVideo. Registrovanje ovih modula u okviru XMIRS-a i definisanje njihovih parametara vrši se dokumentom sa listinga 3.9. Prikazani dokument sadrži definicije pet modula. Prvi modul, namenjen za pristup Xindice serveru, definisan je pomoću klasa koje implementiraju odgovarajuće interfejse indeksera i pretraživača i inicijalizovan sa tri parametra. Ostali moduli, za razliku od njega, nemaju definisane inicijalizacione parametre.

```
<?xml version="1.0" encoding="UTF-8"?>
<modules xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="modules.xsd">
  <module name="xindice" content="outer">
    <param name="host" value="localhost"/>
    <param name="port" value="4080"/>
    <param name="rootcoll" value="/db/xmirs"/>
    <indexer class="com.gint.app.xmirs.modules.xindice.XindiceIndexer"/>
    <retriever class="com.gint.app.xmirs.modules.xindice.XindiceRetriever"/>
  </module>
  <module name="lucene" content="inner">
    <indexer class="com.gint.app.xmirs.modules.lucene.LuceneIndexer"/>
```

```
    <retriever class="com.gint.app.xmirs.modules.lucene.LuceneRetriever"/>
  </module>
  <module name="bisis" content="outer">
    <indexer class="com.gint.app.xmirs.modules.bisis.BisisIndexer"/>
    <retriever class="com.gint.app.xmirs.modules.bisis.BisisRetriever"/>
  </module>
  <module name="intermedia" content="inner">
    <indexer class="com.gint.app.xmirs.modules.intermedia.InterMediaIndexer"/>
    <retriever class="com.gint.app.xmirs.modules.intermedia.InterMediaRetriever"/>
  </module>
  <module name="cuevideo" content="inner">
    <param name="programpath" value="G:\CueVideo\bin\SpawnCuts.wsf"/>
    <indexer class="com.gint.app.xmirs.modules.cuevideo.CueVideoIndexer"/>
    <retriever class="com.gint.app.xmirs.modules.cuevideo.CueVideoRetriever"/>
  </module>
</modules>
```

Listing 3.9: Konfiguracija modula

3.4.4 Konfiguracija indeksa

Nakon definisanja tipova dokumenata i registracije modula može se pristupiti konfigurisanju indeksa. Definicija indeksa, data elementom *index*, obuhvata: (1) naziv indeksa, (2) naziv tipa dokumenta na koga se odnosi indeks, (3) naziv modula zaduženog za rukovanje indeksom, (4) način pozivanja indeksera, (5) specifikaciju elemenata dokumenta koji ulaze u indeks, i (6) parametre indeksa.

Dokument sa listinga 3.10 predstavlja konfiguraciju indeksa za NDLTD sistem. Prvu grupu metapodataka, sadržanih u elementu */record/meta/common*, obrađuje Xindice modul koji omogućava pretraživanje pomoću XPath upita. Drugu grupu metapodataka (*/record/meta/additional*) obrađuje Lucene modul. Ovaj modul se koristi za definisanje više odvojenih indeksa nad pojedinim elementima metapodataka (naslov, apstrakt, prošireni apstrakt, itd), kao i nad tekstualnim opisima slika i video zapisa. Lucene modul je namenjen i za rukovanje indeksom nad originalnim tekstom dokumenta datim u PDF formatu. Obe grupe metapodataka istovremeno obrađuje i BISIS modul.

Slike sadržane u NDLTD dokumentima (*/record/image/imageData*) obrađuje *inter*Media modul. Sadržaj video zapisa pretražuje se takođe pomoću *inter*Media modula, pri čemu su ulazni podaci tog modula zapravo rezultati rada CueVideo modula zaduženog za parsiranje video zapisa.

Način pozivanja indeksera ne utiče na osnovnu funkcionalnost sistema. Mogućnost izbora načina pozivanja omogućava prilagođavanje konkretnog sistema datim potrebama. Prikazana konfiguracija indeksa se, prema tome, može prilagoditi i instalacijama koje podrazumevaju više serverskih čvorova radi poboljšanja ukupnih performansi sistema, ne menjajući pri tom njegovu funkcionalnost.

```
<?xml version="1.0" encoding="UTF-8"?>
<indexes xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="indexes.xsd">
  <index name="metadata.common" schema="ndltd" module="xindice" invocation="sync">
    <params>
      <param name="collection" value="ndltd-common"/>
    </params>
    <elements>
      <element xpath="/record/meta/common"/>
    </elements>
  </index>
  <index name="title" schema="ndltd" module="lucene" invocation="sync">
    <params>
      <param name="rootdir" value="G:\xmirs\data\lucene\title"/>
      <param name="analyzer"
        value="org.apache.lucene.analysis.standard.StandardAnalyzer"/>
      <param name="format" value="ascii"/>
    </params>
    <elements>
      <element xpath="/record/meta/additional/title"/>
    </elements>
  </index>
  <index name="abstract" schema="ndltd" module="lucene" invocation="sync">
    <params>
      <param name="rootdir" value="G:\xmirs\data\lucene\abstract"/>
      <param name="analyzer"
        value="org.apache.lucene.analysis.standard.StandardAnalyzer"/>
      <param name="format" value="ascii"/>
    </params>
    <elements>
      <element xpath="/record/meta/additional/abstract"/>
    </elements>
  </index>
  <index name="ex.abstract" schema="ndltd" module="lucene" invocation="sync">
    <params>
      <param name="rootdir" value="G:\xmirs\data\lucene\exabstract"/>
      <param name="analyzer"
        value="org.apache.lucene.analysis.standard.StandardAnalyzer"/>
      <param name="format" value="ascii"/>
    </params>
    <elements>
```

```
    <element xpath="/record/meta/additional/extendedAbstract"/>
  </elements>
</index>
<index name="keywords" schema="ndltd" module="lucene" invocation="sync">
  <params>
    <param name="rootdir" value="G:\xmirs\data\lucene\keywords"/>
    <param name="analyzer"
      value="org.apache.lucene.analysis.standard.StandardAnalyzer"/>
    <param name="format" value="ascii"/>
  </params>
  <elements>
    <element xpath="/record/meta/additional/subjectKeywords"/>
  </elements>
</index>
<index name="fulltext" schema="ndltd" module="lucene" invocation="sync">
  <params>
    <param name="rootdir" value="G:\xmirs\data\lucene\fulltext"/>
    <param name="analyzer"
      value="org.apache.lucene.analysis.standard.StandardAnalyzer"/>
    <param name="format" value="pdf"/>
  </params>
  <elements>
    <element xpath="/record/text"/>
  </elements>
</index>
<index name="imagecaption" schema="ndltd" module="lucene" invocation="sync">
  <params>
    <param name="rootdir" value="G:\xmirs\data\lucene\imgcaptions"/>
    <param name="analyzer"
      value="org.apache.lucene.analysis.standard.StandardAnalyzer"/>
    <param name="format" value="ascii"/>
  </params>
  <elements>
    <element xpath="/record/image/caption"/>
  </elements>
</index>
<index name="videocaption" schema="ndltd" module="lucene" invocation="sync">
  <params>
    <param name="rootdir" value="G:\xmirs\data\lucene\vidcaptions"/>
    <param name="analyzer"
      value="org.apache.lucene.analysis.standard.StandardAnalyzer"/>
    <param name="format" value="ascii"/>
  </params>
  <elements>
    <element xpath="/record/video/caption"/>
  </elements>
</index>
<index name="bisis" schema="ndltd" module="bisis" invocation="sync">
  <params>
    <param name="dbtype" value="oracle"/>
```

```
      <param name="driver" value="oracle.jdbc.driver.OracleDriver"/>
      <param name="url" value="jdbc:oracle:thin:@localhost:1521:BIS9"/>
      <param name="username" value="bisis"/>
      <param name="password" value="bisis"/>
      <param name="converter" value="com.gint.app.xmirs.modules.bisis.NdltdConverter"/>
    </params>
    <elements>
      <element xpath="/record/meta"/>
    </elements>
  </index>
  <index name="images" schema="ndltd" module="oracle" invocation="async">
    <params>
      <param name="type" value="image/jpeg"/>
      <param name="url" value="jdbc:oracle:thin:@localhost:1521:BIS9"/>
      <param name="username" value="intmed"/>
      <param name="password" value="intmed"/>
    </params>
    <elements>
      <element xpath="/record/image/imageData"/>
    </elements>
  </index>
  <index name="videos" schema="ndltd" module="oracle" invocation="async">
    <params>
      <param name="type" value="image/jpeg"/>
      <param name="url" value="jdbc:oracle:thin:@localhost:1521:BIS9"/>
      <param name="username" value="cue"/>
      <param name="password" value="cue"/>
    </params>
    <index name="videos.1" schema="ndltd" module="cuevideo" invocation="sync">
      <params>
        <param name="outputdir" value="G:\xmirs\data\cuevideo"/>
        <param name="policy" value="first"/>
      </params>
      <elements>
        <element xpath="/record/video/videoData"/>
      </elements>
    </index>
  </index>
</indexes>
```

Listing 3.10: Konfiguracija indeksa

3.4.5 Konfiguracija modela

Modeli pronalaženja dokumenata koje može da upotrebljava XMIRS definišu se posebnim konfiguracionim dokumentom. Sadržaj ovog dokumenta za slučaj primene na NDLTD sistem dat je listingom 3.11. Definisana su tri

modela pronalaženja koji predstavljaju implementacije modela definisanih u odeljku 2.3. Svaki model je, sa stanovišta konfiguracije, predstavljen klasom koja implementira interfejs *RetrievalModel* (v. odeljak 2.4.5). Implementacija vektorskog modela, kao svoje parametre, ima tri veličine koje figurišu u jednačini 2.6 za RF ciklus. Model sličnih klastera koristi relacionu bazu podataka za smeštanje podataka o rezultatima pronalaženja. Parametri ovog modela opisuju konekciju sa bazom podataka. Pored ovih parametara, parametar *minDistance* definiše minimalno rastojanje klastera koje predstavlja kriterijum zaustavljanja algoritma za klasterovanje.

```
<?xml version="1.0" encoding="UTF-8"?>
<models xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="models.xsd">
  <model name="vector" class="com.gint.app.xmirs.models.vector.VectorModel">
    <param name="alpha" value="1"/>
    <param name="beta" value="1"/>
    <param name="gamma" value="-1"/>
  </model>
  <model name="extbool" class="com.gint.app.xmirs.models.extbool.BooleanModel"/>
  <model name="cluster" class="com.gint.app.xmirs.models.cluster.ClusterModel">
    <param name="url" value="jdbc:oracle:thin:@localhost:1521:BIS9"/>
    <param name="username" value="clusmod"/>
    <param name="password" value="clusmod"/>
    <param name="minDistance" value="0.1"/>
  </model>
</models>
```

Listing 3.11: Konfiguracija modela

3.5 Okruženje implementacije

Implementacija XMIRS sistema sastoji se od implementacije jezgra, prikazane u odeljku 2.5, i implementacije prikazanih pet modula koji se koriste za potrebe pretraživanja NDLTD dokumenata. Prototip je konfigurisan dokumentima prikazanim u ovom poglavlju.

U ovom odeljku prikazana je osnovna struktura i najvažnije karakteristike implementacije modula. Paketi iz kojih se sastoje moduli su navedeni u tabeli 3.10. Svi navedeni paketi se nalaze u okviru korenskog paketa *com.gint.app.xmirs*.

Instalacija prototipa se sastoji iz odgovarajuće strukture direktorijuma, prikazane tabelom 2.3. Ova struktura proširena je novim direktorijumima koje koriste pojedini moduli. Novi direktorijumi su pobrojani u tabeli 3.11.

Naziv	Opis
modules.bisis	Modul za pristup BISIS-u
modules.cuevideo	Modul za pristup CueVideo parseru
modules.intermedia	Modul za pristup Oracle9i interMedia serveru
modules.lucene	Modul za pristup Lucene biblioteci
modules.xindice	Modul za pristup Xindice serveru

Tabela 3.10: Paketi iz kojih se sastoje moduli XMIRS-a

Naziv	Opis
data/cuevideo	Privremeni direktorijum za CueVideo modul
data/lucene	Skladištenje Lucene indeksa

Tabela 3.11: Direktorijumi koje koriste moduli prototipa

Implementacije BISIS i *inter*Media modula koriste Oracle9i sistem za upravljanje bazama podataka. Šema baze podataka BISIS modula objavljena je u [187]. Šema baze podataka koju koristi *inter*Media modul prikazana je na slici 3.6.

IMAGES

kolona	*tip*
image_id	INTEGER
doc_id	VARCHAR2(255)
xpath	VARCHAR2(255)
image	ORDSYS.Image
image_sig	ORDSYS.ImageSignature

SEARCH

kolona	*tip*
image_id	INTEGER
image	ORDSYS.Image
image_sig	ORDSYS.ImageSignature

Slika 3.6: Shema baze podataka *inter*Media modula

Relacija *IMAGES* namenjena je za skladištenje slika koje ulaze u indeks. Kolona *image* predstavlja originalni sadržaj slike. Kolona *image_sig* predstavlja vektor osobina slike koji se koristi tokom pretraživanja. Relacija *SEARCH* namenjena je za privremeno skladištenje slike-uzorka tokom procesa pretrage.

3.6 Primeri korišćenja

Ovaj odeljak sadrži primere upita koji se mogu postaviti XMIRS sistemu. Svi primeri podrazumevaju prethodno prikazanu konfiguraciju XMIRS-a i

upotrebu razvijenog prototipa. Dobijeni rezultati upita predstavljaju rezultat rada prototipa nad odgovarajućim skupom dokumenata.

