



Univerzitet u Novom Sadu
Fakultet Tehničkih Nauka u
Novom Sadu



Valentin Penca

Modeliranje i implementacija sistema za pretragu naučno-istraživačkih rezultata

- doktorska disertacija -

Нови Сад, 2014

Predgovor

Predmet istraživanja je sistem za pretragu naučno-istraživačkih rezultata. Sistem za pretragu naučno-istraživačkih rezultata je nastao kao proširenje postojećeg sistema za unos ovih podataka (CRIS UNS). Analizom postojećih sistema za naučno-istraživačkih sistema došlo se do zaključka da je korišćeno vrlo malo opšte prihvaćenih standarda za predstavljanje i pretragu podataka iz naučno-istraživačkog domena. Standardizacija u ovakvim sistemima je od ključnog značaja jer treba da obezbedi nezavisnost pretraživanja od samih modela za reprezentaciju podataka naučno-istraživačke delatnosti. Stoga je i rezultat istraživanja sistem pretrage naučno-istraživačkih rezultata kreiran na postojećim standardima (Z39.50, SRU/W ...) i tehnologijama (Web 2.0, AJAX, Lucene ...) aktuelnim u oblasti pretrage. Testiranje i verifikacija sistema urađena je na naučno-istraživačkim podacima CRIS UNS sistema.

Disertacija sadrži sledeća poglavlja:

1. Uvod
2. Analiza sistema pretraga
3. Udaljeno pretraživanje podataka CRIS sistema
4. Organizacija podataka unutar CRIS UNS sistema
5. Pretraživanje podataka unutar CRIS UNS sistema
6. Implementacija SRU/W standarda u CRIS UNS sistemu

U **prvom**, odnosno uvodnom poglavlju dat je pregled postojeće literature u vezi sa pretragom naučno-istraživačkih podataka. Poseban akcenat je na podržanim funkcionalnostima i korišćenim standardima u domenu pretrage naučnih rezultata. Takođe, u ovom poglavlju su istaknute i tehnologije koje se koriste u sistemima za pretragu (Web 2.0, AJAX, Lucene ...). Na kraju poglavlja dati su ciljevi disertacije, kao i postavljene hipoteze za izradu disertacije.

Pregled postojećih sistema koji omogućuju nekakav vid pretrage naučno-istraživačkih podataka dat je u **drugom** poglavlju. Pored opisa osnovnih karakteristika samih sistema sprovedena je i analiza ovih sistema u odnosu na: dostupne indekse, modove pretrage, tipove upita.

Treće poglavlje sadrži detaljan opis CRIS profila pretrage. Pomenuti profil je u skladu sa SRU/W standardom. CRIS profil je nastao analizom postojećih SRU/W profila i na osnovu zaključaka iz analize sistema pretraga naučno-istraživačkih podataka. Ovo poglavlje sadrži i detaljnu specifikaciju SRU/W standarda i upitnog jezika CQL.

Četvrto poglavlje sadrži opis podataka iz postojećeg CRIS UNS sistema. Opisana je i komponenta za preliminarnu obradu, indeksiranje i pretragu tekstualnih sadržaja. Prikazano je i mapiranje uskladištenih podataka sistema na adekvatne indekse pretrage. Implementirani indeksi sistema proizašli su iz analize navedene u prethodnom poglavlju.

U **petom** poglavlju je opisana specifikacija, arhitektura i implementacija sistema za pretraživanje naučno-istraživačkih podataka. Prvo je data specifikacija informacionih zahteva sistema upotrebom dijagrama slučajeva korišćenja. Nakon toga je opisana arhitektura sistema upotrebom dijagrama razmeštaja. Na kraju ovog poglavlja je opisana implementacija sistema za pretraživanje naučno-istraživačkih podataka unutar postojećeg informacionog sistema naučno-istraživačke delatnosti CRIS UNS.

U **šestom** poglavlju je data specifikacija i implementacija serverske strane SRU/W protokola. Na kraju je izneta i diskusija o mogućim scenarijima korišćenja udaljenog pretraživanja.

Zaključci i mogući pravci daljih istraživanja su navedeni u **sedmom** poglavlju

U disertaciji je usvojen i korišćen Harvard stil citiranja referenci.

Zahvaljujem se svom mentoru dr **Draganu Ivanoviću** na sugestijama, korisnim savetima, tokom mojih doktorskih studija i izrade doktorske disertacije.

Zahvaljujem se i dr **Dušanu Surli** i dr **Zori Konjović** na brojnim sugestijama, korisnim idejama, a sve u cilju podizanja kvaliteta ove disertacije.

Zahvaljujem i svojoj porodici, rodbini i prijateljima koji su mi bili ogromna podrška tokom čitavog školovanja. Posebnu zahvalnost dugujem svojoj životnoj saputnici **Maji** na bezrezervnoj podršci, ljubavi i strpljenju.

Sadržaj

1. Uvod.....	1
1.1. Aktuelno stanje u oblasti	2
1.2. Predmet i cilj istraživanja.....	10
2. Analiza sistema pretraga.....	12
2.1. Analiza indeksa	12
2.2. Analiza modova pretrage	19
2.3. Funkcionalnosti pretrage.....	23
3. Udaljeno pretraživanje podataka CRIS sistema.....	29
3.1. SRU/W standard.....	29
3.1.1. SearchRetrive servis.....	32
3.1.2. Expain Servis	47
3.1.3. Scan servis	52
3.1.4. Obrada grešaka.....	54
3.2. CQL.....	55
3.2.1. Sintaksa CQL upita.....	56
3.3. Context Set.....	61
3.3.1. CQL Context Set.....	62
3.3.2. Dublin Core Context Set.....	69
3.3.3. Bib Context Set.....	71
3.3.4. Bath Context Set	77
3.4. SRU/W profili pretrage	78
3.4.1. SRU/W Base Profile	78
3.4.2. CQL Profile for Bibliographic Searching (Bib Profile).....	79
3.4.3. Bath Profile	80
3.4.4. CRIS Profil pretrage	81

4.	Organizacija podataka unutar CRIS UNS sistema.....	88
4.1.	CRIS UNS model podataka.....	89
4.1.1.	CERIF	89
4.1.2.	MARC 21	90
4.1.3.	CERIF kompatibilni model podataka baziran na MARC 21 formatu 91	
4.2.	Indeksiranje podataka.....	93
4.2.1.	Lucene.....	94
4.2.1.1.	Lucene Arhitektura	95
4.2.1.2.	Lucene indeksiranje.....	97
4.2.1.3.	Invertovani indeks	99
4.2.1.4.	Lucene API.....	100
4.2.1.5.	Indeksna Analiza	102
4.2.1.6.	Ugrađeni analizatori	103
4.2.1.7.	Sopstveni Analizator.....	105
4.3.	Proces indeksiranja u sistemu pretrage CRIS UNS	106
5.	Pretraživanje podataka unutar CRIS sistema.....	112
5.1.	Specifikacija informacionih zahteva	112
5.1.1.	Pretraga publikacija	113
5.1.2.	Pretraga autora	116
5.1.3.	Pretraga tekstualnim zadavanjem upita	118
5.1.4.	Procesiranje CQL upita i vraćanje rezultata	119
5.2.	Implementacija sistema za pretragu podataka naučno-istraživačke delatnosti.....	121
5.2.1.	Statički model sistema	121
5.2.2.	Dijagram komponenti	124
5.2.3.	Dijagram klasa	125

5.3. Opis korišćenja aplikacije	128
5.4. Implementacija podsistema za Zadužbinu Andrejević.....	135
6. Implementacija SRU/W standarda u CRIS UNS sistemu.....	139
6.1. JAX-WS	143
6.2. Apache CXF.....	147
6.3. Korišćenje udaljenog pretraživanja	157
7. Zaključak.....	160
Prilog 1- Analizirani sistemi pretraga	164
Prilog 2 - SRU/W tipovi podata.....	168
Prilog 3 - WSDL specifikacija SRU/W servisa	174
Literatura.....	176
Biografija	189

1. Uvod

Naučni i industrijski razvoj društva zasnovanog na znanju usko zavise od mogućnosti korišćenja i razmene podataka. Preduslov naučnom razvoju društva je sistematizacija, kategorizacija i dostupnost prisutnog znanja. Napredak društva u određenoj naučnoj oblasti iziskuje upoznavanje istraživača sa: tematikom, ključnim ljudima (vodećim istraživačima) i njihovim naučnim rezultatima. Za uspešan naučno-istraživački razvoj od posebne važnosti je uvid ostvarenih naučno-istraživačkih rezultata. Značaj dostupnosti naučnih rezultata za dalji razvoj nauke diskutovan je u radovima (Lawrence, 2001; Harnard & Brody, 2004; Antelman, 2004; Anderson et al., 2001; Kurtz et al., 2005a; Kurtz et al., 2005a; Kurtz et al., 2005b; Eysenbach, 2006).