Pronalaženje dokumenata pomoću XMIRS-a poseduje i funkcionalnost klasičnih sistema za pretraživanje tekstualnih dokumenata koji se zasnivaju na egzaktnom poređenju teksta.

Primer 3.1 *Potrebno je pronaći dokumente koji su pisani na engleskom jeziku.*

Podatak o jeziku na kome je pisan originalni dokument nalazi se u elementu */record/meta/common/languageOfText* čiji sadržaj pripada kontrolisanom rečniku. Element rečnika koji predstavlja engleski jezik glasi *eng*. Element *languageOfText* ulazi u sastav indeksa pod nazivom *metadata.common*, za koji je zadužen modul *xindice*. Upit će biti upućen modulu u formi XPath izraza. Kako su kod upita za Xindice modul svi pogoci jednako rangirani, izbor modela pronalaženja dokumenata nije od značaja. Za ovaj primer izabran je vektorski model, sa L_∞ metrikom i bez limita na maksimalan broj pronađenih dokumenata. Upitni dokument dat je listingom 3.12.

```
<?xml version="1.0" encoding="UTF-8"?>
<vectorQuery>
  <elemQueries>
    <elemQuery index="common.metadata">
      <xindice>/common[languageOfText=&apos;eng&apos;]</xindice>
    </elemQuery>
  </elemQueries>
  <metric>L_inf</metric>
  <maxHits>-1</maxHits>
</vectorQuery>
```

Listing 3.12: Upit za primer 3.1

□

XMIRS omogućava kombinovanje tekstualnih upita koji odgovaraju različitim njegovim modulima, odnosno različitim modelima pretraživanja teksta.

Primer 3.2 *Potrebno je pronaći dokumente pisane na engleskom jeziku, u čijem apstraktu se javlja fraza „information systems“.*

Tekst apstrakta vezanog za originalni dokument nalazi se među metapodacima u elementu */record/meta/additional/abstract*. Upiti koji traže datu frazu (niz reči) u tekstu apstrakta mogu se formulisati pomoću modula *bisis*, odnosno indeksa *bisis*. Deo kriterijuma pretrage vezan za jezik originalnog dokumenta može se formulisati kao i u prethodnom primeru. Oba upotrebljena modula, Xindice i BISIS, jednako rangiraju sve pogotke. Upotrebljen je isti vekorski model pronalaženja dokumenata kao i u prethodnom primeru. Jednak rezultat dobija se i upotrebom proširenog bulovskog modela kada se elementarni upiti kombinuju operatorom AND_∞. Upitni dokument kojim se formuliše dati upit prikazan je na listingu 3.13.

```
<?xml version="1.0" encoding="UTF-8"?>
<vectorQuery>
  <elemQueries>
    <elemQuery index="common.metadata">
      <xindice>/[languageOfText=&apos;eng&apos;]</xindice>
    </elemQuery>
    <elemQuery index="bisis">
      <bisis>AB=information [W1] AB=systems</bisis>
    </elemQuery>
  </elemQueries>
  <metric>L_inf</metric>
  <maxHits>-1</maxHits>
</vectorQuery>
```

Listing 3.13: Upit za primer 3.2

□

Pored tekstualnih upita koji podrazumevaju egzaktno poređenje teksta i ne omogućavaju rangiranje rezultata pronalaženja, u XMIRS-u je moguće i kombinovati upite koji omogućavaju rangiranje rezultata.

Primer 3.3 *Potrebno je pronaći dokumente u čijem se proširenom apstraktu javlja fraza „information systems“, a u naslovu dokumenta reč „XML“. Rangirati pogotke prema učestalosti pojavljivanja datih izraza.*

Za razliku od prethodnog primera, gde su korišćeni moduli koji ne vrše rangiranje pogodaka, u ovom slučaju potrebno je upotrebiti modul *lucene* koji to omogućava. Konfiguracijom sistema definisana su dva posebna indeksa nad proširenim apstraktom (element */record/meta/additional/extendedAbstract*)

i naslovom (element */record/meta/additional/title*). Oba indeksa koriste Lucene modul. Konačno rangiranje pogodaka obaviće se u okviru vektorskog modela, pri čemu će korišćena metrika biti L_2, a maksimalan broj prikazanih pogodaka iznosi 100. Upitni dokument kojim se formuliše dati upit prikazan je na listingu 3.14.

```
<?xml version="1.0" encoding="UTF-8"?>
<vectorQuery>
  <elemQueries>
    <elemQuery index="ex.abstract">
      <lucene>&quot;information systems&quot;</lucene>
    </elemQuery>
    <elemQuery index="title">
      <lucene>XML</lucene>
    </elemQuery>
  </elemQueries>
  <metric>L_2</metric>
  <maxHits>100</maxHits>
</vectorQuery>
```

Listing 3.14: Upit za primer 3.3

□

Kombinovanje upita koji sadrže multimedijalne tipove podataka u okviru kriterijuma pretrage podrazumeva mogućnost rangiranja rezultata. Prilikom formulisanja upita korisnik često nije u stanju da precizno izrazi svoju potrebu za informacijama pa je potrebno omogućiti reformulaciju upita na osnovu analize inicijalnih rezultata pronalaženja. Upotreba vektorskog modela sa *relevance feedback* ciklusima u okviru XMIRS-a obezbeđuje ovu funkcionalnost.

Primer 3.4 *Potrebno je pronaći dokumente u kojima se u apstraktu pojavljuje reč „Seychelles“, u tekstualnom opisu vezanom za sliku fraza „sandy beach“, a slika prikazuje peščanu plažu sa pogledom na pučinu u vreme zalaska sunca. Kako korisnik nije siguran u važnost pojedinih elemenata kriterijuma pretrage, potrebno je omogućiti pregled najbolje rangiranih dokumenata i unos informacija o njihovoj relevantnosti. Korigovati upit potreban broj puta.*

Apstrakt dokumenta odnosno element */record/meta/additional/abstract* pretraživ je pomoću indeksa *abstract* kojim rukuje modul *lucene*. Tekstualni opis slike nalazi se u elementu */record/image/caption* nad kojim je kreiran

indeks *imagecaption* pomoću modula *lucene*. Sama slika se nalazi u elementu */record/image/imageData*, nad kojim je generisan indeks *images* pomoću modula *intermedia*.

Treća komponenta kriterijuma pretrage podrazumeva posedovanje slike-uzorka sa kojom će se porediti slike iz dokumenata u kolekciji. Za sliku koja predstavlja snimak peščane plaže u sumrak potrebno je odrediti važnost osnovnih karakteristika slike koje se koriste tokom poređenja. Boje sadržane u slici imaju veliku važnost u poređenju – očekuje se da tražene slike sadrže slične boje (more, nebo, pesak) i to i u sličnom prostornom rasporedu u okviru slike. Oblik figura koje predstavljaju pojedine objekte na slici manje je važan. Tekstura slike nije od značaja. Na osnovu ove analize izabrane su težine za svaku od karakteristika slike: $w_{col} = 0.8$, $w_{tex} = 0.0$, $w_{shp} = 0.2$, $w_{loc} = 0.8$.

Tri navedene komponente kriterijuma pretrage kombinovaće se u okviru vektorskog modela. Upotrebljena metrika biće L_2, a rezultat će sadržati 20 najbolje rangiranih dokumenata. Inicijalni upit dat je dokumentom sa listinga 3.15. Iz prikazanog dokumenta izostavljen je sadržaj elementa *image*, koji predstavlja sadržaj slike-uzorka kodiran Base64 algoritmom.

```
<?xml version="1.0" encoding="UTF-8"?>
<vectorQuery>
  <elemQueries>
    <elemQuery index="abstract">
      <lucene>Seychelles</lucene>
    </elemQuery>
    <elemQuery index="title">
      <lucene>&quot;sandy beach&quot;</lucene>
    </elemQuery>
    <elemQuery index="images">
      <intermedia>
        <image contentType="image/jpeg">...</image>
        <weights color="0.8" texture="0.0" shape="0.2" location="0.8"/>
      </intermedia>
    </elemQuery>
  </elemQueries>
  <metric>L_2</metric>
  <maxHits>50</maxHits>
</vectorQuery>
```

Listing 3.15: Inicijalni upit za primer 3.4

Struktura dokumenta koji predstavlja rezultat upita zavisi od izabranog modela pronalaženja. U ovom primeru koristi se vektorski model, tako da

rezultat izgleda kao na listingu 3.16, pri čemu su tu navedena samo prva tri pogotka.

```
<?xml version="1.0" encoding="UTF-8"?>
<vectorResult>
  <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/21" similarity="0.0721353"/>
  <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/24" similarity="0.1543244"/>
  <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/17" similarity="0.2598943"/>
  ...
</vectorResult>
```

Listing 3.16: Inicijalni rezultat za primer 3.4

Po pregledu primljenih pogodaka korisnik šalje informaciju o njihovoj relevantnosti. Struktura ovog dokumenta takođe zavisi od izabranog modela, a listing 3.17 predstavlja njegov sadržaj za tekući primer.

```
<?xml version="1.0" encoding="UTF-8"?>
<relevanceFeedback>
  <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/21" relevant="yes"/>
  <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/24" relevant="yes"/>
  <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/17" relevant="no"/>
  ...
</relevanceFeedback>
```

Listing 3.17: *Relevance feedback* za primer 3.4

Na osnovu podataka o relevantnosti pronađenih dokumenata, upit se reformuliše na osnovu jednačine 2.6 (v. odeljak 2.3.1). Novi rezultat se vraća korisniku u obliku dokumenta čija struktura je identična inicijalnom rezultatu prikazanom na listingu 3.16. Iterativni postupak se ponavlja potreban broj ciklusa, sve dok korisnik ne bude zadovoljan rezultatom.

□

Povezivanje rezultata rada dva modula u okviru prototipa je ilustrovano povezivanjem rada CueVideo i *inter*Media modula. CueVideo modul predstavlja video parser koji generiše ključne frejmove na osnovu analize sadržaja video zapisa. Upotreba *inter*Media modula za pretraživanje ovako dobijenih ključnih frejmova omogućava korisniku da na sličan način tretira statične slike i video zapise sa stanovišta pronalaženja dokumenata.

Primer 3.5 *Potrebno je pronaći dokumente u kojima se u apstraktu pojavljuje reč „Fellini“, ili gde pridruženi video zapis sadrži kadar koji prikazuje ulicu sa biciklistom.*

Pošto upit sadrži disjunkciju, koristiće se prošireni bulovski model. Apstrakt dokumenta, odnosno element */record/meta/additional/abstract*, pretraživ je pomoću indeksa *abstract* kojim rukuje modul *lucene*. Video zapis, sadržan u elementu */record/video/videoData* obuhvaćen je indeksom *videos.1* kojim rukuje modul *cuevideo*. Pretraživanje korišćenjem ovog modula neće dati rezultate (modul ne omogućava pretraživanje), nego se rezultat indeksiranja ovog modula upućuje *intermedia* modulu. Ovaj modul se koristi za pretraživanje upotrebom indeksa *videos*.

```
<?xml version="1.0" encoding="UTF-8"?>
<boolQuery>
  <operator type="OR" metric="L_2" maxHits="100">
    <elemQuery index="abstracts">
      <lucene>Fellini</lucene>
    </elemQuery>
    <elemQuery index="videos">
      <intermedia>
        <image type="image/jpeg">...</image>
        <weights color="0.8" texture="0.0" shape="0.2" location="0.8"/>
      </intermedia>
    </elemQuery>
  </operator>
</boolQuery>
```

Listing 3.18: Upit za primer 3.5

□

Model sličnih klastera (v. odeljak 2.3.3) omogućava korišćenje rezultata iz ranijih sesija pronalaženja dokumenata. Ukoliko su u ranijim sesijama pronalaženja od strane korisnika identifikovani skupovi dokumenata koji su relevantni za određenu temu, buduće sesije mogu da iskoriste ove informacije poređenjem rezultata ranijih sesija sa rezultatom tekuće sesije.