Postoji mnoštvo dostupnih sistema za upravljanje i distribuciju naučno-istraživačkih podataka kao što su: digitalne biblioteke, institucioni repozitorijumi (IR), istraživački informacioni sistemi (*Current Research Information System* - CRIS), itd. Navedeni sistemi najčešće podržavaju samo određeni segment podataka iz naučno-istraživačke oblasti. Pomenuti sistemi poseduju ograničenja koja se odnose na njihovu primenu, a koja su nastala kao posledica specijalizacije samih pri radu sa određenim tipovima podataka naučno-istraživačke delatnosti (npr. digitalna biblioteka elektronskih časopisa, institucioni repozitorijum fakulteta, ...). Tako je eksploatacija i uvođenje ovakvih sistema otežana usled nedostatka opšte prihvaćenih standarda za predstavljanje podataka naučno-istraživačkog domena. Standardizacija u ovakvim sistemima je od ključnog značaja. Sistem u kome se nalaze podaci o: publikacijama, događajima, produktima istraživanja, istraživačima i naučnim institucijama morao bi da se temelji na opšte prihvaćenim standardima, gde bi adekvatan primer takvog sistema bio CRIS (*Current Research Information System*, 2000) baziran na CERIF (*Common European Research Information Format* | CERIF, 2000) standardu. Pomenuti standard i njegova primena detaljno je opisana u radovima (Asserson et al., 2002; Jörg, 2008; Jeffery, 2010; Jeffery and Asserson, 2010; Jeffery et al., 2014; Gartner et al., 2013; Bailo and Jeffery, 2014; Guskov et al., 2014; Pinto et al., 2014). Standard CERIF opisuje fizički model podataka (Jörg et al., 2012) i razmenu XML poruka između CRIS sistema (Dvořák and Jörg, 2013). CRIS može da sadrži relevantne

naučno-istraživačke podatke: digitalnih biblioteka, IR i drugih izvora informacija iz naučno-istraživačke oblasti. Evidentno je da CRIS sistemi sadrže veliku količinu podataka, pa je stoga neophodno obezbediti efikasan način pretrage istih, koja se mora zasnivati na nekom standardu. Uvođenje standarda kod pretrage podataka treba da obezbedi nezavisnost pretraživanja od samih modela za reprezentaciju podataka naučno-istraživačke delatnosti. Standardizacija sistema podrazumeva upotrebu postojećih standarda Z39.50 (National Information Standards Organization (U.S.), 2003, p. 50) SRU/W (SRU- Search/Retrieve via URL, 2004) i tehnologija ("What Is Web 2.0 - O'Reilly Media," 2005), AJAX (Asynchronous JavaScript and XML, 2007), Lucene (Apache Lucene, 2005) koji se intenzivno koriste u domenu pretrage kod IR, digitalnih biblioteka, CRIS sistema.

Rezultat ove doktorske disertacije je sistem pretrage naučno-istraživačkih rezultata zasnovanog na postojećim standardima (Z39.50, SRU/W ...) i tehnologijama (Web 2.0, AJAX, Lucene ...) koji se intenzivno koriste u domenu pretrage kod IR, digitalnih biblioteka, CRIS sistema.

1.1. Aktuelno stanje u oblasti

U ovoj sekciji je izvršen pregled postojeće literature. Korišćeni su naučni radovi iz oblasti razvoja CRIS sistema i modela podataka CERIF, kao i niz radova iz domena konstruisanja korisničkog interfejsa . Poseban akcenat će biti na podržanim funkcionalnostima i korišćenim standardima u domenu pretrage naučnih rezultata. Takođe, u ovom poglavlju će biti istaknute i tehnologije koje se koriste u sistemima za pretragu (Web 2.0, AJAX, Lucene ...). Pored toga, korišćeni su i svi publikovani radovi na razvoju softverskog sistema CRIS UNS (<http://cris.uns.ac.rs>), koji se razvija na Univerzitetu u Novom Sadu.

U sadašnje vreme proces pronalaženja i pribavljanja naučnog znanja neretko se svodi na korišćenje široko rasprostranjenih pretraživača opšte namene. Postoji rašireno mišljenje kako je za pretragu podataka iz naučno-istraživačke delatnosti uglavnom dovoljan neki od opštih sistema kao što je Google ili Yahoo. Deo korisnika koristi i specijalizovane podsisteme za pretragu naučno-istraživačkih podataka prethodno navedenih opštih sistema kao što su GoogleScholar (<http://scholar.google.com/>) ili Microsoft Academic (<http://academic.research.microsoft.com/>). Stoga se postojanje

sistema pretraga za repozitorijume „lokalnog“ karaktera često stavlja u drugi plan. Postavlja se pitanje koliko jedan opšti pretraživač zaista može doprineti pronalazenju specifičnog naučnog znanja (npr. publikacije iz domena upotrebe telemedicine u Srbiji) i da li bi u tom slučaju primerenije bilo koristiti specijalizovane sisteme pretraga. Odgovor na ovo pitanje može se naslutiti iz činjenice da su pretraživači opšte namene uglavnom namenjeni pretrazi veb resursa i da postoji veliki broj informacija koje se ne nalaze indeksirane u ovakvim sistemima (npr. sadržaji baza podataka). U (Lossau, 2004) se ističe da postoji niz informacija koje se nalaze u tkz. „*deep web*“, koje nisu dostupne putem GoogleScholar ili Microsoft Academic. Tako se u (Brophy and Bawden, 2005) jasno ističe da je dostupnost i brzina na strani Google dok je preciznost na strani digitalnih biblioteka.

Na veb sajtu (“Stanford E-Journal User Survey,” 2004) može se pronaći Stanford upitnik, deo projekta Stanford E-Journal User Study, kreiran za evaluaciju elektronskih časopisa i njihovih baza podataka. Takođe, na veb sajtu su dostupni statistički rezultati koji su oformljeni analizom odgovora od preko 10.000 upitanika. U domenu pretrage, ispitaniku su postavljena pitanja sa ciljem da se odredi konkretni sistem pretrage (pretraživač) koji je korišćen za pronalazenje časopisa, kao i skup relevantnih indeksa pogodnih za opis samih rezultata pretrage. Uzimajući u obzir da su relevantni indeksi opisa rada takođe bitni i za njegovu pretragu, statistika upitnika mogla je biti iskorišćena za identifikaciju indeksa pretrage koji su najviše u upotrebi. U upitniku su izdvojeni sledeći indeksi: naslov, autor, ime časopisa, datum izdavanja, broj strana, ISSN, broj citata i apstrakt.

Rad (Biswas and Paul, 2010) opisuje digitalne repozitorijume otvorenog koda, kao i njihovu primenu kod realizacije IR-a sa naglaskom na DSpace (*DSpace* - <http://www.dspace.org/>) i Greenstone (*Greenstone* - <http://www.greenstone.org/>). U radu su opisane mogućnosti pretrage DSpace (pretraga fraza, upotreba džoker znakova...).

Rad (Crudge and Johnson, 2004) opisuje evaluaciju pretraživača iz perspektive korisnika. U ovom radu su data korisnikova zapažanja o testiranim pretraživačima, koja je Crudge iskoristio za formiranje tkz. „koncepta pretrage“. Pomenutim konceptima definisan je korisnikov mentalni model pretrage, koji su autori rada kasnije iskoristili za kreiranje modela evaluacije pretraživača (user evaluation criteria and measures).

Nakon izvršene evaluacije, autori kroz rezultate ukazuju da je korisnicima bitno postojanje sledećih funkcionalnosti pretrage: predloga vrednosti terma pretrage (kontekstna pomoć, automatsko kompletiranje pretrage), postojanje standardne (jednostavne) i napredne pretrage, pomoć u pretrazi, rezultati za slično napisane reči (približna pretraga), logička pretraga (zadavanje logičkih upita) i standardizovana funkcionalnost interfejsa („Easy to use, even though have not used it before“).

U (Mehrad and Gilvari, 2006) izvršena je analiza repozitorijuma elektronskih časopisa sa akcentom na pretragu istih. Za svaki od analiziranih časopisa autori su posmatrali postojanje standarda, korišćenje indeksa i mogućnosti same pretrage. Iz rezultata analize došlo se do relevantnih predloga i zaključka neophodnih za sisteme pretrage publikacija. Jedan od rezultata analize bio je predlog za uvođenje standarda pretrage kao što su: ISO 8777, ISO 10162/10163 i NISO Z39. Takođe, zaključeno je da pretraga treba da podrži sve relevantne indekse (subject, author, year, title...) koji bliže opisuju elektronske časopise. Mogućnost konstruisanja složenih upita **logičkim operatorima**, korišćenje **džoker karaktera** kod pretrage nedovršenih reči/rečenica i pretraga na osnovu **udaljenosti između reči**, izdvojene su kao vrlo popularne funkcionalnosti kod korisnika.

Osnovni cilj u radu (Madhusudhan and Aggarwal, 2011) bio je da se ispituju različite funkcije i komponente veb interfejsa *online public access catalogues* (OPACs) kod Indian Institutes of Technology (IIT) biblioteka. Između ostalog, rad se bavi funkcijom pretrage u IIT. Tabelarno su prikazani indeksi i mogućnosti pretrage različitih biblioteka. Upotrebljeni kriterijumi za evaluaciju IIT biblioteka u ovom radu su: **modovi** pretrage (jednostavna, napredna), upotreba logičkih operatora, definisanje fraza, unos nedovršenih reči i pretrage po udaljenosti. Predloženi kriterijumi pretrage direktno određuju poželjne funkcionalnosti (mogućnosti) koje bi svaki savremeni sistem morao da podrži.

Razvoj sistema pretrage kako se navodi u (Wilson, 2011) može se posmatrati kroz razvoj *Search User Interfejs* (SUI) i *backend* sistema koji izvršava pretragu. Wilson (Wilson, 2011) opisuje SUI kao „gateway“ između korisnika, koji su pokretači pretrage i digitalnih skladišta informacija i podataka (gateway between people who have a task to

complete, and the repositories of information and data stored around the world), koji tu pretragu sprovode.