Primer 3.6 *Potrebno je pronaći dokumente u kojima se u apstraktu pojavljuje fraza „lung disease“, a pridružena slika predstavlja rentgenski snimak pluća koji je sličan datom uzorku. Tokom formiranja konačnog skupa rezultata potrebno je uzeti u obzir i rezultate ranijih sesija pronalaženja.*

Korišćenje modela sličnih klastera podrazumeva i *relevance feedback* ciklus u kome korisnik analizira rezultate ranijih sesija i na osnovu njih modifikuje

rezultat pronalaženja u tekućoj sesiji. Kada je korisnik zadovoljan rezultatom, sadržaj rezultata se trajno čuva za potrebe narednih sesija.

```
<?xml version="1.0" encoding="UTF-8"?>
<clusterQuery>
  <elemQueries>
    <elemQuery index="abstracts">
      <lucene>&quot;lung disease&quot;</lucene>
    </elemQuery>
    <elemQuery index="images">
      <intermedia>
        <image type="image/jpeg">...</image>
        <weights color="0.2" texture="0.0" shape="0.8" location="1.0"/>
      </intermedia>
    </elemQuery>
  </elemQueries>
</clusterQuery>
```

Listing 3.19: Inicijalni upit za primer 3.6

Inicijalni rezultat predstavljen je dokumentom sa listinga 3.20. Rezultat sadrži nekoliko klastera čiji sadržaj je ovde dat u skraćenom obliku.

```
<?xml version="1.0" encoding="UTF-8"?>
<clusterResult>
  <cluster id="77">
    <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/22" distance="0.54029614"/>
    ...
  </cluster>
  <cluster id="92">
    <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/61" distance="0.70710677"/>
    ...
  </cluster>
  <cluster id="89">
    <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/141" distance="0.8660254"/>
    ...
  </cluster>
  ...
</clusterResult>
```

Listing 3.20: Inicijalni rezultat za primer 3.6

Dokument kojim se zahteva sadržaj svih klastera koji su rezultat upita nad istim skupom indeksa (razlika složenih upita je 0), a koji sadrže dokumente iz klastera tekućeg rezultata sa identifikatorom 92 dat je listingom 3.21.

```
<?xml version="1.0" encoding="UTF-8"?>
<clusterHistory>
  <getSimilar clusterID="92" level="0"/>
</clusterHistory>
```

Listing 3.21: Zahtev za klasterima koji su slični klasteru sa identifikatorom 92

Rezultat prethodno postavljenog zahteva ima strukturu identičnu inicijalnom rezultatu i prikazan je na listingu 3.22.

```
<?xml version="1.0" encoding="UTF-8"?>
<clusterResult>
  <cluster id="81">
    <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/261" distance="0.70710677"/>
    <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/241" distance="0.70710677"/>
  </cluster>
  <cluster id="79">
    <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/242" distance="0.73615205"/>
    <hit docid="http://branko.tmd.ns.ac.yu/docs/ndltd/262" distance="0.73615205"/>
  </cluster>
</clusterResult>
```

Listing 3.22: Rezultat zahteva za sličnim klasterima

Nakon pregleda dokumenata iz traženih klastera, korisnik je u mogućnosti da promeni sadržaj relevantnog klastera iz tekuće sesije. Dokument sa listinga 3.23 definiše modifikaciju klastera sa identifikatorom 92 dodavanjem dokumenta br. 242 i uklanjanjem dokumenta br. 81 i proglašavanjem klastera br. 79 jednakim sa klasterom br. 92 čime se klaster br. 79 uklanja iz trajne memorije.

```
<?xml version="1.0" encoding="utf-8"?>
<clusterFeedback>
  <cluster id="91">
    <add docid="http://branko.tmd.ns.ac.yu/docs/ndltd/242"/>
    <remove docid="http://branko.tmd.ns.ac.yu/docs/ndltd/81"/>
    <sameAs id="79"/>
  </cluster>
</clusterFeedback>
```

Listing 3.23: Modifikacija klastera sa identifikatorom 92

□

Poglavlje 4

Zaključak

Predmet istraživanja disertacije pripada oblasti pronalaženja informacija (*information retrieval*, IR). Istraživanja u ovoj oblasti bave se pre svega problemima pronalaženja dokumenata kao osnovnih nosilaca informacija. Pojam dokumenata u oblasti IR razvijao se od nestrukturiranih tekstualnih dokumenata do strukturiranih multimedijalnih dokumenata. Model proširivog sistema koji omogućava pronalaženje multimedijalnih dokumenata je osnovna tema ove disertacije. Proširivost prikazanog sistema ima dva aspekta: (1) mogućnosti proširivanja sistema modulima koji omogućavaju pretraživanje različitih tipova medija i (2) mogućnosti proširivanja sistema različitim modelima pronalaženja dokumenata.

Sistematizacija dostupne literature, izvšena u prvom poglavlju, definiše četiri osnovna pravca istraživanja u oblasti IR. Pronalaženje tekstualnih dokumenata predstavlja najstariji pravac istraživanja. Noviji pravci bave se problemima pronalaženja slika, video zapisa i multimedijalnih objekata. Pronalaženje multimedijalnih objekata se u literaturi tretira na dva načina: sa jedne strane nalaze se istraživanja koja multimedijalne objekte tretiraju kao nezavisne pojave različitih tipova medija. Sa druge strane, postoje istraživanja koja multimedijalne objekte tretiraju kao složene objekte koji poseduju elementarne multimedijalne sadržaje organizovane u određenu strukturu. Ovakav tretman multimedijalnih objekata odgovara pojmu multimedijalnih dokumenata.

Problemu pronalaženja strukturiranih multimedijalnih dokumenata posvećen je relativno mali broj istraživanja, imajući u vidu ukupan broj objavljenih

rezultata koji se odnose na pronalaženje multimedijalnih objekata shvaćenih u širem smislu. Postojeći dostupni rezultati u ovoj oblasti definišu modele strukturiranih multimedijalnih dokumenata i modele njihovog pronalaženja koji, po pravilu, nameću svoj model sadržaja i/ili svoj model pronalaženja za sve podržane tipove medija.

Drugo poglavlje predstavlja centralni deo disertacije. U njemu je definisan model proširivog sistema za pronalaženje multimedijalnih dokumenata (*extensible multimedia information retrieval system*, XMIRS). Formalna specifikacija modela obuhvata model dokumenata XMIRS-a, modele pronalaženja dokumenata koji su prilagođeni XMIRS-u i softversku arhitekturu sistema koja omogućava proširivost sistema različitim modulima za pronalaženje multimedijalnih objekata i različitim modelima pronalaženja dokumenata.

Dokumenti kojima rukuje XMIRS su XML dokumenti, čiji tip se definiše odgovarajućim XML Schema dokumentima. Model dokumenata XMIRS-a zasniva se na upotrebi XML-a kao jezika za reprezentaciju dokumenata.

Modeli pronalaženja dokumenata predstavljaju okvir u kome se vrši izračunavanje rezultata za dati upit u procesu pronalaženja. Vektorski i prošireni bulovski modeli nastali su na osnovu modela koji postoje u klasičnim tekstualnim IR sistemima i prilagođeni su okruženju i konceptima XMIRS-a. Model sličnih klastera predstavlja novi model pronalaženja dokumenata koji omogućava analizu rezultata ranijih sesija pronalaženja i, na osnovu toga, modifikaciju rezultata tekuće sesije.

Softverska arhitektura sistema polazi od osnovnog koncepta proširivosti XMIRS-a. Arhitekturom je specificirano jezgro sistema koje se, dopunjeno modulima, može prilagoditi potrebama konkretne instalacije sistema. Jezgro je proširivo, osim različitim modulima, i različitim modelima pronalaženja dokumenata. XMIRS je dekomponovan na četiri podsistema: skladištenje dokumenata, rukovanje dokumentima, indeksiranje i pronalaženje dokumenata. Na kraju drugog poglavlja prikazano je okruženje implementacije prototipa jezgra XMIRS-a.

Treće poglavlje sadrži verifikaciju prikazanog modela XMIRS-a na realnom primeru digitalne biblioteke doktorskih i magistarskih teza (NDLTD). Prikazana je struktura XML dokumenata kojima rukuje NDLTD sistem i definisani su zahtevi u pogledu funkcionalnosti pronalaženja dokumenata koje je potrebno ispuniti. Za implementaciju pojedinačnih zahteva iskorišćeni su postojeći

raznorodni softverski paketi koji su u XMIRS integrisani u formi njegovih modula. Verifikacija podrazumeva i implementaciju ovih modula i njihovu integraciju u prototip jezgra XMIRS-a prikazan na kraju drugog poglavlja.

Implementirani prototip XMIRS-a konfigurisan je na odgovarajući način tako da podrži funkcije pronalaženja dokumenata u NDLTD sistemu. Pored prikaza konfiguracije, treće poglavlje sadrži i primere upotrebe XMIRS-a u prikazanom ambijentu. Primeri upotrebe demonstriraju polazne ciljeve u formiranju modela: (1) mogućnost upotrebe različitih modela pronalaženja dokumenata i (2) mogućnost kombinovanog pretraživanja različitih tipova medija pomoću odgovarajućih modula.

Na osnovu detaljne analize dostupne literature, date u prvom poglavlju, može se zaključiti da postojeći modeli pronalaženja multimedijalnih dokumenata imaju neke od sledećih nedostataka:

- dokumenti kojima se rukuje imaju unapred određenu, fiksnu strukturu,
- podržan je ograničen broj tipova medija, bez mogućnosti proširivanja modela novim tipovima, i
- nameće se sopstveni model reprezentacije sadržaja i pronalaženja za sve tipove medija.

Osnovni doprinos disertacije dat je modelom XMIRS sistema koji prevazilazi navedene nedostatke. Model XMIRS-a karakterišu sledeće osobine:

- mogućnost upotrebe različitih modela pronalaženja dokumenata,
- mogućnost upotrebe različitih postojećih rešenja za pretraživanje objekata koji pripadaju različitim tipovima medija, sa mogućnošću proširivanja sistema podrškom za nove tipove,
- konfigurabilnost sistema koja omogućava prilagođavanje konkretne implementacije potrebama korisnika sa aspekta rukovanja dokumentima i njihovog pronalaženja i
- kombinovanje funkcionalnosti pojedinih modula u celinu koja donosi veće mogućnosti u pronalaženju dokumenata nego pojedinačni moduli upotrebljeni zasebno.

Prikazana prototipska implementacija koja ispunjava ciljeve u pogledu funkcionalnosti postavljene pred XMIRS predstavlja potvrdu praktične vrednosti predloženog modela.

Analiza performansi IR sistema je u ambijentu klasičnih nestrukturiranih tekstualnih dokumenata obavljena u velikom broju dosadašnjih istraživanja. Međutim, u slučaju multimedijalnih dokumenata ovakvih analiza do sada nije bilo pre svega zbog nepostojanja opšteprihvaćenog modela multimedijalnih dokumenata. XMIRS sistem može se upotrebiti kao osnova za analizu performansi pojedinih modela pronalaženja multimedijalnih dokumenata, uz pretpostavku da se usvoji prikazani model dokumenata. Rezultati ovog procesa mogu biti polazna tačka u razvoju sistema koji, na osnovu analize performansi postojećih modela pronalaženja, može korisniku sugerisati onaj model koji se u ranijim sesijama pokazao kao najbolji, odnosno može pružiti pomoć korisniku tokom formulacije odgovarajućeg upita kojim će se zadovoljiti njegova potreba za informacijama.

Bibliografija

[1] Kjersti Aas and Line Eikvil. A Survey on: Content-based Access to Image and Video Databases, 1997. http://citeseer.nj.nec.com/aas97survey.html.

[2] K. Aberer and W. Klas. Supporting Temporal Multimedia Operations in Object-Oriented Database Systems. In *IEEE Intl. Conference on Multimedia Computer Systems*, 1994.

[3] S. Abiteboul, S. Cluet, V. Christophides, T. Milo, G. Moerkotte, and J. Siméon. Querying Documents in Object Databases. *International Journal on Digital Libraries*, 1(1):5–19, 1997.

[4] S. Abiteboul, L. Segoufin, and V. Vianu. Representing and Querying XML with Incomplete Information. In *Proceedings of 20th Symposium on Principles of Database Systems*, pp. 150–161, 2001.

[5] Donald A. Adjeroh and Kingsley C. Nwosu. Multimedia Database Management – Requirements and Issues. *IEEE Multimedia*, 4(3):24–33, 1997.

[6] Rakesh Agrawal, Christos Faloutsos, and Arun Swami. Efficient Similarity Search in Sequence Databases. In *Proceedings of Fourth Intl. Conference on Foundations of Data Organization and Algorithms*, pp. 69–84, 1993. http://-olympos.cs.umd.edu/pub/TechReports/fodo.ps.

[7] Wasfi Al-Khatib, Y. Francis Day, Arif Ghafoor, and P. Bruce Berra. Semantic Modeling and Knowledge Representation in Multimedia Databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):64–80, 1999.

[8] Maria Grazia Albanesi, Marco Ferretti, and Alessandro Giancane. A Compact Wavelet Index for Retrieval in Image Database. In *IEEE 10th Intl. Conference on Image Analysis and Processing*, 1999.

[9] E. Ardizzone, M. La Cascia, and D. Mionelli. Motion and Colour-Based Video Indexing and Retrieval. In *Intl. Conference on Pattern Recognition (ICPR)*, volume C, pp. 135–139, 1996.

[10] E. Ardizzone, G. A. M. Gioiello, M. La Cascia, and D. Molinelli. A Real-Time Neural Approach to Scene Cut Detection. In *Proceedings of IS&T/SPIE – Storage & Retrieval for Image and Video Databases IV*, 1996.

[11] Y. Ariki and T. Teranishi. Indexing and Classification of TV News Articles Based on Telop Recognition. In *IEEE 4th Intl. Conference on Document Analysis and Recognition*, pp. 422–427, 1997.

[12] Yasuo Ariki, Yoshiaki Sugiyama, and Noriyuki Ishikawa. Face Indexing on Video Data: Extraction, Recognition, Tracking and Modeling. In *IEEE 3rd Intl. Conference on Face and Gesture Recognition*, 1998.

[13] T. Arnold-Moore, M. Fuller, and R. Sacks-Davis. Approaches for Structured Document Management. In *Proceedings of Markup Technologies*, 1999.

[14] Y. Alp Aslandogan, Chuck Thier, and Clement Yu. A System for Effective Content-Based Image Retrieval. In *Proceedings of the 4th ACM Intl. Conference on Multimedia*, 1996.

[15] Y. Alp Aslandogan, Chuck Thier, Clement T. Yu, Chengwen Liu, and Krishnakumar R. Nair. Design, Implementation, and Evaluation of SCORE (A System for Content-Based Retrieval of Pictures). In *Proceedings of the 11th IEEE Intl. Conference on Data Engineering*, pp. 280–287, 1995.