Zbog činjenice da je pretraga namenjena ljudima, za izgradnju kvalitetnog sistema pretrage koji je prilagođen njihovim potrebama, neophodno je proučiti kako korisnici izvršavaju proces pretrage. Još u eri razvoja baza podataka Bates (Bates, 1979) je na osnovu proučavanja ponašanja korisnika razvio koncept taktika za pretragu informacija (information search tactics) koje je moguće primeniti na *online* bazama podataka. Pomenute taktike imale su veliki uticaj na razvoj *online* korisničkih interfejsa i ponašanja korisnika pri pretrazi. Rad (Smith, 2012) uzima kao osnovu Bates taktike i proširuje ih i prilagođava tako da odgovaraju pretragama na internetu. Internet taktike pretrage kao što su PARALLEL, EXHAUST, REDUCE i TELEPORT omogućuju sagledavanje modela ponašanja korisnika pri pretrazi informacija (information-seeking behaviour). Neophodno je navesti i rad (Mansourian, 2008) koji objašnjava ponašanje korisnika usled neuspešne pretrage (information seeking failure) i predlaže strategije uz pomoć kojih korisnik može da prevaziđe neuspeh u pretrazi. Iako veći deo strategija utiče samo na akcije korisnika, neke od strategija utiču i na dizajn SUI. Interfejs bi trebao da se prilagodi tako da podrži određene taktike (npr. za aktivnu Revising strategiju treba dizajnirati interfejs tako da omogući korisniku redifinisanje upita i sužavanje domena pretrage, ne bi li rezultat pretrage postao zadovoljavajući).

Spink u svom radu (Spink, 2003) opisuje izgled upita pretrage kod veb pretraživača i daje objašnjenja kako ljudi pretražuju veb. Tendencija porasta broja reči u upitu, upotreba *boolean* operatora i korišćenje naprednih funkcionalnosti pretrage sugeriše da se sistemi pretrage moraju dizajnirati tako da podrže korisnike sa **različitim nivoima iskustva** (od početnika do iskusnih korisnika). Idealno bi bilo da se korisniku inicijalno prikaže jednostavan ali ipak dovoljno moćan interfejs koji je lak za korišćenje i koji sadrži svega nekoliko jednostavnih komponenti sa kojima bi se kreirali prosti upiti. Takođe treba ostaviti opcije da se inicijalni interfejs lako može proširiti uključivanjem dodatnih komponentata, sa ciljem da se podrži kreiranje kompleksnih upita koji sadrže napredne funkcionalnosti pretrage. Sličan predlog dizajniranja korisničkog interfejsa dat je u (Ahmed et al., 2009), gde projektovanje sistema treba da bude okrenuto ka korisniku (*user centric*). Predlaže se dizajniranje jednostavnog interfejsa namenjenog

korisnicima početnicima (lak za korišćenje, intuitivan, jednostavni upiti) i naprednog interfejsa namenjenog iskusnim korisnicima (kompleksni upiti).

Dizajn SUI opisan je u (Wilson, 2011). Wilson u knjizi daje pregled savremenih korisničkih interfejsa pretrage, opise komponenti interfejsa i opise naprednih svojstva pretrage. Poseban doprinos knjige je istaknut nizom preporuka za dizajn SUI. Saračević je u (Saracevic, 1997) prezentovao slojeviti model za izgradnju interaktivnog IR, gde na dizajn korisničkog interfejsa utiču kako faktori korisničkog ponašanja i očekivanja tako i faktori tehnologija (hardvera i softvera). Rad (Wellhausen, 2006) opisuje model interfejsa zasnovan na *patern* jeziku (pattern language). Kroz 14 *paterna* opisuju se dizajn alternative za SUI koji bi korisnicima trebao da omogući lako kreiranje upita pretrage. Wellhausen navodi da su neki od *paterna* (npr. Dynamic Search Dialog) zbog svoje kompleksnosti implementacije pogodni samo za *richclient* aplikacije. Uzimajući u obzir trend napretka tehnologija (npr. pojavu Web 2.0 tehnologije kao što je AJAX), moguće je danas sve *paterne* primeniti i na veb aplikacije.

Pretraga korisnicima može biti olakšana uvođenjem različitih vidova funkcionalnosti. Relevantan primer prethodnog moglo bi biti uvođenje tkz. „Rečnika“ (*thesarus*) u korisničkim interfejsima (Shiri and Revie, 2005). Na taj način se ubrzava pretraga po terminima koji su unapred poznati i koji imaju određeni vid hijerarhije (npr. spisak naučnih institucija i organizacija u jednoj državi).

Sveobuhvatna analiza raznovrsnih sistema pretraga koji sadrže naučno-istraživačke podatke data je u (Penca et al., 2014b). U pomenutom radu analizirani su institucionni repozitorijumi, baze naučnih podataka, CRIS sistema i portala pretrage naučnih podataka. U rezultatima rada je predložen sistem pretrage koji bi se oslanjao na CRIS profil pretrage baziran na SRU/W standardu i set modova i funkcionalnosti koje se svrstavaju kao nezaobilazne za sistem pretraga naučno-istraživačkih podataka. Rad (Smith, 2000) opisuje poželjne funkcionalnosti sistema pretrage kod digitalnih biblioteka. Upotreba različitih operatora (boolean, odsecanja, fraza, blizinskih) i različitih modova pretrage razmatraju se kao neophodne za efektivno pretraživanje digitalnih biblioteka.

U (Mi and Weng, 2008) se ističe da se *onlajn* katalozi u 21 veku ne mogu zamisliti ako nisu u skladu sa Web 2.0 specifikacijom. Web 2.0 predstavlja

značajan koncept u izradi WWW sadržaja koji je u potpunosti okrenut krajnjim korisnicima (Gavrilis et al., 2008). Neophodnost da se tehnologije Web 2.0 primenjuju u akademskim repozitorijumima navodi se u (Mi and Weng, 2008). U (Sastry and Reddy, 2010) se navodi da su Web 2.0 tehnologije nezaobilazne u konstrukciji visoko funkcionalnih interfejsa, koji bi u potpunosti bili intuitivni njihovim korisnicima (*Rich User Interfaces*). Takođe, u radu je navedeno da se iskoristivost (*usability*) postojećih digitalnih biblioteka može povećati ako se koriste Web 2.0 tehnologije. Na osnovu zahteva korisnika implementiran je korisnički interfejs u skladu sa MVC paternom, koristeći pri tom JSF tehnologiju.

Jedna od najznačajnijih tehnologija u razvoju Web 2.0 baziranih aplikacija je svakako i AJAX. U (Cavaleri, 2008) se navodi da se aplikacije bibliotekskih sistema mogu učiniti umnogome fleksibilnije sa stanovišta korisničkog interfejsa upotrebom AJAX. Iz ugla korisnika AJAX doprinosi konstruisanju veb aplikacija visokog odziva (*fluid application*). Posebno se ističe prednost ove tehnologije pri unosu terma u polje pretrage, gde se nakon unetih par karaktera korisniku prikazuju potencijalne konačne vrednosti samog terma u vidu padajuće liste. Rad (Wusteman and O'Hiceadha, 2006) ističe kompletan AJAX frejmovork pod nazivom OJAX. Cilj OJAX je da obezbedi visoko funkcionalan interfejs koji predstavlja osnovu sistema pretrage metapodataka. Autori ujedno ističu da se i sama pretraga najpoznatijeg servisa pretrage Google značajno unapredila nakon dodavanja dinamičkih predloga pretrage koji je usklađen sa onim što korisnik trenutno unosi u polje pretrage. Autor (Nicollet, 2009) zagovara da se korišćenjem tehnologije AJAX može konstruisati tkz. „inkrementalna pretraga“. Ovakav vid pretrage se u potpunosti razlikuje od tradicionalne *delimited* pretrage, u kojoj korisnik prvo mora u potpunosti da formira upit, a da tek nakon toga po slanju zahteva, dobije rezultate pretrage. Savremeni i interaktivni pristup predstavlja inkrementalna pretraga, gde se nakon svakog unetog karaktera (koji je samo deo upita) korisniku prikazuju mogući rezultati, koji su u skladu sa unetim karakteristikama.

Korišćenje Apache Lucene tehnologije u konstruisanju *backend* segmenta sistema pretrage izložen je u (Milosavljevic et al., 2010). *Search engine* baziran na Apache Lucene biblioteci korišćen je za pronalaženje bibliografskih zapisa BISIS bibliotečkog sistema. Apache Lucene je javno dostupna biblioteka implementirana u Javi, namenjena pretraživanju teksta.

Uticaj i rasprostranjenost Lucene se svakako ogleda i u činjenici da se koristi kao tekst server u softverskom paketu DSpace.

U radu (Schatz, 1997) se ističe da su sistemi za pretragu digitalnih repozitorijuma počeli svoj razvoj od ideje tekstualne pretrage još iz 1960 da bi danas evoluirali u distribuirane sisteme koji podržavaju sofisticirane semantičke pretrage najrazličitijih tipova podataka. Kako se povećava broj korisnika i količina podataka koja se pretražuje, tako je sve više i distribuiranih sistema za pretragu i pribavljanje podataka što se navodi u sledećim radovima (Callan et al., 1995; Lagoze and Fielding, 1998 ;Nicholas Eric Craswell, 2000; Bender et al., 2005; Milosavljević and Tešendrić, 2010).

Neophodnost za standardizacijom pretrage je opisana u nizu radova. U (McCallum, 2006) i (Baranov et al., 2000) se navodi da je Z39.50 najkorišćeniji standard u domenima pretrage i da predstavlja esencijalni element svake savremene biblioteke. Jedan od mnogobrojnih primera korišćenja Z39.50 standarda opisan je i u (Hafezi, 2008), gde on primenjuje za pretragu i povezivanje Iranskih biblioteka. Radovi kao što su (Morgan, 2004), (Anuradha et al., 2011), i (Zaric et al., 2012) svedoče o upotrebi standarda SRU/W. Zaric opisuje klijentsku aplikaciju koja poseduje mogućnost povezivanja sa udaljenim serverima po protokolu Z39.50 ili SRU/W i paralelnu pretragu podataka udaljenih servera. U (Hoppe and Razum, 2008) su prikazani detalji implementacije SRU/W standarda u digitalnoj biblioteci Fedora. U (Boberić Krstićev, 2013) je opisan "mediator/wrapper" pristup koji omogućuje distribuiranu pretragu bibliotekskih sistema korišćenjem različitih standarda (Z39.50, SRU/W). Prilikom pretrage korisnik zadaje upit po nekom od podržanih standarda a softverska komponenta "mediator" pretvara, ako je to potrebno, u oblik koji podržava biblioteka iz koje se pretražuju podaci.

Rad (Sander-Beuermann et al., 2008) navodi da se u CRIS sistemima često zanemaruje funkcionalnost pretrage, što svakako umanjuje njihovu upotrebnost vrednost. CRIS sistemi sadrže velike količine podataka koji se mogu različito interpretirati u pojedinačnim CRIS sistemima. Jasno se uočava problem u pretraživanju podataka više takvih sistema istovremeno. EuroCRIS predlaže uvođenje CERIF kao jedinstvenog standarda za prevazilaženje problema pretrage podataka. Ističe se da bi najefikasniji metod za pretragu CRIS sistema bio na osnovu meta podataka.

U (Jeffery, 2007) se navodi lista zapisa CRIS sistema za koje je potrebno implementirati funkcionalnost pretrage. CRIS sistemi bazirani na CERIF standardu treba da omoguće pretragu: istraživača, organizacionih jedinica (institucija), projekata, publikacija, produkata, patenata, opreme, događaja (konferencije i sl.) i sponzora.

Karakteristično za sva prikazana istraživanja je da su, iako se bave pojedinačnim specifičnim sistemima koji sadrže delove podataka o naučno-istraživačkoj delatnosti, rezultati istraživanja u značajnoj meri saglasni po pitanju poželjnih funkcionalnosti i indeksa sistema pretrage

Zaključci izvedeni na osnovu pregledane literature su sledeći:

- Pokazano je da nije dovoljno koristiti opšte pretraživače za pretragu naučno-istraživačkih podataka jer oni ne indeksiraju sve podatke iz naučno-istraživačke oblasti.
- Istaknut je značaj distribuiranih sistema za pretragu i uočen je nedostatak upotrebe standarda u domenu pretrage podataka naučno-istraživačke delatnosti, te će se poseban akcenat staviti na implementaciji istog u sistemu pretrage koji je predmet istraživanja.
- Postojeći sistemi prilikom implementacije intenzivno koriste napredne tehnologije kao što su Web 2.0, AJAX i Lucene. Pomenute tehnologije pružaju pogodnosti prilikom implementacije korisničkog interfejsa i komponente koja pribavlja tražene podatke iz sistema. Tako upotrebom Web 2.0 i AJAX tehnologija stvara okruženje za konstrukciju vizuelno bogatog SUI sa visokim stepenom performansi i mogućnosti. Što se tiče jezgra pretrage ono bi trebalo da se bazira na nekom od široko korišćenih standarda pretrage (Z39.50, SRU/W). Sama implementacija jezgra pretrage bila bi znatno olakšana ukoliko bi se koristile tehnologije koje su uveliko u upotrebi (npr. Lucene tekst server).
- Ustanovljeno je da se pri implementaciji sistema pretrage **ravnomerno** treba fokusirati kako na **korisnički interfejs** tako i na samo **jezgro sistema** koje izvršava pretragu. SUI treba da bude (koliko god je to moguće) u skladu sa korisnikovim očekivanjima, što direktno ukazuje da treba podržati niz taktika koji korisnici koriste prilikom pretrage. Takođe je neophodno obezbediti

modularni SUI koji se sastoji iz više sekcija koje su različitih nivoa kompleksnosti i mogućnosti sa ciljem da se odgovori na zahteve svih vrsta korisnika (*basic, advance* tipovi korisnika).

- U cilju uvođenja standarda, a pre same implementacije sistema pretrage, bilo je neophodno izvršiti detaljnu analizu relevantnih sistema koji poseduju neki vid pretrage podataka naučno-istraživačke delatnosti.

1.2. Predmet i cilj istraživanja

Predmet istraživanja je sistem za pretragu naučno-istraživačkih rezultata. Na osnovu pregleda vladajućih stavova konstatovano je da je od velike važnosti da postoji dostupnost naučno-istraživačkih rezultata. Zato je na Prirodno-matematičkom fakultetu i Fakultetu tehničkih nauka Univerziteta u Novom Sadu započet 2009. godine razvoj informacionog sistema naučno-istraživačke delatnosti Univerziteta u Novom Sadu. Prva faza razvoja ovog sistema bilo je modeliranje i unos podataka o publikovanim naučnim rezultatima. Testiranje i verifikacija sistema urađena je na rezultatima istraživača sa Prirodno-matematičkog fakulteta u Novom Sadu. Ukupan broj svih vrsta zapisa je početkom 2013. godine bio oko 73.000. Druga faza razvoja ovog sistema započeta je 2011. godine, koja se odnosi na pretraživanje naučnih rezultata što je i tema ove doktorske disertacije. Stoga su osnovni motivi ovog istraživanja bili da se:

- obezbedi javni uvid i pretraga podataka o institucijama/organizacijama, istraživačima i publikovanim naučnim rezultatima unutar Univerziteta u Novom Sadu, i to proširenjem CRIS sistem za unos i skladištenje naučno-istraživačkih podataka univerziteta u Novom Sadu.
- iskoriste opšte prihvaćeni standardi iz domena pretrage kako bi se stvorio preduslov za povezivanje sličnih sistema koji sadrže naučno-istraživačke podatke.

Cilj istraživanja u okviru doktorske disertacije je specifikacija, modeliranje i implementacija sistema pretrage naučno-istraživačkih podataka. Sistem pretrage treba da omogući bogatu i efikasnu pretragu naučno-istraživačkih

podataka. Osnovna ideja je da se pretraga zasniva na bibliotečkim standardima SRU/W i CQL (CQL: the Contextual Query Language: Specifications, 2008) kako bi se mogli prihvatati zahtevi za pretragu sa udaljenih računara, a da korisnici sistema mogu da pretražuju putem veb forme implementirane upotrebom Web 2.0 tehnologije pri čemu se od korisnika ne očekuje poznavanje pomenutih standarda.

Polazne hipoteze u istraživanju su bile da je moguće:

- formirati unifikovani profil pretrage za CRIS sisteme. Specificirani profil pretrage bio bi zasnovan na nekom široko prihvaćenom standardu kojim bi se omogućila pretraga sistema naučno-istraživačke delatnosti na jedinstveni način.
- implementirati standardizovani sistem pretrage za CRIS sisteme, koji bi korisnicima omogućio unifikovanu, semantički bogatu i efikasnu pretragu naučno-istraživačkih podataka. Glavne karakteristike sistema bile bi: interfejs baziran na Web 2.0 tehnologijama, implementirani profil pretrage, mogućnost zadavanja upita upotrebom standardnog upitnog jezika.
- omogućiti udaljeno pretraživanje eksternim sistemima zadavanjem adekvatnih zahteva u skladu sa široko prihvaćenim protokolom za ovu namenu kao što je SRU/W.

Rezultati istraživanja u ovoj disertaciji su potvrdili navedene polazne hipoteze. Specificiran i implementiran je sistem za pretragu naučno-istraživačkih rezultata baziran na Web 2.0 tehnologijama i upotrebi CQL upitnog jezika koji se koristi na Univerzitetu u Novom Sadu (poglavlje 5). Razvijen je unifikovani profil pretrage za CRIS sisteme i implementiran je standardizovani sistem pretrage naučno-istraživačkih podataka (poglavlje 3). Specificiran i implementiran je servis koji omogućuje pretragu sa udaljenih računara upotrebom SRU/W protokola (poglavlje 6).

2. Analiza sistema pretraga

Na osnovu pregledane literature došlo se do smernica za analizu sistema koji sadrže naučno-istraživačke podatke. Kompletan spisak analiziranih sistema nalazi se u **Prilogu 1 - Analizirani sistemi pretraga**.

Analizirane su četiri grupe sistema: Institucionalni repozitorijumi (IR), baze naučnih sadržaja, CRIS sistemi i portali pretrage. Svi sistemi su analizirani iz sledećih aspekata:

1. dostupni indeksi
2. modovi pretrage (jednostavna, napredna, tekstualna)
3. podržane vrste upita (logički upit, pretraga po udaljenosti, pretraga nedovršenih reči/rečenica – upotreba džoker karaktera, približna pretraga i pretraga fraze).

Rezultati analize prikazani su u nastavku ovog odeljka. Analiza je publikovana u naučnom radu (Penca et al., 2014b). Na osnovu izvršene analize predložene su neophodne funkcionalnosti i relevantni set indeksa koji pretraga CRIS sistema treba da podrži. Predložen je novi profil za pretragu podataka naučno-istraživačke delatnosti (CRIS profil - **odeljak 5** Udaljeno pretraživanje podataka iz CRIS UNS).