[16] Y. Alp Aslandogan, Chuck Thier, Clement T. Yu, Jon Zou, and Naphtali Rishe. Using Semantic Contents and WordNet in Image Retrieval. In *Proceedings of SIGIR-97, 20th ACM Conference on Research and Development in Information Retrieval*, pp. 286–295, 1997.

[17] Solomon Atnafu, Lionel Brunie, and Harald Kosch. Similarity-Based Algebra for Multimedia Database Systems. In *Proceedings of the 12th Australasian conference on Database technologies*, pp. 115–122, 2001.

[18] Yannis S. Avrithis, Nikolaos D. Doulamis, Anastasios D. Doulamis, and Stefanos D. Kollias. Efficient Content Representation in MPEG Video Databases. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pp. 91–95, 1998.

[19] R. Baeza-Yates and G. Navarro. XQL and Proximal Nodes. In *Proceedings of the XML Workshop of 23rd ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000.

[20] Ricardo Baeza-Yates. A Hybrid Query Model for Full Text Retrieval Systems. Technical Report DCC-1994-2, Dept. of Computer Science, Univ. of Chile, 1994.

[21] Ricardo Baeza-Yates, Benjamín Bustos, Edgar Chávez, Norma Herrera, and Gonzalo Navarro. Clustering in Metric Spaces with Applications to Information Retrieval. In W. Wu, H. Xiong, and S. Shekhar, editors, *Information Retrieval and Clustering*. Kluwer Academic Press, 2002.

[22] Ricardo Baeza-Yates and Gonzalo Navarro. Integrating Contents and Structure in Text Retrieval. *ACM SIGMOD Record*, 25(1):67–79, 1996.

[23] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[24] N. Beckmann, H.-P. Kriegel, R. Schneider, and B. Seeger. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. In *Proceedings of ACM SIGMOD Conference on Management of Data*, pp. 322–331, 1990.

[25] Nicholas J. Belkin and W. Bruce Croft. Information Filtering and Information Retrieval: Two Sides of the Same Coin? *Communications of the ACM*, 35(12):29–38, 1992.

[26] Stefan Berchtold, Daniel A. Keim, and Hans-Peter Kriegel. The X-tree: An Index Structure for High-Dimensional Data. In *Proceedings of the Intl. Conference on Very Large Databases*, pp. 28–39, 1996.

[27] T. Berners-Lee, R. Fielding, and L. Masinter. *Uniform Resource Identifiers (URI): Generic Syntax*. IETF RFC 2396. http://www.ietf.org/rfc/rfc2396.txt.

[28] Elisa Bertino, Fausto Rabitti, and Simon Gibbs. Query Processing in a Multimedia Document System. *ACM Transactions on Office Information Systems*, 6(1):1–41, 1988.

[29] M. Bichsel and A. Pentland. Human Face Recognition and the Face Image Set's Topology. *CVGIP: Image Understanding*, 59(2):254–261, 1994.

[30] P. V. Biron and A. Malhotra. *XML Schema Part 2: Datatypes*. W3C Recommendation, 2001. http://www.w3.org/TR/xmlschema-2.

[31] A. Bonifati and S. Ceri. A Comparative Study of Five XML Query Languages. *ACM SIGMOD Record*, 29(1):68–79, 2000.

[32] N. Borenstein and N. Freed. *MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies*. IETF RFC 1341. http://www.ietf.org/rfc/rfc1341.txt.

[33] F. Bourel, C. C. Chibelushi, and A. A. Low. Robust Facial Expression Recognition Using a State-Based Model of Spatially-Localised Facial Dynamics. In *Proceedings of the 5th IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pp. 113–118, 2002.

[34] Ronald Bourret. Mapping DTDs to Databases. http://www.xml.com/pub/a/2001/05/09/dtdtodbs.html, 2001.

[35] J. Boyer. *Canonical XML Version 1.0.* W3C Recommendation, 2001. http://www.w3.org/TR/2001/REC-xml-c14n-20010315/.

[36] Ben Bradshaw. Semantic Based Image Retrieval: A Probabilistic Approach. In *Proceedings of MM-2000, 8th ACM International Conference on Multimedia*, pp. 167–176, 2000.

[37] T. Bray, J. Paoli, C. M. Sperber-McQueen, and E. Maler. *Extensible Markup Language (XML) 1.0 (Second Edition).* W3C Recommendation, 2000. http://www.w3.org/TR/REC-xml.

[38] Tim Bray, Dave Hollander, and Andrew Layman. *Namespaces in XML.* W3C Recommendation, 1999. http://www.w3.org/TR/1999/REC-xml-names-19990114/.

[39] Anne Brink, Sherry Marcus, and V. S. Subrahmanian. Heterogenous Multimedia Reasoning. *IEEE Computer*, 28(9):33–39, 1995.

[40] Roberto Brunelli and Omella Mich. Image Retrieval By Examples. *IEEE Transactions on Multimedia*, 2(3), 2000.

[41] M. Caliani, C. Colombo, A. D. Bimbo, and P. Pala. Computer Analysis of TV Spots: The Semiotics Perspective. In *IEEE Intl. Conference on Multimedia Computing and Systems*, 1998.

[42] Marc Cavazza, Roger Green, and Ian Palmer. Multimedia Semantic Features and Image Content Description. In *Proceedings of IEEE Multimedia Modeling*, pp. 39–44, 1998.

[43] D. Chamberlin, J. Clark, D. Florescu, J. Robie, J. Siméon, and M. Stefanescu. *XQuery 1.0: An XML Query Language.* W3C Working Draft, 2001. http://www.w3.org/TR/2001/WD-xmlquery-req-20010215/.

[44] S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang. An Eigenspace Update Algorithm for Image Analysis. *GMIP: Graphical Models and Image Processing Journal*, 59(5):321–332, 1997.

[45] C. Chang and S. Lee. Retrieval of Similar Pictures on Pictorial Databases. *Pattern Recognition*, 24(7):675–680, 91.

[46] S.-K. Chang, Q.-Y. Shi, and C.-W. Yan. Iconic Indexing by 2-D Strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):413–428, 1987.

[47] Y. Chen, X. S. Zhou, and T. S. Huang. One-Class SVM for Learning in Image Retrieval. In *Proceedings of IEEE Intl. Conference on Image Processing*, 2001.

[48] T. T. Chinenyanga and N. Kushmerick. Expressive and Efficient Ranked Queries for XML Data. In *Proceedings of 4th International Workshop on the Web and Databases (WebDB'01)*, 2001.

[49] S. Christodoulakis, M. Theodoridou, F. Ho, M. Papa, and A. Pathria. Multimedia Document Presentation, Information Extraction, and Document Formation in MINOS: A Model and a System. *ACM Transactions on Office Information Systems*, 4(4):345–383, 1986.

[50] S. Christodoulakis, J. van der Broek, J. Li, T. Li, S. Wan, Y. Wang, M. Papa, and E. Bertino. Development of a Multimedia Information System for an Office Environment. In *Proceedings of 10th VLDB Conference on Very Large Databases*, pp. 261–271, 1984.

[51] V. Christophides, S. Abiteboul, S. Cluet, and M. Scholl. From Structured Documents to Novel Query Facilities. *ACM SIGMOD Record*, 23(2):313–324, 1994.

[52] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José Luis Maroquín. Searching in Metric Spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.

[53] James Clark. *XSL Transformations (XSLT) 1.0*. W3C Recommendation, 1999. http://www.w3.org/TR/xslt.

[54] James Clark and Steven DeRose. *XML Path Language (XPath) Version 1.0*. W3C Recommendation, 1999. http://www.w3.org/TR/xpath.

[55] C. Clarke, G. Cormack, and F. Burkowski. An Algebra for Structured Text Search and a Framework for its Implementation. *The Computer Journal*, 1995.

[56] C. Clarke, G. Cormack, and F. Burkowski. Schema-Independent Retrieval from Heterogenous Structured Text. In *Proceedings of 4th Annual Symposium on Document Analysis and Information Retrieval*, 1995.

[57] Jonathan D. Courtney. Automatic, Object-Based Indexing for Assisted Analysis of Video Data. In *Proceedings of ACM Conference on Multimedia*, pp. 423–424, 1996.

[58] John Cowan and Richard Tobin. *XML Information Set*. W3C Recommendation, 2001. http://www.w3.org/TR/xml-infoset/.

[59] W. Bruce Croft. Experiments with Representation in a Document Retrieval System. *Information Technology: Research and Development*, 2(1):1–21, 1983.

[60] W. Bruce Croft and D. J. Harper. Using Probabilistic Models of Retrieval Without Relevance Information. *Journal of Documentation*, 35(4):285–295, 1979.

[61] W. Bruce Croft, Howard R. Turtle, and David D. Lewis. The Use of Phrases and Structured Queries in Information Retrieval. In *Proceedings of SIGIR-91, 14th ACM Conference on Research and Development in Information Retrieval*, pp. 32–45, 1991.

[62] *CueVideo*. IBM Almaden Research Center. http://www.almaden.ibm.com/-projects/cuevideo.shtml.

[63] Young Francis Day, Serhan Dagtas, Mitsutoshi Iino, Ashfaq Khokhar, and Arif Ghafoor. Object-Oriented Conceptual Modeling of Video Data. In *Proceedings of IEEE Conference on Data Engineering (ICDE)*, pp. 401–408, 1995.

[64] S. DeRose, E. Maler, and R. Daniel Jr. *XML Pointer Language (XPointer)*. W3C Recommendation, 1999. http://www.w3.org/TR/xptr.

[65] S. DeRose, E. Maler, and D. Orchard. *XML Linking Language (XLink)*. W3C Recommendation, 2001. http://www.w3.org/TR/xlink.

[66] B. Desai, P. Goyal, and S. Sadri. A Data Model for Use with Formatted and Textual Data. *Journal of the American Society for Information Science*, 37(3):158–165, 1986.

[67] A. Deutsch, M. Fernandez, D. Florescu, A. Y. Levy, and D. Suciu. *XML-QL: A Query Language for XML*. Submission to W3C, 1998. http://www.w3.org/TR/NOTE-xml-ql/.

[68] *Successful Searching with Dialog*. Dialog Corporation. http://www.dialog.com.

[69] I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks, Theory and Applications*. John Wiley & Sons, 1996.

[70] Nevenka Dimitrova and Forouzan Golshani. Rx for semantic video database retrieval. In *Proceedings of ACM Conference on Multimedia*, pp. 219–226, 1994.

[71] Nevenka Dimitrova and Forouzan Golshani. Motion recovery for video content classification. *ACM Transactions on Information Systems*, 13(4):408–439, 1995.

[72] Nevenka Dimitrova, Thomas McGee, and Herman Elenbaas. Video Keyframe Extraction and Filtering: A Keyframe is not a Keyframe to Everyone. In *Proceedings of ACM 6th Intl. Conference on Information and Knowledge Management (CIKM'97)*, pp. 113–120, 1997.

[73] Irfan A. Essa and Alex P. Pentland. Facial Expression Recognition using a Dynamic Model and Motion Energy. In *Proceedings of the 5th IEEE Intl. Conference on Computer Vision*, pp. 360–367, 1995.

[74] C. Y. R. Chen et al. Design of a Multimedia Object-Oriented DBMS. *Multimedia Systems*, 3(5–6):217–227, 1995.

[75] M. T. Ozsu et al. An Object-Oriented Multimedia Database System for a News-on-Demand Application. *Multimedia Systems*, 3(5–6):182–203, 1995.

[76] Emmanuel Etiévent, Frank Lebourgeois, and Jean-Michel Jolion. Assisted Video Sequences Indexing: Motion Analysis Based on Interest Points. In *IEEE 10th Intl. Conference on Image Analysis and Processing*, 1999.

[77] B. Everitt. *Cluster Analysis*. Halsted Press, 1980.

[78] C. Faloutsos and K. Lin. FastMap: A Fast Algorithm for Indexing, Data Mining and Visualization of traditional and multimedia. In *Proceedings of ACM SIGMOD-95*, pp. 163–174, 1995.

[79] Christos Faloutsos. Multimedia IR: Indexing and Searching. In Ricardo Baeza-Yates and Berthier Ribeiro-Neto, editors, *Modern Information Retrieval*, pp. 345–365. ACM Press / Addison Wesley, 1999.

[80] Christos Faloutsos, Ron Barber, Myron Flickner, Jim Hafner, Wayne Niblack, Dragutin Petkovic, and William Equitz. Efficient and Effective Querying by Image Content. *Journal of Intelligent Information Systems*, 3(3/4):231–262, 1994.

[81] W. Fan and L. Libkin. On XML Integrity Constraints in the Presence of DTDs. In *Proceedings of 20th Symposium on Principles of Database Systems*, pp. 114–125, 2001.

[82] Li Fang and Ang Yew Hock. Image Retrieval with Relevance Feedback. In *Proceedings of the 29th Applied Imagery Pattern Recognition Workshop*, 2000.

[83] Daniel Fasulo. An Analysis of Recent Work on Clustering Algorithms. Technical Report 01-03-02, Dept. of Computer Science and Engineering, University of Washington, 1999.

[84] H. Fawcett. *PAT 3.3 User's Guide*. UW Centre for New OED and Text Research, Univ. of Waterloo, 1989.

[85] Mary Fernández, Ashok Malhotra, Jonathan March, Marton Nagy, and Norman Walsh. *XQuery 1.0 and XPath 2.0 Data Model*. W3C Working Draft, 2002. http://www.w3.org/TR/query-datamodel/.