2.1. Analiza indeksa

Indeksi pretrage predstavljaju nezaobilazan aspekt za analizu obzirom da oni predstavljaju kriterijume na osnovu kojih se vrši pretraga.

Prvi korak je bio da se izdvoje indeksi iz svake od pomenutih grupa sistema. Većina indeksa se u okviru jedne grupe podudara, međutim postoje indeksi koji su karakteristični samo za neke od analiziranih sistema. U nastavku je u okviru podsekcija data pojedinačna analiza za svaku od grupa.

Institucionalni repozitorijumi. Analizirani IR se oslanjaju na DSpace ili EPrints digitalne repozitorijume. Sistemi zasnovani na DSpace podrazumevaju da su metapodaci (indeksi) definisani u skladu sa DC LAP profilom (DC-Library Application Profile (DC-LAP), 2007). U sistemu EPrints u samoj instalaciji postoji predefinisani set indeksa koji se lako može proširiti. Obzirom da svaki IR može definisati svoj skup indeksa, bilo

je neophodno analizirati više predstavnika IR koji se baziraju na DSpace ili EPrints digitalnim repozitorijumima (tabela 2.1).

Baze naučnog sadržaja. Baze naučnog sadržaja predstavljaju informacione sisteme koji su implementirani kao digitalne biblioteke, baze punog teksta i/ili apstraktne i citatne baze podataka. Indeksi baza naučnih sadržaja podržavaju pretragu različitih tipova naučno-istraživačkih podataka. Najveći broj indeksa se očekivano odnosi na pretragu podataka o publikacijama, obzirom da je većina sistema iz ove grupe u vlasništvu izdavačkih kuća specializovanih za naučnu literaturu (tabela 2.1). Pored indeksa za pretragu publikacija u sistemima *IEEE Computer Society* i *SciVerse Scopus* indentifikovani su i specifični indeksi za pretragu autora (tabela 2.2), a u sistemu *SciVerse Scopus* i indeksi za pretragu institucija (tabela 2.3).

CRIS sistemi. Pretraga CRIS sistema obuhvata različite tipove naučno-istraživačkih podataka i tako u većini sistema postoje indeksi vezani za publikacije, autore, institucije i projekte navedeni u tabelama 2.1, 2.2, 2.3, 2.4. Ovi sistemi imaju mali broj indeksa po kojima je moguće pretraživati podatke, sa izuzecima koji nastaju prilikom specijalizacije CRIS sistema za određeni tip naučno-istraživačkog podataka (kao što je *USDA CRIS* koji podržava veliki broj indeksa namenjenih isključivo za pretragu projekata) ili neki od komercijalnih CRIS sistema (npr. *Pure* i *Converis*) (tabela 2.4).

Portali pretrage naučno-istraživačkih podataka. Pripadnici četvrte grupe analiziranih sistema pretražuju preuzete zapise iz drugih sistema po indeksima navedenim u tabeli 2.1. Analizom sistema *DRIVER* i pregledom njegove relevantne dokumentacije ustanovljeno je da sistem za definisanje metapodataka (indeksa) koristi *Dublin Core* rečnik. U sistemu *Google Scholar* postojeći indeksi, pored pretrage publikacija, omogućuju i pretragu tehničkih izveštaja, teza i disertacija i veb resursa koji se odnose na naučno-istraživačke podatke. Sistem *Microsoft Academic Search* dozvoljava da se indeksi za pretragu autora, časopisa, konferencija i organizacija mogu upotrebiti pri pretrazi publikacija. Na primer, pretraga po indeksu autor može rezultovati relevantnim zapisima tipa autor kao i njihovim publikovanim materijalima.

Rezultati analize

Kako bi zaključili koji su indeksi relevantni za naš sistem, izvršena je sistematizacija indeksa analiziranih sistema. Sistematizacija indeksa

podrazumevala je njihovo razvrstavanje po određenim kategorijama. Kategorije su formirane tako da reprezentuju najrelevantnije tipove zapisa iz naučno-istraživačkog domena. Izdvojene kategorije odnose se na entitete publikacija, autor, institucija i projekat i prikazane su u tabelama **2.1, 2.2, 2.3, 2.4**. Zaglavlja kolona u tabelama sadrže nazive analiziranih sistema, dok zaglavlja vrsta definišu njihove identifikovane indekse. Pored imena sistema postoje brojevi koji ukazuju na njihovu poziciju u dodatku A. U svakoj tabeli postoji izdvojena kolona *Frequency* koja prikazuje procenat sistema koji podržavaju određeni indeks. Pre početka detaljnijeg objašnjavanja tabela, ističe se da su iz zaglavlja kolona izostavljeni svi analizirani sistemi koji ne poseduju posmatranu kategoriju naučno-istraživačkog zapisa i samim tim oni neće uticati na formiranje statistike indeksa. Sve tabele u kojima je prikazana statistika indeksa su sortirane u odnosu na parametar *Frequency*.

Pored pomenute kategorizacije po tipovima entiteta, indeksi su kategorisani i kao „**obavezni**“, „**poželjni**“ i „**neobavezni**“. Kategorija obaveznih izdvaja esencijalni skup indeksa za pretragu publikovanih naučno-istraživačkih rezultata. Ujedno ova kategorija omogućuje pretragu podataka koji su u „neposrednoj vezi“ sa navedenim publikovanim rezultatima, tj. povezivanje publikacija sa ostalim vrstama zapisa (institucijama, autorima) iz naučno-istraživačkog domena. U kategoriju poželjnih indeksa su ubrajaju oni koji predstavljaju specijalizaciju indeksa iz grupe „obaveznih“. Naime to su indeksi koji se često mogu opisati kombinacijom indeksa iz grupe obaveznih. Kategorija neobaveznih indeksa sadrži one koji su zastareli, povučeni ili se vrlo retko koriste (samo u nekim specifičnim situacijama) u analiziranim sistemima. Kategorizacija je napravljena u skladu sa pojavom (frekvencijom) indeksa u analiziranim sistemima (tabele **2.1, 2.2., 2.3, 2.4**) i na osnovu višegodišnjeg iskustva autora ove disertacije u razvoju CRIS sistema.

	Dspace										Eprints										Pure			Converis			Frequency											
	Cambridge ¹	MARS ²	MIT ³	Edinburgh ⁴	ROSE ⁵	NSU ⁶	BeSU ⁷	UnisaIR ⁸	UJF Repository ⁹	Kingston ¹⁰	MBRO ¹¹	Southampton ¹²	PiDUA ¹³	UJFAS ¹⁴	Nottingham ¹⁵	EEE ¹⁶	SpringerLink ¹⁷	Emerald ¹⁸	Scopus ¹⁹	WARCIS ²⁰	Cristin ²⁴	FRIS ²⁵	OpenAIRE ²⁷	GTR ²⁸	EuroRiset ²⁹	Royal Holloway ³⁰		Aalborg ³¹	HuGent ³²	MU Sweden ³³	WWU Münster ³⁴	Hul ³⁵	DBN ER ³⁶	Google Scholar ³⁷	Microsoft ³⁸			
Title	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	100%		
Author	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	91%		
Abstract	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	82%		
Date						+		+																												74%		
Research Unit (Institution/ Department/ Research	+	+	+	+	+	+	+		+	+	+	+	+	+					+	+							+	+	+	+	+	+	+			+	74%	
Identifier (local, doi, issn, isbn)	+	+	+	+	+	+	+	+				+	+			+	+	+	+	+	+	+	+	+			+	+	+	+	+	+	+	+	+	+	71%	
Keyword		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+												68%	
Type	+								+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	68%	
Source (Journal/Event /Conference/ Series)										+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	59%	
Full text			+		+	+		+		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	53%	
Journal name																											+	+	+	+	+	+	+	+	+	+	50%	
Language	+			+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	41%	
Conference name												+																+	+	+	+	+	+	+	+	+	38%	
ISBN/ISSN	+					+		+								+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	38%	
Series		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	38%	
Contributor			+		+					+	+		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	35%	
Publisher						+		+				+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	35%	
Research area												+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	35%	
Subject	+	+		+		+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	29%
Refereed (peer-review)	+				+				+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	26%	
Editor									+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	24%	
Format (Mime-Type)			+		+	+		+			+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	21%	
Sponsor	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	21%	
Status									+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	21%	
Repository																																				15%		
ISSUE																																					12%	
Page																																					12%	
Advisor (Supervisor)			+				+		+																												9%	
FirstAuthor																																					9%	
Affiliation county																												+	+								6%	
Content title (chapter)																																					6%	
Presentation type																																					6%	
Project																																					6%	
Volume																																					6%	
Affiliation city																	+	+																			3%	
Author email										+																											3%	
CAS number																																					3%	
Chemical name																																					3%	
CODEN																																					3%	
Presentation title																																					3%	
References																																					3%	

Tabela 2.1 Statistika indeksa za kategoriju publikacija

U tabeli **2.1** navedeni su indeksi kojima je moguće opisati bilo kakav vid **naučne publikacije**. Indeksi ove tabele formirani su spajanjem indeksa iz pretraga različitih tipova publikovanih rezultata. Tako *IEEE Computer Society* i *Emerald* sistemi, pored pretrage publikacija sadrže i pretrage radova u časopisima, knjiga, saopštenja na konferencijama, itd. Posmatrajući kolonu **Frequency** lako se može zaključiti koji su indeksi relevantni za pretragu publikacija. U odnosu na parametar učestalost indeksi se mogu svrstati na obavezne (neophodne), poželjne i neobavezne. Tako su prvih 12 indeksa iz tabele **2.1** (počevši od indeksa *Title* zaključno sa *Language*). Devet indeksa iz Tabele **2.1** koji slede su kategorisani kao poželjni (od indeksa *Conference Name* do indeksa *Editor*). opisane su specijalizacije indeksa iz grupe neophodnih. Na primer *ISSN/ISBN* iz ove kategorije se može opisati i indeksom *Identifier* iz kategorije neophodnih. Ostali indeksi iz tabele **2.1** su kategorisani kao neobavezni. Adekvatan primer neobaveznih indeksa bio bi indeks *CODEN* (tip bibliografskog koda koji više nije u upotrebi) koji se pojavljuje sa učestalošću od samo 4%.

U poređenju sa pretragom publikacija, **pretraga zapisa autora** je realizovana u nešto manje od polovine (16 od 38 u tabeli **2.2**) analiziranih sistema, iako većina sistema skladišti podatke o autorima. Kao relevantan primer može se uzeti sistem *DSpace* koji u okviru svog modela podataka skladišti informacije o autorima ali ne pruža mogućnost njihove pretrage. Činjenica je da većina CRIS sistema podržava pretragu autora. Evidentno je da ne postoji ni veliki broj indeksa koji se koriste pri pretrazi ove vrste naučno-istraživačkih zapisa, samim tim nam parametar *Frequency* ne može puno pomoći pri odabiru indeksa autora relevantnih za pretragu. Po parametru *Frequency* jedini neophodan indeks bi bio *Name*. Indeks *Name* ukazuje na ime i prezime Autor/Istraživača. Za potrebe CRIS profila (poglavlje **5** Udaljeno pretraživanje podataka iz CRIS UNS) je usvojeno da se indeksi *Affiliation*, *Research Area*, *Identifier*, *Description* i *Keywords* kategorišu kao poželjni pri pretrazi zapisa autora.

	IEEE ¹⁶	Scopus ¹⁹	NARCIS ²⁰	E-CRIS SR ²³	SICRIS ²²	Cristin ²⁴	FRIS ²⁶	GTR ²⁷	EuroRISNet ²⁸	Pure ³⁰	Converis ³¹	Royal Holloway ³⁰	Aalborg ³¹	HoGent ³²	Mid Sweden ³³	WWU Münster ³⁴	Hull ³⁴	Microsoft ²⁸	Frequency
Name	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	100%
Affiliation		+	+			+	+			+	+	+	+	+					44%
Research area		+	+	+	+		+						+	+					44%
Identifier			+	+	+					+	+	+	+	+					38%
Researcher profile/ Description										+	+	+	+	+			+		31%
Keywords			+	+	+										+				25%
First name (Name)		+						+											13%
Last name (Name)		+						+											13%
Position			+						+										13%
Status							+				+								13%
Gender						+													6%
Grants / Prizes			+																6%
Location (region / country)						+													6%

Tabela 2.2 Statistika indeksa za kategoriju autor

Slično kao kod pretrage zapisa o autorima, **pretraga institucija** je realizovana u CRIS sistemima, kao što je prikazano u tabeli 2.3. Iako je identifikovano da se u sistemima baziranim na *DSpace* repozitorijumu nalazi pretraga organizacija ona nije prikazana u statistici indeksa. Razlog izostavljanja *DSpace* iz statistike je činjenica da ne postoje izdvojeni indeksi za pretragu organizacija već se u tu svrhu koriste indeksi rezervisani za pretragu publikacija. Na osnovu statistike indeksa, predloženo je da se indeks *Name* kategoriše kao neophodan, dok bi se *Description*, *Identifier* i *Research Area* svrstali kao poželjni indeksi.

	Pure													Converis			
	Scopus ¹⁹	NARCIS ²⁰	E-CRIS SR ²³	SICRIS ²²	Cristin ²⁴	FRIS ²⁶	GTC ²⁷	EuroRisNet+ ²⁹	Royal Holloway ³⁰	Aalborg ³¹	HoGent ³²	Mid Sweden ³³	WWU Münster ³⁴	Hull ³⁴	Microsoft ³⁸	Frequency	
Name	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	100%	
Organisation profile/ Description		+				+			+	+		+	+	+		47%	
Identifier			+	+					+	+	+					33%	
Research area		+	+	+		+								+		33%	
City/Country			+	+				+						+		27%	
Head of organisation (Author)			+	+	+											20%	
Type										+	+					13%	
Keywords						+										7%	
Researcher (Author)					+											7%	
Status						+										7%	
Subject					+											7%	

Tabela 2.3 Statistika indeksa za kategoriju institucija

Pretraga projekata je karakteristična samo za CRIS sisteme. Sami projekti se opisuju na najrazličitije načine u zavisnosti od institucije koja ih definiše i oblasti za koju su vezani. Iako samo nešto manje od polovine analiziranih informacionih sistema podržavaju pretragu projekata, u tabeli **2.4** navedeno je čak 28 različitih indeksa. Predloženo je da se indeksi *Name*, *Description*, *Affiliation*, *Date*, i *Identifier* posmatraju kao neophodni za pretragu projekata. Poželjni indeksi bili bi *Keywords*, *Research Area*, *Author* i *Status*.

	NARCIS ²⁰	USDA CRIS ²¹	E-CRIS SR ²³	SICRIS ²²	Cristin ²⁴	HunCRIS ²⁵	FRIS ²⁶	OpenAIRE ²⁷	GTR ²⁸	EuroRISNet+ ²⁹	Royal Holloway ³⁰	Aalborg ³¹	HoGent ³²	Mid Sweden ³³	WWU Münster ³⁴	Frequency
Name	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	100%
Abstract/Description	+								+	+	+	+	+	+	+	60%
Affiliation	+	+			+	+	+				+	+	+			60%
Date		+			+		+		+		+	+	+		+	53%
Identifier	+	+	+	+				+			+	+	+			53%
Keywords		+	+	+		+	+								+	40%
Research area			+	+		+	+			+			+			40%
Researcher (Author)	+	+			+	+					+	+				40%
Status	+	+		+			+		+		+					40%
Head of project (Author)	+		+	+							+	+				33%
Type	+	+									+		+			27%
Classification		+							+	+						20%
Country (Location)		+								+						13%
Full text		+												+		13%
Funding programme						+						+				13%
Grant No. (Identifier)		+						+								13%
Location		+				+										13%
Project ID (Identifier)	+	+														13%
Acronym								+								7%
Agency		+														7%
City (Location)		+														7%
Division/Station Code		+														7%
Grant year (Date)		+														7%
Ortelius thesaurus						+										7%
Project No (Identifier)		+														7%
Region (Location)		+														7%
Result type						+										7%
Subject					+											7%

Tabela 2.4 Statistika indeksa za kategoriju projekat

2.2. Analiza modova pretrage

Modovi pretrage predstavljaju niz grupisanih funkcionalnosti koje pruža sistem pretrage. Tako sistemi pretraga najčešće sadrže više različitih modova. u zavisnosti od vrste (tipova) korisnika koji koriste sam sistem pretrage.

U okviru analize tipova (modova) pretrage ustanovljeni su modovi **jednostavne, napredne** pretrage i mod **tekstualnog zadavanja upita**. Jednostavna pretraga izvršava se upotrebom samo jednog terma koji se vezuje za određeni/e indeks/e pretrage. Napredna pretraga omogućava korisniku kreiranje složenih upita pretrage. Ona se definiše nizom indeksa za

koji se navode vrednosti termova, pri čemu je moguća upotreba logičkih operatora. Poseban mod tekstualnog zadavanja upita podrazumeva da korisnik zadaje tekst u struktuiranoj formi koji je u skladu sa propisanom specifikacijom upita servera pretrage. Podržani modovi svih analiziranih sistema nalaze se u tabeli **2.5**.

IR. IR po pravilu podržavaju modove jednostavne i napredne pretrage, dok poseban mod tekstualnog zadavanja upita nije dostupan ni u jednom od analiziranih IR. Međutim, kuriozitet predstavljaju sistemi bazirani na *DSpace* u kojima se mogu definisati upiti pretrage bazirani na sintaksi upitnog jezika *Lucene* tekst servera.

Baze naučnog sadržaja. Slično institucionalnim repozitorijumima sve testirane baze naučnog sadržaja podržavaju jednostavnu i naprednu pretragu. Izuzetak je *SciVerse Scopus* koji pored pomenutih pretraga podržava i pretragu tekstualnog zadavanja upita. Tekstualni mod pomenutog sistema okarakterisan je posebnom sintaksom i omogućava korišćenje dodatnog seta indeksa koji nije podržan putem ostalih modova. U slučaju da sistemi sadrže više tipova zapisa, za svaki ponaosob obezbeđena je jednostavna i/ili napredna pretraga (npr. postoji posebna pretraga za publikacije, autore, institucije, ...). Obzirom da pojedini sistemi iz ove grupe (*Emerald Group Publishing Limited*) podržavaju čak i više različitih oblika publikacija (radovi u časopisu, knjige, publikacije, ...) indentifikovane su posebne jednostavne i napredne pretrage za svaki od oblika publikacija (npr. odvojene su pretrage radova u časopisu od saopštenja na konferencijama).