[86] R. Fielding, J. Gettys, J.Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. IETF RFC 2616. http://www.ietf.org/rfc/rfc2616.txt.

[87] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petković, David Steele, and Peter Yanker. Query by Image and Video Content: The QBIC System. *IEEE Computer*, 28(9):23–32, 1995.

[88] D. Florescu and D. Kossmann. Storing and Querying XML Data Using an RDBMS. *IEEE Data Engineering Bulletin*, 22(3):27–34, 1999.

[89] D. Forsyth, J. Malick, M. Fleck, H. Greenspan, T. Leung, S. Belengie, C. Carson, and C. Bregler. Finding Pictures in Large Collections of Images. Technical Report CSD96-905, Univ. of California, Berkeley, 1996.

[90] Edward A. Fox. Characterization of Two New Experimental Collections in Computer and Information Science Containing Textual and Bibliographical Concepts. Technical report, NCSTRL, 1983.

[91] William B. Frakes and Ricardo Baeza-Yates, editors. *Information Retrieval: Data Structures & Algorithms*. Prentice Hall, 1992.

[92] J. M. Francos, A. Zvi Meiri, and B. Porat. A Unified Texture Model Based on a 2-D Wold Like Decomposition. *IEEE Transactions on Signal Processing*, 41:2665–2678, 1993.

[93] B. V. Funt and G. D. Finlayson. Color Constant Color Indexing. Technical Report 91-09, School of Computer Science, Simon Fraser University, Vancouver, 1991.

[94] G. W. Furnas, S. Deerwester, S. T. Dumais, T. K. Landauer, R. A. Harshman, L. A. Streeter, and K. E. Lochbaum. Information Retrieval Using a Singular Value Decomposition Model of Latent Semantic Structure. In *Proceedings of the 11th ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 465–480, 1988.

[95] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1994.

[96] Michael Garey and David Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, 1979.

[97] Simon Gibbs. Composite Multimedia and Active Objects. In *Proceedings of Intl. Conference on Object-Oriented Programming: Systems, Languages, and Applications*, pp. 97–112, 1991.

[98] G. Gonnet. Examples of PAT Applied to the Oxford English Dictionary. Technical Report OED-87-02, UW Centre for New OED and Text Research, Univ. of Waterloo, 1987.

[99] Andrew Graves and Mounia Lalmas. Video Retrieval Using an MPEG-7 Based Inference Network. In *Proceedings of SIGIR'02, 25th ACM Conference on Research and Development in Information Retrieval*, pp. 339–346, 2002.

[100] W. I. Grosky and Y. Tao. Multimedia Data Mining and its Implication for Query Processing. In R. R. Wagner, editor, *Proceedings of IEEE Workshop on Database and Expert Systems and Applications (DEXA)*, 1998.

[101] Lifang Gu and Don Bone. Skin Colour Region Detection in MPEG Video Sequences. In *IEEE 10th International Conference on Image Analysis and Processing*, 1999.

[102] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, and Henrik Frystyk Nielsen. SOAP Version 1.2 Part 1: Messaging Framework. W3C Proposed Recommendation. http://www.w3.org/TR/soap12-part1.

[103] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, and Henrik Frystyk Nielsen. SOAP Version 1.2 Part 2: Adjuncts. W3C Proposed Recommendation. http://www.w3.org/TR/soap12-part2.

[104] Venkat N. Gudivada and Vijay V. Raghavan. Design and Evaluation of Algorithms for Image Retrieval by Spatial Similarity. *ACM Transactions on Information Systems*, 13(2):115–144, 1995.

[105] Eugene J. Guglielmo and Neil C. Rowe. Natural-Language Retrieval od Images Based on Descriptive Captions. *ACM Transactions on Information Systems*, 14(3):237–267, 1996.

[106] Amarnath Gupta and Ramesh Jain. Visual Information Retrieval. *Communications of the ACM*, 40(5):71–79, 1997.

[107] Armanath Gupta, Simone Santini, and Ramesh Jain. In Search of Information in Visual Media. *Communications of the ACM*, 40(12):35–42, 1997.

[108] Antonin Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In *Proceedings of ACM SIGMOD Conference on Management of Data*, pp. 47–57, 1984.

[109] S. W. Haas. A Feasibility Study of the Case Hierarchy Model for the Construction and Porting of Natural Language Interfaces. *Information Processing & Management*, 26(5):615–628, 1991.

[110] Mohand-Said Hacid, Cyril Decleir, and Jacques Kouloumdjian. A Database Approach for Modeling and Querying Video Data. *IEEE Transactions on Knowledge and Data Engineering*, 12(5):729–750, 2000.

[111] J. Hafner, H. S. Sawhney, W. Equitz, M. Flickner, and W. Niblack. Efficient Color Histogram Indexing for Quadratic Form Distance Functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):729–736, 1995.

[112] David Haines and W. Bruce Croft. Relevance Feedback and Inference Networks. In *Proceedings of the SIGIR-93, 16th ACM Conference on Research and Development in Information Retrieval*, pp. 2–11, 1993.

[113] M. Hammer and D. MacLeod. Database Description with SDM: A Semantic Database Model. *ACM Transactions on Database Systems*, 6(3):351–386, 1981.

[114] Donna Harman. Relevance feedback revisited. In *Proceedings of the SIGIR-92, 15th ACM Conference on Research and Development in Information Retrieval*, pp. 1–10, 1992.

[115] Donna K. Harman. Overview of the Third Text Retrieval Conference. In *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*, pp. 1–19, 1995.

[116] D. J. Harper and Cornelis J. van Rijsbergen. An Evalation of Feedback in Document Retrieval Using Co-occurence Data. *Journal of Documentation*, 34(3):189–216, 1978.

[117] A. G. Hauptmann and M. A. Smith. Text, Speech and Vision for Video Segmentation: The Informedia Project. In *AAAI-95 Fall Symp. on Computational Models for Integrating Language and Vision*, 1995.

[118] Ji He, Ah-Hwee Tan, Chew-Lim Tan, and Sam-Yuan Sung. On Quantitative Evaluation of Clustering Systems. In W. Wu, H. Xiong, and S. Shekhar, editors, *Information Retrieval and Clustering*. Kluwer Academic Press, 2002.

[119] Qin He. A Review of Clustering Algorithms as Applied in IR. Technical Report UIUCLIS-1999/6+IRG, Univ. of Illinois at Urbana-Champaign, 1999.

[120] Marti A. Hearst and Jan O. Pedersen. Reexamining the Cluster Hypothesis: Scatter/Gather on Retrieval Results. In *Proceedings of SIGIR-96, 19th ACM Conference on Research and Development in Information Retrieval*, pp. 76–84, 1996.

[121] Silvia Hollfelder, André Everts, and Ulrich Thiel. Concept-Based Browsing in Video Libraries. In *IEEE Forum on Research and Technology Advances in Digital Libraries*, 1999.

[122] P. Hong, Q. Tian, and T. S. Huang. Incorporate Support Vector Machines to Content-Based Image Retrieval with Relevance Feedback. In *Proceedings of IEEE Intl. Conference on Image Processing*, 2000.

[123] Arnaud Le Hors, Phillipe Le Hégaret, Lauren Wood, Gavin Nicol, Jonathan Robie, Mike Champion, and Steve Byrne. *Document Object Model (DOM) Level 2 Core Specification Version 1.0.* W3C Recommendation, 2000. http://www.w3.org/TR/DOM-Level-2-Core.

[124] P. R. Hsu and H. Harashima. Detecting Scene Changes and Activities in Video Databases. In *IEEE Intl. Conference on Acoustics, Speech, and Signal Processing*, volume 5, pp. 33–36, 1994.

[125] E. Ide. New Experiments in Relevance Feedback. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 337–354. Prentice-Hall, 1971.

[126] International Standards Organization. *Information Processing – Text and Office Systems – Office Document Architecture (ODA) and Interchange Format*, 1986. ISO/DIS 8613.

[127] International Standards Organization. *Information Processing – Text and Office Systems – Standard Generalized Markup Language (SGML)*, 1986. ISO 8879-1986.

[128] H. Ishikawa, F. Suzuki, F. Kozakura, A. Makinouchi, M. Miyagishima, M. Aoshima, Y. Izumida, and Y. Yamane. The Model, Language, and Implementation of an Object-Oriented Multimedia Knowledge Base Management System. *ACM Transactions on Database Systems*, 18:1–50, 1993.

[129] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Query Databases Through Multiple Examples. In *Proceedings of Intl. Conference on Very Large Databases*, pp. 218–227, 1998.

[130] A. K. Jain, M. N. Murty, and P. J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[131] Anil K. Jain and Aditya Vailaya. Image Retrieval Using Color and Shape. *Pattern Recognition*, 29(8):1233–1244, 1996.

[132] Haitao Jiang and Ahmed K. Elmagarmid. WVTDB – A Semantic Content-Based Video Database System on the World Wide Web. *IEEE Transactions on Knowledge and Data Engineering*, 10(6):947–966, 1998.

[133] Haitao Jiang, Danilo Montesi, and Ahmed K. Elmagarmid. VideoText Database Systems. In *Proceedings of IEEE Intl. Conference on Multimedia Computing and Systems*, pp. 344–351, 1997.

[134] *Digital Compression and Coding of Continuous-Tone Still Images.* ISO/IEC 10918-3. http://www.jpeg.org/public/spiff.pdf.

[135] Ibrahim Kamel and Christos Faloutsos. Hilbert R-tree: An Improved R-tree Using Fractals. In *Proceedings of the Intl. Conference on Very Large Databases*, pp. 500–509, 1994.

[136] T. Kanade, S. Satoh, and Y. Nakamura. Accessing Video Contents: Cooperative Approach Between Image and Natural Language Processing. In *International Symposium on Research, Development and Practice in Digital Libraries (ISDL'97)*, pp. 143–150, 1997.

[137] Marcin Kaszkiel, Justin Zobel, and Ron Sacks-Davis. Efficient Passage Ranking for Document Databases. *ACM Transactions on Information Systems*, 17(4):406–439, 1999.

[138] Norio Katayama and Shin'ichi Satoh. The SR-tree: An Index Structure for High-Dimensional Nearest Neighbor Queries. In *Proceedings of ACM SIGMOD Conference on Management of Data*, pp. 369–380, 1997.

[139] P. Kilpeläinen and H. Manilla. Retrieval from Hierarchical Texts by Partial Patterns. In *Proceedings of SIGIR-93, 16th ACM Conference on Research and Development in Information Retrieval*, pp. 214–222, 1993.

[140] P. Kilpeläinen and H. Manilla. Ordered and Unordered Tree Inclusion. *SIAM Journal on Computing*, 24(2):340–356, 1995.

[141] Satoshi Kimura and Masahiko Yachida. Facial Expression Recognition and Its Degree Estimation. In *Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 295–300, 1997.

[142] M. Kirby and L. Sirovich. Application of the Karhunen-Loève Procedure for the Characterization of Human Faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108, 1990.

[143] Donald E. Knuth. *Sorting and Searching*, volume 3 of *The Art of Computer Programming*. Addison Wesley, Second edition, 1998.

[144] Andrej Kovčić and Branko Milosavljević. Data Model for Indexing and Searching XML Documents. *Novi Sad Journal of Mathematics*, 31(1):151–158, 2000.

[145] Tony C. T. Kuo and Arbee L. P. Chen. A Content-Based Query Language for Video Databases. In *Proceedings of IEEE Intl. Conference on Multimedia Computing and Systems (ICMCS)*, pp. 209–214, 1996.

[146] Young H. Kwon and Niels da Vitoria Lobo. Age Classification from Facial Images. *Computer Vision and Image Understanding*, 74(1):1–21, 1999.

[147] J. Laaksonen, M. Koskela, and E. Oja. PicSOM: Self-Organizing Maps for Content-Based Image Retrieval. In *Intl. Joint Conference on Neural Networks (IJCNN'99)*, 1999.

[148] Andreas Laux and Lars Martin. *XUpdate – XML Update Language*. XML:DB Working Draft. http://www.xmldb.org/xupdate/xupdate-wd.html.

[149] N. Leavitt. Whatever Happened to Object-Oriented Databases? *IEEE Computer*, 33(8):16–19, 2000.

[150] S.-Y. Lee and F.-J. Hsu. 2D C-String: A New Spatial Knowledge Representation for Image Database Systems. *Pattern Recognition*, 23(10):1077–1087, 1990.

[151] S.-Y. Lee, M.-K. Shan, and W.-P. Wang. Similarity Retrieval of Iconic Image Databases. *Pattern Recognition*, 22(6):675–682, 1989.

[152] Taekyong Lee, Lei Sheng, Tolga Bozkaya, Nevzat Hurkan Balkir, Z. Meral Özsoyoglu, and Gultekin Özsoyoglu. Querying Multimedia Presentations Based on Content. *IEEE Transactions on Knowledge and Data Engineering*, 11(3):361–385, 1999.

[153] Anton Leuski. Evaluating Document Clustering for Interactive Information Retrieval. In *Proceedings of CIKM'01, 10th ACM Conference on Information and Knowledge Management*, pp. 33–40, 2001.

[154] John Z. Li, Tamer Öszu, and Duane Safron. Modeling of Video Spatial Relationships in an Object Database Management System. In *Proceedings of IEEE Intl. Workshop on Multimedia Database Management Systems*, pp. 124–132, 1996.

[155] John Z. Li, Tamer Öszu, and Duane Safron. Modeling of Moving Objects in a Video Database. In *Proceedings of IEEE Intl. Conference on Multimedia Computing and Systems*, pp. 336–343, 1997.

[156] W. Li, S. Gauch, J. Gauch, and K. M. Pua. VISION: A Digital Video Library. In *Proceedings of ACM Digital Libraries*, 1996.

[157] Rainer Lienhart, Silvia Pfeiffer, and Wolfgang Effelsberg. Video Abstracting. *Communications of the ACM*, 40(12):55–62, 1997.

[158] Reiner Lienhart. Indexing and Retrieval of Digital Video Sequences based on Automatic Text Recognition. In *Proceedings of ACM Conference on Multimedia*, 1996.

[159] King-Ip Lin, H. V. Jagadish, and Christos Faloutsos. The TV-tree — An Index Structure for High-Dimensional Data. *VLDB Journal*, 3:517–542, 1994.

[160] Fang Liu and Rosalind W. Picard. Periodicity, Directionality, and Randomness: Wold Features for Image Modeling and Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):722–733, 1996.

[161] H. C. Liu and G. L. Zick. Scene Decomposition of MPEG Compressed Video. In *Digital Video Compression: Algorithms and Technologies*, volume SPIE 2419, pp. 26–37, 1995.

[162] K. L. Liu, P. Sistla, C. Yu, and N. Rishe. Query Processing in a Video Retrieval System. In *IEEE 14th International Conference on Data Engineering*, 1998.

[163] R. M. Losee and A. Bookstein. Integrating Boolean Queries in Conjunctive Normal Form with Probabilistic Retrieval Models. *Information Processing & Management*, 24(3):315–321, 1988.

[164] A. Lu, M. Ayoub, and J. Dong. Ad Hoc Experiments Using EUREKA. In *TREC-96*, 1996.

[165] Ye Lu, Chunhui Hu, Xingquan Zhu, and Hongjiang Zhang Qiang Yang. A Unified Framework for Semantics and Feature-Based Relevance Feedback in Image Retrieval Systems. In *Proceedings of ACM Conference on Multimedia*, pp. 31–37, 2000.

[166] *Apache Lucene*. The Apache Software Foundation. http://jakarta.apache.org/-lucene.

[167] Carol Lundquist, David A. Grossman, and Ophir Frieder. Improving Relevance Feedback in the Vector Space Model. In *Proceedings of CIKM-97, 6th ACM Conference on Information and Knowledge Management*, pp. 16–23, 1997.

[168] W. J. Ma and B. S. Manjunath. Texture Features and Learning Similarity. In *Proceedings of the 1996 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 425–430, 1996.

[169] Sean D. MacArthur, Carla E. Brodley, and Chi-Ren Shyu. Relevance Feedback Decision Trees in Content-Based Image Retrieval. In *IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'00)*, 2000.

[170] I. MacLeod. Storage and Retrieval of Structured Documents. *Information Processing & Management*, 26(2):197–208, 1990.

[171] I. MacLeod. A Query Language for Retrieving Information from Hierarchic Text Structures. *The Computer Journal*, 34(3):254–264, 1991.

[172] M. K. Mandal and T. Aboulnasr. Fast Wavelet Histogram Techniques for Image Indexing. *Computer Vision and Image Understanding*, 75(1/2):99–110, 1999.

[173] B. S. Manjunath and W. Y. Ma. Texture Features for Browsing and Retrieval of Image Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–842, 1996.

[174] J. Mao and A. K. Jain. Texture Classification and Segmentation Using MultiResolution Simultaneous Autoregressive Models. *Pattern Recognition*, 25(2):173–188, 1992.

[175] Sherry Marcus and V. S. Subrahmanian. Foundations of Multimedia Database Systems. *Journal of the ACM*, 43(3):474–523, 1996.

[176] Y. Masunaga. Multimedia Databases: A Formal Framework. In *Proceedings of IEEE CS Office Automation Symposium*, pp. 36–45, 1987.

[177] Katsuhiro Matsuno, Chil-Woo Lee, Satoshi Kimura, and Saburo Tsuji. Automatic Recognition of Human Facial Expressions. In *Proceedings of the 5th IEEE Intl. Conference on Computer Vision*, pp. 352–359, 1995.

[178] Carlo Meghini. An Image Retrieval Model Based on Classical Logic. In *Proceedings of SIGIR-95, 18th ACM Conference on Research and Development in Information Retrieval*, pp. 300–308, 1995.

[179] Carlo Meghini, Fausto Rabitti, and Costantino Thanos. Conceptual Modeling of Multimedia Documents. *IEEE Computer*, 24(10):23–30, 1991.

[180] Carlo Meghini, Fabrizio Sebastiani, and Umberto Straccia. A Model of Multimedia Information Retrieval. *Journal of the ACM*, 48(5):909–970, 2001.

[181] Carlo Meghini, Fabrizio Sebastiani, Umberto Straccia, and Constantino Thanos. A Model of Information Retrieval based on a Terminological Logic. In *Proceedings of SIGIR-93, 16th ACM Conference on Research and Development in Information Retrieval*, pp. 298–307, 1993.

[182] Carlo Meghini and Umberto Straccia. A Relevance Terminological Logic for Information Retrieval. In *Proceedings of SIGIR-96, 19th ACM Conference on Research and Development in Information Retrieval*, pp. 197–205, 1996.

[183] C. Meilhac and C. Nastar. Relevance Feedback and Category Search in Image Databases. In *Proceedings of IEEE Intl. Conference on Multimedia Computing and Systems*, pp. 512–517, 1999.

[184] J. Meng, Y. Juan, and S. F. Chang. Scene Change Detection in a MPEG Compressed Video Sequence. In *Digital Video Compression: Algorithms and Technologies*, volume SPIE 2419, pp. 14–25, 1995.

[185] Klaus Meyer-Wegener. Multimedia Databases: Integrated Storage and Retrieval of Text, Images, Sound, and Video. Technical Report IMMD 25(12), Institut für Mathematische Maschinen und Datenverarbeitung, Friedrich-Alexander-Universität Erlangen-Nürnberg, 1992.

[186] G. A. Miller, R. Beckwith, C. Felbaum, D. Gross, and K. Miller. Introduction to WordNet: An Online Lexical Database. *International Journal of Lexicography*, 3(4):235–244, 1990.

[187] Branko Milosavljević. Tekst server UNIMARC zapisa. Master's thesis, Fakultet tehničkih nauka, Novi Sad, 1999.

[188] Branko Milosavljević and Zora Konjović. Design of an XML-Based Extensible Multimedia Information Retrieval System. In *Proceedings of IEEE Multimedia Software Engineering 2002*, pp. 114–121, 2002.

[189] Farzin Mokhtarian. Silhouette-Based Isolated Object Recognition through Curvature Scale Space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):539–554, 1995.

[190] Farzin Mokhtarian, Sadegh Abbasi, and Josef Kittler. Efficient and Robust Retrieval by Shape Content through Curvature Scale Space. In *Proceedings of International Workshop on Image Databases and MultiMedia Search*, pp. 35–42, 1996.

[191] *Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s.* ISO/IEC 11172-1:1993. http://mpeg.-telecomitalialab.com/standards/mpeg-1/mpeg-1.htm.

[192] A. Nagasaka and Y. Tanaka. Automatic Video Indexing and Full Video Search for Object Appearances. In W. E. Knuth, editor, *IFIP Trans., Visual Database Systems II*, pp. 119–133. North-Holland, 1992.

[193] Gonzalo Navarro and Ricardo Baeza-Yates. A Language for Queries on Structure and Contents of Textual Databases. In *Proceedings of SIGIR-95, 18th ACM Conference on Research and Development in Information Retrieval*, pp. 93–101, 1995.

[194] Gonzalo Navarro and Ricardo Baeza-Yates. Proximal Nodes: A Model to Query Document Databases by Content and Structure. *ACM Transactions on Information Systems*, 15(4):400–435, 1997.

[195] *Networked Digital Library of Theses and Dissertations.* http://www.ndltd.org.

[196] *NDLTD Union Catalog.* VTLS Inc. http://www.vtls.com/ndltd.

[197] Surya Nepal, Uma Srinivasan, and Graham Reynolds. Automatic Detection of 'Goal' Segments in Basketball Videos. In *Proceedings of ACM Conference on Multimedia*, pp. 261–269, 2001.

[198] W. Niblack and R. Barber. The QBIC Project: Querying Images by Content Using Color, Texture, and Shape. In *Storage and Retrieval for Image and Video Databases*, 1993.

[199] J. Nievergelt and H. Hinterberger. The Grid File: An Adaptable, Symmetric Multikey File Structure. *ACM Transactions on Database Systems*, 9(1):38–71, 1984.

[200] Kingsley C. Nwosu, Bhavani Thuraisingham, and P. Bruce Berra. Multimedia Database Systems—A New Frontier. *IEEE Multimedia*, 4(3):21–23, 1997.

[201] Y. Ogawa, T. Morita, and K. Kobayashi. A Fuzzy Document Retrieval System Using the Keyword Connection Matrix and a Learning Method. *Fuzzy Sets and Systems*, 39:163–179, 1991.

[202] Virginia E. Ogle and Michael Stonebreaker. Chabot: Retrieval from a Relational Database of Images. *IEEE Computer*, 28(9):40–48, 1995.

[203] *Oracle9i Database*. Oracle Corporation. http://www.oracle.com/ip/deploy/-database/oracle9i/.

[204] Dimitris Papadias, Marios Mantzourogiannis, Panos Kalnis, Nikos Mamoulis, and Ishfaq Amad. Content-Based Retrieval using Heuristic Search. In *Proceedings of SIGIR-99, 22nd ACM Conference on Research and Development in Information Retrieval*, pp. 168–175, 1999.

[205] G. Pass, R. Zabih, and J. Miller. Comparing Images using Color Coherence Vectors. In *ACM Intl. Conference on Multimedia*, pp. 65–73, 1996.

[206] *Portable Document Format Reference Manual*. Adobe Systems Inc., 1999. http://partners.adobe.com/asn/developer/acrosdk/DOCS/pdfspec.pdf.

[207] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[208] A. Pentland, R. Picard, and S. Sclaroff. Photobook: Content-based Manipulation of Image Databases. In *Storage and Retrieval for Image and Video Databases II*, 1994.

[209] M. Petkovic and W. Jonker. Content-Based Video Retrieval by Integrating Spatio-Temporal ans Stochastic Recognition of Events. In *Proceedings of IEEE Workshop on Detection and Recognition of Events in Video*, 2001.

[210] E. G. M. Petrakis and S. C. Orphanoudakis. Methodology for the Representation Indexing, and Retrieval of Images by Content. *Image and Vision Computing*, 11(8):504–521, 1993.

[211] Euripides G. M. Petrakis and Christos Faloutsos. Similarity Searching in Medical Image Databases. *IEEE Transactions on Knowledge and Data Engineering*, 9(3):435–447, 1997.

[212] D. Pfoser and Y. Theodoridis. Generating Semantics-Based Trajectories of Moving Objects. In *Intl. Workshop on Emerging Technologies for Geo-Based Applications*, 2000.

[213] R. W. Picard and T. Kabir. Finding Similar Patterns in Large Image Databases. In *Proceedings of IEEE Conference on Acoustics, Speech, and Signal Processing*, pp. V–161–V–164, 1993.

[214] Rosalind W. Picard and Fang Liu. A New Wold Ordering for Image Similarity. In *IEEE Intl. Conference on Acoustics, Speech, and Signal Processing*, pp. 129–132, 1994.

[215] Niki Pissinou, Ivan Radev, Kia Makki, and William J. Campbell. Spatio-Temporal Composition of Video Objects: Representation and Querying in Video Database Systems. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1033–1040, 2001.

[216] J. Postel and J. Reynolds. File Transfer Protocol. IETF RFC 959. http://www.ietf.org/rfc/rfc959.txt.

[217] Sujeet Pradhan, Keishi Tajima, and Katsumi Tanaka. A Query Model to Synthesize Answer Intervals from Indexed Video Units. *IEEE Transactions on Knowledge and Data Engineering*, 13(5):824–838, 2001.

[218] William K. Pratt. *Digital Image Processing.* John Wiley & Sons, Second edition, 1991.

[219] *QuickTime File Format.* Apple Computer, Inc., 2000. http://developer.-apple.com/techpubs/quicktime/qtdevdocs/PDF/QTFileFormat.pdf.

[220] Fausto Rabitti. Document Model. ESPRIT project MULTOS. Technical Report IEI-87-01, Istituto di Elaborazione della Informazione, Pisa, Italy, 1986.

[221] Tadeusz Radecki. Mathematical Model of Information Retrieval System Based on the Concept of Fuzzy Thesaurus. *Information Processing & Management*, 12:313–318, 1976.

[222] Tadeusz Radecki. Mathematical Model of Time-Effective Information Retrieval System Based on the Theory of Fuzzy Sets. *Information Processing & Management*, 13:109–116, 1977.

[223] Tadeusz Radecki. Incorporation of Relevance Feedback into Boolean Retrieval Systems. In *Proceedings of SIGIR-83, 6th ACM Conference on Research and Development in Information Retrieval*, pp. 133–150, 1983.

[224] T. C. Rakow and M. Lohr. Audio Support for an Object-Oriented Database Management System. *Multimedia Systems*, 3(5–6):286–297, 1995.

[225] A. R. Rao and G. L. Lohse. Towards a Texture Naming System: Identifying Relevant Dimensions of Texture. In *Proceedings of IEEE Conference on Visualization*, pp. 220–227, 1993.