CRIS sistemi. U CRIS sistemima je jednostavna pretraga uglavnom podržana po nekom od dostupnih naučno-istraživačkih zapisa. Izuzetak su *HunCRIS* i *USDA CRIS* koji ne podržavaju jednostavnu pretragu. U sistemu *Cristin* realizacija moda jednostavne pretrage se razlikuje u odnosu na druge sisteme. Jednostavna pretraga u pomenutom sistemu dozvoljava upotrebu više od jednog indeksa što je zapravo odlika moda napredne pretrage. Modom jednostavne pretrage obuhvaćeni su osnovni indeksi, dok su u modu napredne pretrage podržani svi indeksi odabranog tipa zapisa u sistemu *Cristin*. *USDA CRIS* podržava moćan tekstualni mod zadavanja upita. Korišćenje moda pretrage tekstualnim zadavanjem upita zahteva poznavanje sintakse i semantike internog upitnog jezika. Broj indeksa koji se mogu koristiti pri tekstualnom zadavanju upita proširen je u odnosu na indekse iz

napredne pretrage. Slično *DSpace* baziranim sistemima, *Pure* i *Converis* omogućuju zadavanje *Lucene* upita pretrage.

Portali pretrage naučno-istraživačkih podataka. U svim analiziranim naučno-istraživačkim portalima podržani su modovi jednostavne i napredne pretrage. Jednostavna pretraga portala *GoogleScholar* razlikuje se od ostalih pretraga iz posmatrane grupe sistema jer se realizuje po indeksu pun tekst. U modu jednostavne pretrage *Microsoft Academic Search* zadaje se samo vrednost terma na osnovu kojeg se mogu dobiti rezultati sačinjeni od svih podržanih tipova zapisa. Treba istaći da se jednostavna pretraga pomenutog sistema može ograničiti odabirom indeksa domen (naučna oblast). Napredna pretraga *GoogleScholar* pored mogućnosti odabira indeksa pruža i set mogućih odnosa i relacija koji će se detaljnije prikazati u odeljku **2.3** (Analiza funkcionalnosti). Sistem *Microsoft Academic Search* je specifičan po pitanju modova obzirom da oni nisu u potpunosti odvojeni kao što je to bio slučaj u ostalim sistemima. Napredna pretraga proširuje jednostavnu pretragu tako što omogućuje specifikaciju određenih indeksa za navedenu vrednost terma. Mod tekstualnog zadavanja upita podržan je u sistemima *Microsoft Academic Search* i *GoogleScholar*. Kod portala *GoogleScholar* pretraga tekstualnim zadavanjem nastaje proširenjem moda jednostavne pretrage definisanjem specifičnih konstrukcija (npr. Pretraga autora pod imenom John definišemo izrazom autor:“John“), koje će biti transformisane u GQL upit (Google Query Language - <https://developers.google.com>). Tekstualni mod pretrage sistema *GoogleScholar* je poprilično skroman jer se bazira na svega nekoliko koncepata iz GQL. Slično kao kod *GoogleScholar* tekstualni mod pretrage realizovan je u okviru jednostavne pretrage. Tekstualni mod pretrage *Microsoft Academic Search* karakteriše upitni jezik specifične sintakse koji je dokumentovan u sistemu pomoći samog servisa.

Rezultati analize

Svi sistemi bazirani na *DSpace*, *EPrints*, *Pure* i *Converis* predstavljeni su samo sa jednim predstavnikom u tabeli obzirom da poseduju istovetne mogućnosti. Posmatrajući tabelu **2.5** primećuje se da je mod jednostavne pretrage implementiran kod skoro svih posmatranih sistema. U slučaju da sistem sadrži više tipova zapisa, za svaki ponaosob obezbeđena je jednostavna pretraga. Mod napredne pretrage predstavlja nezaobilaznu stavku svakog ozbiljnog sistema za pretragu zapisa. Sama napredna pretraga

podrazumeva formiranje složenih upita za pretragu jer pruža mogućnost upotrebe većeg broja indeksa. U realizaciji moda napredne pretrage najdalje se otišlo u sistemima *SciVerse Scopus* i *DRIVER* gde je korisniku omogućeno da samostalno zadaje „neograničen“ broj ponavljanja nekog indeksa pri konstrukciji upita pretrage. Obzirom na statistiku navedenu u tabeli 2.5 moglo bi se zaključiti da je tekstualni mod pretrage neobavezna stavka u sistemima pretraga. Prethodnu konstataciju treba uzeti sa velikom rezervom jer nepostojanje tekstualnog moda pretrage u analiziranim sistemima je zapravo rezultat komplikovanog procesa njene implementacije. Testiranjem tekstualnog moda pretrage kod sistema koji je implementiraju (*USDA CRIS*, *SciVerse Scopus* i *DSpace*), prepoznate su suštinske prednosti ovakvog moda pretrage. Osnovna prednost tekstualnog moda u odnosu na ostale modove je mogućnost formiranja upita pretrage najvišeg nivoa složenosti. U analiziranim sistemima se kao potencijalni nedostatak navodi nepostojanje opšte prihvaćenog standarda na kome bi se bazirao upitni jezik pomenute pretrage. Svi sistemi koji poseduju tekstualni mod pretrage implementiraju neku vrstu sopstvenog upitnog jezika, osim *DSpace*, *Pure* i *Converis* koji se oslanja na upitni jezik *Lucene* tekst servera, koga možemo smatrati standardom kad je reč o tekstualnim pretragama. Standardizacijom upitnog jezika bi se de fakto omogućilo njegovo lakše savladavanje svim korisnicima i pružila bi se osnova za jednostavniju integraciju sa drugim sistemima pretraga. Kod sistema naučno-istraživačke delatnosti koji sadrže veliku količinu raznorodnih podataka bilo bi praktično neophodno implementirati tekstualnu pretragu.

Repositories	Modes		
	simple	advance	textual
DSPACE ¹⁻⁹	+	+	
EPRINTS ^{10,15}	+	+	
IEEE ¹⁶	+	+	
SpringerLink ¹⁷	+	+	
Emerald ¹⁸	+	+	
Scopus ¹⁹	+	+	+
NARCIS ²⁰	+		
USDA CRIS ²¹		+	+
E-CRIS SR ²²	+		
SICRIS ²³	+		
Cristin ²⁴	+	+	
HunCRIS ²⁵		+	
FRIS ²⁶	+	+	
OpenAIRE ²⁷	+	+	
GTR ²⁸	+		
EuroRIsNet+ ²⁹	+		
Pure ^{30,32}	+	+	
Converis ³³⁻³⁵	+	+	
DRIVER ³⁶	+	+	
Google Scholar ³⁷	+	+	+
Microsoft ³⁸	+	+	+
Frequency	90%	76%	19%

Tabela 2.5 Statistika modova pretraga

2.3. Funkcionalnosti pretrage

Funkcionalnost pretrage predstavlja atomičku akciju sistema pretrage. Korisnik specificira akciju korišćenjem dela interfejsa pretrage i/ili termova i/ili podržanih operatora.

Na osnovu pregleda literature dobijene su relevantne smernice za identifikaciju karakterističnih funkcionalnosti u sistemima pretraga. Tako se u analiziranim sistemima posmatrale sledeće funkcionalnosti: pretraga po udaljenosti, pretraga nedovršenih reči/rečenica, približna pretraga i pretraga fraza. U analiziranim sistemima se navedene funkcionalnosti razlikuju po načinu upotrebe (implementacije) u samim sistemima. Za analizu je od većeg značaja bilo identifikovati da li je neka funkcionalnost omogućena od toga

kako je ona imlementirana u konkretnom sistemu. Tako se za potrebe analize nije uzimala činjenica da se džoker karakteri mogu naći na različitim pozicijama u termu (ispred, iza, u sredini) prilikom definisanja pretrage nedovršenih reči/rečenica već da li je ta pretraga uopšte omogućena.

IR. U IR je izvedba svih funkcionalnosti vezana za digitalnu biblioteku na kojoj se baziraju (*DSpace* ili *Eprints*). Konkretno funkcionalnost pretrage sistema baziranih na *DSpace* predstavlja podskup mogućnosti *Lucene* tekst servera. Svi analizirani IR podržavaju pretragu fraza dok se pretraga po udaljenosti i pretraga nedovršenih reči i rečenica sreće samo kod sistema baziranih na *DSpace* digitalnoj biblioteci.

Baze naučnih sadržaja. Kod baza naučnih sadržaja izvedba funkcionalnosti je karakteristična za svaki sistem ponaosob, obzirom da ne postoji jedinstveni softverski sistem na kome se temelje sve baze naučnih sadržaja. Pretraga fraza i nedovršenih reči/rečenica je implementirana u svim bazama naučnih sadržaja. Približna pretraga je dostupna u sistemima *SpringerLink* i *Emerald Group Publishing Limited*. Pretraga po udaljenosti je realizovana u *Emerald Group Publishing Limited* i *SciVerse Scopus* upotrebom specifične sintakse dokumentovane u okviru korisničke pomoći pomenutih sistema.

CRIS sistemi. Realizacija funkcionalnosti u CRIS sistemima je nalik bazama naučnih sadržaja. Svi analizirani CRIS sistemi osim *HunCRIS*, *OpenAIRE* i *EuroRIsNet+* i *Observatory* podržavaju funkcionalnost nedovršenih reči i rečenica. Pretraga fraza je omogućena u svim analiziranim CRIS sistemima. Navedena pretraga je specifična u sistemima *Cristin* i *HunCRIS* jer se svi uneseni termini implicitno pretražuju kao fraze. Samo *HunCRIS*, *OpenAIRE* i *EuroRIsNet+* i *Observatory* podržavaju približnu (fuzzy) pretragu, dok je pretraga po udaljenosti omogućena kod *USDA CRIS*, *Fris*, *Gateway to research* i *Pure*. Bitno je istaći da je fazi pretraga u sistemu *EuroRIsNet+* i jedini oblik pretrage, tako da je svaka pretraga bazirana na fazi upitu.