[226] A. L. Ratan, O. Maron, W. E. L. Grimson, and T. Lozano-Perez. A Framework for Learning Query Concepts in Image Classification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 423–431, 1999.

[227] S. Ravela and R. Manmatha. Image Retrieval by Appearance. In *Proceedings of SIGIR-97, 20th ACM Conference on Research and Development in Information Retrieval*, pp. 278–285, 1999.

[228] *RealVideo 9*. Real Networks. http://www.realnetworks.com/solutions/leadership/realvideo.html.

[229] Berthier Ribeiro-Neto and Richard Muntz. A Belief Network Model for IR. In *Proceedings of the 19th ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 253–260, 1996.

[230] S. E. Robertson and Karen Sparck Jones. Relevance Weighting of Search Terms. *Journal of the American Society for Information Science*, 27(3):129–146, 1976.

[231] Jonathan Robie. *XQL (XML Query Language)*, 1999. http://www.ibiblio.org/xql/xql-proposal.html.

[232] J. J. Rocchio. Relevance Feedback in Information Retrieval. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pp. 313–323. Prentice-Hall, 1971.

[233] R. Rosales and S. Sclaroff. 3D Trajectory for Tracking Multiple Objects and Trajectory Guided Recognition of Actions. In *IEEE Computer Vision and Pattern Recognition*, 1999.

[234] N. C. Rowe and B. Frew. Automatic Classification of Objects in Captioned Descriptive Photographs for Retrieval. In M. T. Maybury, editor, *Intelligent Multimedia Retrieval*. AAAI Press / MIT Press, 1997.

[235] Y. Rui and T. Huang. Optimizing Learning in Image Retrieval. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 236–243, 1999.

[236] Y. Rui, T. S. Huang, and S. Mehrotra. Content-Based Image Retrieval with Relevance Feedback in MARS. In *Proceedings of IEEE Conference on Image Processing*, pp. 815–818, 1997.

[237] Airi Salminen and Frank Wm. Tompa. PAT Expressions: An Algebra for Text Search. In *Proceedings of COMPLEX-92*, pp. 309–332, 1992.

[238] Airi Salminen and Frank Wm. Tompa. Requirements for XML Document Database Systems. In *Proceedings of the ACM Symposium on Document Engineering*, pp. 85–94, 2001.

[239] Gerald Salton, Ellen Voorhees, and Edward Fox. A Comparison of Two Methods for Boolean Query Relevance Feedback. *Information Processing & Management*, 20(5/6):637–651, 1984.

[240] Gerard Salton. *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, 1971.

[241] Gerard Salton and Chris Buckley. Term-Weighting Approaches in Automatic Retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[242] Gerard Salton and Chris Buckley. Improving Retrieval Performance by Relevance Feedback. *Journal of the American Society for Information Science*, 41(4):288–297, 1990.

[243] Gerard Salton, Edward A. Fox, and Harry Wu. Extended Boolean Information Retrieval. *Communications of the ACM*, 26(11):1022–1036, 1983.

[244] Gerard Salton and M. E. Lesk. Computer Evaluation of Indexing and Text Processing. *Journal of the ACM*, 15(1):8–36, 1968.

[245] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[246] A. Samal and P. A. Iyengar. Automatic Recognition and Analysis of Human Faces and Facial Expressions. *Pattern Recognition*, 25(1):65–77, 1992.

[247] H. Samet. The Quadtree and Related Hierarchical Data Structures. *ACM Computing Surveys*, 16(2):187–260, 1984.

[248] H. S. Sawhney, S. Ayer, and M. Gorkani. Model-Based 2D & 3D Dominant Motion Estimation for Mosaicking and Video Representation. Technical report, IBM Almaden Research Lab, 1994.

[249] Torsten Schlieder. ApproXQL: Design and Implementation of an Approximate Pattern Matching Language for XML. Technical Report B 01-02, Freie Universität Berlin, 2001. http://www.inf.fu-berlin.de/inst/ag-db/publications/2001/report-B-01-02.ps.

[250] Torsten Schlieder and Felix Naumann. Approximate Tree Embedding for Querying XML Data. In *ACM SIGIR Workshop On XML and Information Retrieval*, 2000.

[251] H. Schulzrinne, A. Rao, and R. Lanphier. *Real Time Streaming Protocol (RTSP)*. IETF RFC 2326. http://www.ietf.org/rfc/rfc2326.txt.

[252] S. Sclaroff and A. Pentland. A Finite-Element Framework for Correspondence and Matching. In *Proceedings of Fourth International Conference on Computer Vision*, pp. 308–313, 1993.

[253] Stan Sclaroff. Deformable Prototypes for Encoding Shape Categories in Image Databases. *Pattern Recognition*, 30(4), 1997.

[254] Fabrizio Sebastiani. A Probabilistic Terminological Logic for Modelling Information Retrieval. In *Proceedings of SIGIR-94, 17th ACM Conference on Research and Development in Information Retrieval*, pp. 122–130, 1994.

[255] Kun seok Oh, Kunihiko Kaneko, and Akifumi Makinouchi. Image Classification and Retrieval Based on Wavelet-SOM. In *Proceedings of the 1999 Intl. Symposium on Database Applications in Non-Traditional Environments*, 1999.

[256] I. K. Sethi and N. Patel. A Statistical Approach to Scene Change Detection. In *Storage and Retrieval for Image and Video Databases III*, volume SPIE 2420, pp. 329–338, 1995.

[257] J. Shanmugasundaram, K. Tufte, G. He, C. Zhang, D. J. DeWitt, and J. F. Naughton. Relational Databases for Querying XML Documents: Limitations and Opportunities. In *Proceedings of the 25th International Conference on Very Large Databases*, pp. 302–314, 1999.

[258] Linda G. Shapiro and George C. Stockman. *Computer Vision*. Prentice-Hall, 2001.

[259] B. Shararay. Scene Change Detection and Content-Based Sampling of Video Sequences. In *Digital Video Compression: Algorithms and Technologies*, volume SPIE 2419, pp. 2–13, 1995.

[260] T. Shibata and T. Kato. Modeling of Subjective Interpretation for Street Landscape Image. In E. S. Gerald Quirchmayr and T. J. M. Bench-Kapon, editors, *IEEE Workshop on Database and Expert Systems and Applications (DEXA)*. Lecture Notes in Computer Science, Springer, 1998.

[261] Thomas Sikora. The MPEG-7 Visual Standard for Content Description—An Overview. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(6):696–702, 2001.

[262] W. Siong and J. C. Lee. Efficient Scene Change Detection and Camera Motion Annotation for Video Classification. *Computer Vision and Image Understanding*, 71(2):166–181, 1998.

[263] A. Prasad Sistla, Clement Yu, and Raghu Venkatasubrahmanian. Similarity Based Retrieval of Videos. In *Proceedings of IEEE Intl. Conference on Data Engineering*, pp. 181–190, 1997.

[264] Alan F. Smeaton and Ian Quigley. Experiments on Using Semantic Distances Between Words in Image Caption Retrieval. In *Proceedings of SIGIR-96, 19th ACM Conference on Research and Development in Information Retrieval*, pp. 174–180, 1996.

[265] John R. Smith and Shih Fu Chang. VisualSEEk: a fully automated content-based image query system. In *Proceedings of MM-96, 4th ACM International Conference on Multimedia*, pp. 87–98, 1996.

[266] M. Smith and T. Kanade. Video Skimming for Quick Browsing Based on Audio and Image Characterization. Technical Report CMU-CS-95-186, Computer Science Department, Carnegie Mellon University, 1995.

[267] Karen Sparck Jones. Experiments in Relevance Weighting of Search Terms. *Information Processing & Management*, 15(13):133–144, 1979.

[268] Amanda Spink and Tefko Saračević. Human-Computer Interaction in Information Retrieval: Nature and Manifestations of Feedback. *Interacting with Computers*, 10:249–367, 1998.

[269] R. Sriram, J. M. Francos, and W. A. Pearlman. Texture Coding Using a Wold Decomposition Model. In *Proceedings of the 12th IAPR Conference on Pattern Recognition*, 1994.

[270] A. Stefanidis, P. Partsinevelos, P. Agouris, and P. Doucette. Summarizing Video Datasets in Spatiotemporal Domain. In *Proceedings of IEEE Intl. Workshop on Database and Expert Systems Applications (DEXA)*, 2000.

[271] Michael Stonebraker, H. Stettner, N. Lynn, J. Kalash, and Antonin Guttman. Document Processing in a Relational Database System. *ACM Transactions on Office Information Systems*, 1(2):143–158, 1983.

[272] Markus A. Stricker and Markus Orengo. Similarity of Color Images. In *Storage and Retrieval for Image and Video Databases*, pp. 381–392, 1995.

[273] Zhong Su, Stan Li, and Hongjiang Zhang. Extraction of Feature Subspaces for Content-Based Retrieval Using Relevance Feedback. In *Proceedings of ACM Conference on Multimedia*, pp. 98–106, 2001.

[274] Zhong Su, Hongjiang Zhang, and Shaoping Ma. Using Bayesian Classifier in Relevant Feedback of Image Retrieval. In *Proceedings of 12th IEEE Intl. Conference on Tools with Artificial Intelligence*, pp. 258–261, 2000.

[275] Dušan Surla, Zora Konjović, Branko Milosavljević, and Milan Vidaković. Bibliotečki informacioni sistem BISIS, ver. 3. In *Deveta međunarodna konferencija „Informatika u obrazovanju, kvalitet i nove informacione tehnologije“*, pp. 494–504, 2000.

[276] Michael J. Swain and Dana H. Ballard. Color Indexing. *Intl. Journal of Computer Vision*, 7(1):11–32, 1991.

[277] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Texture Features Corresponding to Visual Perception. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-8(6):460–473, 1979.

[278] Lilian H. Y. Tang, Rudolf Hanka, Horace H. S. Ip, Kent K. T. Cheung, and Ringo Lam. Integration of Intelligent Engines for a Large Scale Medical Image Database. In *IEEE 13th Symposium on Computer-Based Medical Systems (CBMS'00)*, 2000.

[279] L. Teodosio and W. Bender. Salient Video Stills: Content and Context Preserved. In *Proceedings of ACM Conference on Multimedia*, 1993.

[280] A. Theobald and G. Weikum. Adding Relevance to XML. In *Proceedings of 3rd International Workshop on the Web and Databases (WebDB'00)*, 2000.

[281] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn. *XML Schema Part 1: Structures*. W3C Recommendation, 2001. http://www.w3.org/TR/-xmlschema-1.

[282] Simon Tong and Edward Chang. Support Vector Machine Active Learning for Image Retrieval. In *Proceedings of ACM Conference on Multimedia*, pp. 107–118, 2001.

[283] Douglas Tudhope and Daniel Cunliffe. Semantically Indexed Hypermedia: Linking Information Disciplines. *ACM Computing Surveys*, 31(4es):1–6, 1999.

[284] Howard Turtle and W. Bruce Croft. Inference Networks for Document Retrieval. In *Proceedings of the 13th ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1–24, 1990.

[285] Howard Turtle and W. Bruce Croft. Evaluation of an Inference Network-Based Retrieval Model. *ACM Transactions on Information Systems*, 9(3):187–222, 1991.

[286] Shunsuke Uemura, Masatoshi Yoshikawa, and Toshiyuki Amagasa. Heijo – A Database System for Retrieving Semantically Coherent Video Information. In *Proceedings of IEEE Intl. Symposium on Database Applications in Non-Traditional Environments*, 1999.

[287] *UNIMARC Manual: Bibliographic Format/International Federation of Library Associations and Institutions*. IFLA Universal Bibliographic Control and International MARC Programme, New Providence, London, 1994.

[288] Aditya Vailaya, Yu Zhong, and Anil K. Jain. A Hierarchical System for Efficient Image Retrieval. In *Proceedings of Intl. Conference on Pattern Recognition*, 1996.

[289] Cornelis J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[290] Iraklis Varlamis and Michalis Vazirgiannis. Bridging XML-schema and Relational Databases: A System for Generating and Manipulating Relational Databases Using Valid XML Documents. In *Proceedings of the ACM Symposium on Document Engineering*, pp. 105–114, 2001.

[291] Nuno Vasconcelos and Andrew Lippman. A Spatiotemporal Motion Model for Video Summarization. In *IEEE 13th IEEE Symposium on Computer-Based Medical Systems (CBMS'00)*, pp. 361–366, 1998.

[292] V. Vianu. A Web Odyssey: From Codd to XML. In *Proceedings of 20th Symposium on Principles of Database Systems*, pp. 1–15, 2001.

[293] Ellen M. Voorhees. The Cluster Hypothesis Revisited. In *Proceedings of SIGIR-85, 8th ACM Conference on Research and Development in Information Retrieval*, pp. 188–196, 1985.

[294] Ellen M. Voorhees, Narenda K. Gupta, and Ben Johnson-Laird. The Collection Fusion Problem. In *Proceedings of the Third Text Retrieval Conference (TREC-3)*, pp. 95–104, 1995.

[295] Ellen M. Vorhees and Donna K. Harman. Overview of the 6th Text Retrieval Conference (TREC-6). In *Proceedings of the 6th Text REtrieval Conference*, pp. 1–24, 1997.

[296] Howard D. Wactlar, Takeo Kanade, Michael A. Smith, and Scott M. Stevens. Intelligent Access to Digital Video: Informedia Project. *IEEE Computer*, 29(5):46–52, 1996.

[297] Tim T. Y. Wai and Arbee L. P. Chen. Retrieving Video Data via Motion Tracks of Content Symbols. In *Proceedings of ACM 6th Intl. Conference on Information and Knowledge Management (CIKM'97)*, pp. 105–112, 1997.

[298] S. Wartick. Boolean Operations. In William B. Frakes and Ricardo Baeza-Yates, editors, *Information Retrieval: Data Structures & Algorithms*. Prentice-Hall, 1992.

[299] Ji-Rong Wen and Hong-Jiang Zhang. Query Clustering in the Web Context. In W. Wu, H. Xiong, and S. Shekhar, editors, *Information Retrieval and Clustering*. Kluwer Academic Press, 2002.

[300] R. Wilkinson and P. Hingston. Using the Cosine Measure in a Neural Network for Document Retrieval. In *Proceedings of the 14th ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 202–210, 1991.

[301] Darrel Woelk, Won Kim, and Willis Luther. An Object-Oriented Approach to Multimedia Databases. In *Proceedings of ACM SIGMOD Intl. Conference on Management of Data*, pp. 312–325, 1986.

[302] Raymond K. Wong. The Extended XQL for Querying and Updating Large XML Databases. In *Proceedings of the ACM Symposium on Document Engineering*, pp. 95–104, 2001.

[303] S. K. M. Wong, W. Ziarko, and P. C. N. Wong. Generalized Vector Space Model in Information Retrieval. In *Proceedings of 8th ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 18–25, 1985.

[304] M. E. J. Wood, N. W. Campbell, and B. T. Thomas. Iterative Refinement by Relevance Feedback in Content-Based Digital Image Retrieval. In *Proceedings of MM-98, 6th ACM International Conference on Multimedia*, pp. 13–20, 1998.

[305] H. Wu and Gerard Salton. The Estimation of Term Relevance Weights Using Relevance Feedback. *Journal of Documentation*, 37(4):194–214, 1981.

[306] Jian-Kang Wu. Content-Based Indexing of Multimedia Databases. *IEEE Transactions on Knowledge and Data Engineering*, 9(6):978–989, 1997.

[307] P. Wu and B. S. Manjunath. Adaptive Nearest Neighbor Search for Relevance Feedback in Large Image Databases. In *Proceedings of ACM Conference on Multimedia*, pp. 89–97, 2001.

[308] P. Wu, B. S. Manjunath, S. D. Newsam, and H. D. Shin. A Texture Descriptor for Image Retrieval and Browsing. In *IEEE Workshop on Content-Based Access of Image and Video Libraries*, pp. 3–7, 1999.

[309] *Apache Xindice*. The Apache Software Foundation. http://xml.apache.org/-xindice.

[310] Jinxi Xu and W. Bruce Croft. Query Expansion Using Local and Global Document Analysis. In *Proceedings of SIGIR-96, 19th ACM Conference on Research and Development in Information Retrieval*, pp. 4–11, 1996.

[311] Yasser Yacoob and Larry Davis. Recognizing Facial Expression by Spatio-Temporal Analysis. In *Intl. Conference on Pattern Recognition (ICPR)*, pp. 747–749, 1994.

[312] Boon-Lock Yeo and Minerva M. Yeung. Retrieving and Visualizing Video. *Communications of the ACM*, 40(12):43–52, 1997.

[313] M. Yeung and Boon-Lock Yeo. Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(5):771–785, 1997.

[314] M. M. Yeung and B. L. Yeo. Time-Constrained Clustering for Segmentation of Video into Story Units. In *Intl. Conference on Pattern Recognition (ICPR)*, volume C, pp. 375–380, 1996.

[315] M. M. Yeung, B. L. Yeo, W. Wolf, and B. Liu. Video Browsing Using Clustering and Scene Transitions on Compressed Structures. In *Multimedia Computing and Networking*, volume SPIE 2417, pp. 399–413, 1995.

[316] Masatoshi Yoshikawa, Toshiyuki Amagasa, Takeyuki Shimura, and Shunsuke Uemura. XRel: A Path-Based Approach to Storage and Retrieval of XML Documents Using Relational Databases. *ACM Transactions on Internet Technology*, 1(1):110–141, 2001.

[317] Atsuo Yoshitaka and Tadao Ichikawa. A Survey on Content-Based Retrieval for Multimedia Databases. *IEEE Transactions on Knowledge and Data Engineering*, 11(1):81–93, 1999.

[318] Atsuo Yoshitaka, Masanori Yoshimitsu, Masahito Hirakawa, and Tadao Ichikawa. V-QBE: Video Database Retrieval by Means of Example Motion of Objects. In *Proceedings of IEEE Intl. Conference on Multimedia Computing and Systems (ICMCS)*, pp. 453–457, 1996.

[319] J. You, H. Shen, and H. A. Cohen. An Efficient Parallel Texture Classification for Image Retrieval. In *Proceedings of the 1997 IEEE Advances in Parallel and Distributed Computing*, pp. 18–25, 1997.

[320] D. Yow, B. L. Yeo, M. Yeung, and B. Liu. Analysis and Presentation of Soccer Highlights from Digital Video. In *Proceedings of Second Asian Conference on Computer Vision*, 1995.

[321] Hong-Heather Yu and Wayne Wolf. A Visual Search System for Video and Image Databases. In *Proceedings of IEEE Intl. Conference on Multimedia Computing and Systems*, pp. 517–524, 1997.

[322] H. J. Zhang, Y. H. Gong, S. W. Smoliar, and S. Y. Yan. Automatic Parsing of News Video. In *Proceedings of Intl. Conference on Multimedia Computing and Systems*, pp. 45–54, 1994.

[323] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic Partitioning of Full-Motion Video. *Multimedia Systems*, 1:10–28, 1993.

[324] H. J. Zhang, C. Y. Low, S. W. Smoliar, and J. H Wu. Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution. In *Proceedings of ACM Conference on Multimedia*, 1995.

[325] J. Zhang. Application of OODB and SGML Techniques in Text Database: An Electronic Dictionary System. *ACM SIGMOD Record*, 24(1):3–8, 1995.

[326] D. Zhong, H. J. Zhang, and S. F. Chang. Clustering Methods for Video Browsing and Annotation. In *SPIE Conference on Storage and Retrieval for Image and Video Databases*, 1996.

[327] Xiang Sean Zhou and Thomas S. Huang. Comparing Discriminating Transformations and SVM for Learning during Multimedia Retrieval. In *Proceedings of ACM Conference on Multimedia*, pp. 137–146, 2001.

[328] Lei Zhu, Aibing Rao, and Aidong Zhang. Theory of keyblock-based image retrieval. *ACM Transactions on Information Systems*, 20(2):224–257, 2002.

Biografija

Branko Milosavljević je rođen u Novom Sadu 1973. godine. Diplomirao je na Fakultetu tehničkih nauka Univerziteta u Novom Sadu 1997. godine na odseku Elektrotehnika i računarstvo, usmerenju Računarstvo. Diplomski rad sa temom „Implementacija tekst servera za indeksiranje i pretraživanje bibliografske građe u Oracle okruženju“ odbranio je sa ocenom 10 (deset).

Odmah po diplomiranju započeo je rad na Fakultetu tehničkih nauka kao stipendista Ministarstva za nauku i tehnologiju Republike Srbije. Radni odnos je zasnovao 1998. godine u Institutu za računarstvo i automatiku Fakulteta tehničkih nauka. U periodu od septembra 2000. do septembra 2001. bio je na odluženju vojnog roka.

Poslediplomske studije na Fakultetu tehničkih nauka, smer Računarstvo, završio je 1999. godine. Odbranio je magistarski rad pod nazivom „Tekst server UNIMARC zapisa“. Dobitnik je nagrade „Mileva Marić-Ajnštajn“ za najbolji magistarski rad odbranjen u toku 1999. godine iz oblasti računarstva.

U toku studija i rada na Fakultetu objavio je 30 radova u monografijama, domaćim i stranim časopisima i zbornicima sa konferencija, i koautor je dva softverska projekta, sa ukupnim koeficijentom kompetencije $R = 35,1$. Pored toga, održao je jedno predavanje po pozivu, objavio jednu knjigu i učestvovao u izradi nekoliko projekata.

Oženjen je i živi u Novom Sadu. Od stranih jezika govori engleski jezik.

Univerzitet u Novom Sadu
Fakultet tehničkih nauka

Ključna dokumentacijska informacija

Redni broj: RBR	
Identifikacioni broj: IBR	
Tip dokumentacije: TD	monografska dokumentacija
Tip zapisa: TZ	tekstualni štampani dokument
Vrsta rada: VR	doktorska disertacija
Autor: AU	Branko Milosavljević
Mentor: MN	prof. dr Zora Konjović, vanr. prof.
Naslov rada: NR	Proširivi sistem za pronalaženje multimedijalnih dokumenata
Jezik publikacije: JP	srpski (latinica)
Jezik izvoda: JI	srpski (latinica) / engleski
Zemlja publikovanja: ZP	Srbija i Crna Gora

Uže geografsko područje: Vojvodina
UGP

Godina: 2003
GO

Izdavač: autorski reprint
IZ

Mesto i adresa: Fakultet tehničkih nauka,
MA Trg. D. Obradovića 6, Novi Sad

Fizički opis rada: 4/189/328/14/36/0/0
FO

Naučna oblast: računarske nauke
NO

Uža naučna oblast: pronalaženje informacija
ND

Ključne reči: multimedijalni dokumenti, pronalaženje informacija,
PO XML dokumenti

UDK:

Čuva se: Biblioteka Fakulteta tehničkih nauka,
ČU Trg D. Obradovića 6, Novi Sad

Važna napomena:
VN

Datum prihvatanja od
strane NN veća:
DP

Datum odbrane:
DO

Izvod:
IZ
Oblast pronalaženja informacija kao jedan od osnovnih problema razmatra pronalaženje dokumenata u kolekciji koji su relevantni sa stanovišta korisnika. Ova disertacija se bavi problemima pronalaženja strukturiranih multimedijalnih dokumenata. Strukturirani multimedijalni dokumenti mogu, kao svoje elemente, sadržati objekte različitih tipova medija (tekst, slika, zvuk, ili video). Tema disertacije je formalna specifikacija modela sistema koji omogućava pronalaženje multimedijalnih dokumenata obezbeđujući pri tom proširivost sistema podrškom za različite tipove medija (što uključuje upotrebu različitih postojećih rešenja iz ove oblasti) i proširivost sistema različitim modelima pronalaženja dokumenata. XML jezik se koristi kao jezik za reprezentaciju dokumenata i kao jezik za komunikaciju sistema sa klijentima. Sistem je verifikovan na realnom primeru digitalne biblioteke doktorskih i magistarskih teza pomoću razvijenog prototipa. Prikazana prototipska implementacija koja ispunjava ciljeve u pogledu funkcionalnosti postavljene pred sistem predstavlja potvrdu praktične vrednosti predloženog modela.

Komisija:
KO

predsednik: prof. dr Dušan Surla, red. prof., PMF Novi Sad
član: prof. dr Dušan Starčević, red. prof., FON Beograd
član: prof. dr Zora Konjović, vanr. prof., FTN Novi Sad, mentor
član: prof. dr Branko Perišić, vanr. prof., FTN Novi Sad
član: dr Miro Govedarica, doc., FTN Novi Sad

University of Novi Sad
Faculty of Engineering

Keyword Documentation

Accession number:
ANO

Identification number:
INO

Document type: DT	monograph publication
type of record: TR	textual printed material
Contents code: CC	doctoral dissertation
Author: AU	Branko Milosavljević
Menthor: MN	Zora Konjović, Ph.D,. assoc. prof.
Title: TI	Extensible Multimedia Information Retrieval System
Lanugage of text: LT	Serbian (latin)
Lanugage of abstract: LA	Serbian (latin) / English
Country of publication: CP	Serbia and Montenegro

Locality of publication: Vojvodina
LP

Publication year: 2003
PY

Publisher: author's reprint
PU

Publishing place: Faculty of Engineering,
PP Trg. D. Obradovića 6, Novi Sad

Physical description: 4/189/328/14/36/0/0
PD

Scientific field: computer science
SF

Scientific discipline: information retrieval
SD

Subject/keywords: multimedia documents, information retrieval,
SKW XML documents

UDC:

Holding data: Library of Faculty of Engineering,
HD Trg D. Obradovića 6, Novi Sad

Note:
N

Accepted by scientific
board on:
ASB

Defended:
DE

Abstract: The field of information retrieval deals with
AB retrieval of documents judged as relevant by users. This dissertation focuses on problems in retrieval of structured multimedia documents. Structured multimedia documents comprise objects of different media types (such as text, images, audio or video clips) as their elements. The subject of the dissertation is a formal specification of a multimedia information retrieval system providing extensibility with support for different media types (including utilizing existing solutions in this field) and extensibility with different document retrieval models. XML is used as a language for expressing document content and as a langugage for communication between the system and its clients. The system is verified by a case study on a networked digital library of theses and dissertations. The presented prototype implementation epresents a proof of the proposed model's practical value.

Thesis defend board:
DB

president: prof. Dušan Surla, PhD, Faculty of Science Novi Sad
member: prof. Dušan Starčević, PhD, Faculty of Organizational Sciences Belgrade
member: prof. Zora Konjović, PhD, Faculty of Engineering Novi Sad, menthor
member: prof. Branko Perišić, PhD, Faculty of Engineering Novi Sad
member: prof. Miro Govedarica, PhD, Faculty of Engineering Novi Sad