Portali pretrage naučno-istraživačkih podataka. Analizirajući funkcionalnosti portala pretrage naučno-istraživačkih podataka evidentno je da samo *GoogleScholar* podržava sve funkcionalnosti dok ostali sistemi pružaju samo pretragu fraza. *GoogleScholar* omogućuje pretragu po udaljenosti upotrebom nedokumentovanog operatora NEAR. Navedeni operator se zadaje između dva terma pri čemu se mora definisati i njihova međusobna udaljenost.

Rezultati analize.

Svi sistemi bazirani na *DSpace* i *EPrints*, *Pure* i *Converis* predstavljeni su samo sa jednim predstavnikom u tabelama **2.5**, **2.6**, **2.7** obzirom da poseduju istovetne mogućnosti.

Segment kreiranja logičkih upita je izdvojen zbog specifičnosti upotrebe logičkih operatora u analiziranim sistemima. Upotreba logičkih operatora prikazana je u tabeli **2.6**. Svi logički operatori u tabeli razvrstani su u odnosu na način njihovog zadavanja (implicitno ili eksplicitno) i u odnosu na njihovu poziciju u upitu (između indeksa i između vrednosti u termu).

Eksplicitna upotreba logičkih operatora između indeksa je posebno izražena kod IR i baza naučnog sadržaja, dok se kod CRIS njihova upotreba svodi na minimum. Iz prethodnog zaključujemo da je u većini CRIS sistema nemoguće kreirati složene upite pretrage. Nemogućnost kreiranja složenih upita utiče na dostupnost podataka, što može umanjiti upotrebnu vrednost samih sistema. Kod implicitnog zadavanja logičkih operatora između indeksa u upotrebi je samo operator AND. Implicitno korišćenje ostalih operatora između indeksa nema praktični smisao što potvrđuje i statistika pojavljivanja ostalih operatora.

Što se tiče korišćenja logičkih operatora prilikom definisanja odnosa vrednosti u termu, implicitno zadavanje operatora ima blagu prednost u odnosu na eksplicitno zadavanje. Postoji više objašnjenja prethodne činjenice. Korisniku sistema je prihvatljivo (prirodno) da pri zadavanju više vrednosti u okviru terma sistem tretira njihove odnose kao logičko AND. Praksa je pokazala da korisnici radije definišu logičke odnose između termova istih indeksa nego da eksplicitno zadaju logičke odnose između više vrednosti u okviru jednog terma. Zaključuje se da eksplicitna upotreba logičkih operatora između terma nije od suštinske važnosti za sisteme pretraga. Svi analizirani sistemi osim *GoogleScholar* i *Microsoft Academic Search*, *Pure* i *Converis* podržavaju pretragu baziranu na logičkim upitima. Ovi sistemi podržavaju pretragu na osnovu posebnog mehanizma rangiranja rezultata. Mehanizam rangiranja je novi koncept pretrage pogodan za pretrage opšte namene, i ne može se u potpunosti iskoristiti u sistemima naučno-istraživačkih podataka obzirom da se u njima traže visok stepen podudaranja rezultata sa upitom pretrage (Google Scholar and academic libraries, 2012). Međutim u sistemima *GoogleScholar* i *EuroRIsNet+*

Observatory postoji mogućnost emulacije eksplicitnog zadavanja logičkih operatora AND i AND NOT koje se svodi na upotrebu specifičnih operatora "+" (include) i "-" (exclude). Navedeni operatori su unarni i oni obavezuju da se navedena vrednost terma mora naći u rezultatu "+" ili se ne sme naći u skupu rezultata "-". Klasični logički operator AND se emulira navođenjem "+" ispred vrednosti susednih termova/indeksa. Logički operator NOT dobijamo zadavanjem "+" ispred prvog i "-" ispred drugog operanda upita. Od svih logičkih operatora jedino je moguće eksplicitno korišćenje logičkog operatora OR. Implicitno zadavanje logičkih operatora u okviru čitavog upita ne postoji, obzirom da se sistem oslanja na specifični mehanizam rangiranja, a ne na logičke odnose u upitu.

Repositories	Logical operators											
	between index						between values of term					
	Explicit			Implicit			Explicit			Implicit		
	AND	OR	NOT	AND	OR	NOT	AND	OR	NOT	AND	OR	NOT
DSPACE ¹⁻⁹	+	+	+							+		
EPRINTS ¹⁰⁻¹⁵	+	+					+	+				
IEEE ¹⁶	+	+	+							+		
SpringerLink ¹⁷				+			+	+	+	+		
Emerald ¹⁸	+	+	+				+	+				
Scopus ¹⁹	+	+	+				+	+	+	+		
NARCIS ²⁰							+	+	+	+		
USDA CRIS ²¹	+	+	+	+			+	+	+	+		
E-CRIS SR ²²										+		
SICRIS ²³											+	
Cristin ²⁴				+							+	
HunCRIS ²⁵	+	+	+							+		
FRIS ²⁶				+			+	+	+	+		
OpenAIRE ²⁷				+			+	+		+		
GTR ²⁸							+	+	+		+	
EuroRISNet+ ²⁹							+		+		+	
Pure ³⁰⁻³²				+			+	+	+	+		
Converis ³³⁻³⁵				+			+	+	+	+		
DRIVER ³⁶				+			+	+		+		
GoogleScholar ³⁷				+			+	+	+	+		
Microsoft ³⁸	+		+				+	+	+	+		
Frequency	38%	33%	33%	43%	0%	0%	71%	67%	52%	81%	10%	0%

Tabela 2.6 Statistika korišćenja logičkih operatora

Tabela 2.7 opisuje specifične oblike pretraga kao što su približna pretraga, pretraga nedovršenih reči i rečenica, pretraga fraza i pretraga po udaljenosti. Na osnovu razmatranja prethodno navedenih sistema došlo se do zaključaka da je pretraga fraza i nedovršenih reči/rečenica nezaobilazna, dok bi se približna pretraga i pretraga po udaljenosti okarakterisale kao poželjne

funkcionalnosti sistema pretrage. Navedeni specifični oblici pretrage predstavljaju neizbežnu stavku kod kreiranja sofisticiranih upita. Funkcionalnost udaljene pretrage i približne pretrage se retko sreće u testiranim sistemima obzirom da nije toliko popularna kod korisnika početnog nivoa zbog njene nešto komplikovanije izvedbe.

Repositories	Types of search			
	phrase	unfinished word/sentence	distance	fuzzy
DSPACE ¹⁻⁹	+	+	+	
EPRINTS ¹⁰⁻¹⁵	+			
IEEE ¹⁶	+	+		
SpringerLink ¹⁷	+	+		+
Emerald ¹⁸	+	+	+	+
Scopus ¹⁹	+	+	+	
NARCIS ²⁰	+	+		
USDA CRIS ²¹	+	+	+	
E-CRIS SR ²²	+	+		
SICRIS ²³	+	+		
Cristin ²⁴	+	+		
HunCRIS ²⁵	+			
FRIS ²⁶	+	+	+	
OpenAIRE ²⁷	+			
GTR ²⁸	+	+	+	
EuroRISNet+ ²⁹	+			+
Pure ³⁰⁻³²	+	+	+	+
Converis ³³⁻³⁵	+	+		
DRIVER ³⁶	+			
GoogleScholar ³⁷	+	+	+	+
Microsoft ³⁸	+			+
Frequency	100%	71%	38%	29%

Tabela 2.7 Statistika mogućnosti pretrage

Za konstruisanje sistema pretrage sa visokim nivoom funkcionalnosti koji podržava sve navedene oblike zadavanja logičkih upita, kao i mogućnost kreiranja specifičnih oblika pretraga (frazu, približna pretraga, nedovršene reči/rečenice, pretraga po udaljenosti) zaključuju se da je neizbežno koristiti neku vrstu tekstualne pretrage

Iako se naša analiza nije bavila implementacionim aspektima, zbog uočenih različitosti u analiziranim sistemima, možemo zaključiti da su upotreba nekog standardizovanog upitnog jezika od suštinskog značaja za izgradnju

sistema visokih performansi (omogućavanje logičkih upite, kao i specifičnih oblika kao što su pretrage fraza, približna pretraga, pretraga nedovršenih reči/rečenica, i blizinska pretraga), koji je ujedno intuitivan i upotrebljiv.

3. Udaljeno pretraživanje podataka CRIS sistema

U ovom poglavlju ukratko su opisani relevantni koncepti SRU/W standarda kao što su CQL, Context Set i SRU/W profili pretrage. Na kraju je predstavljen profil za pretragu CRIS sistema koj nastao na osnovu rezultata analize iz poglavlja 2 i postojećih profila pretrage.

3.1. SRU/W standard

SRU (SRU- Search/Retrieve via URL, 2004) je protokol za pretraživanje i preuzimanje informacija sa veba projektovan kao naslednik poznatog protokola Z39.50. Nastao je kao rezultat internacionalne saradnje niza organizacija pod pokroviteljstvom Kongresne biblioteke ("Library of Congress," 1999), sa jasnim ciljem da se razvije standardizovani pristup koji omogućuje pretragu tekstualnog sadržaja. Protokol SRU podržava nove informacione tehnologije (WSDL, SOAP, HTTP i XML). SRU se u velikoj meri oslanja na apstraktne modele i funkcionalnost Z39.50 protokola, ali je umnogome pojednostavljen u odnosu na Z39.50 protokol. Trenutno su aktuelne tri verzije standarda SRU 1.1, 1.2 i 2.0. U okviru *Organization for the Advancement of Structured Information Standards* (OASIS | Advancing open standards for the information society, n.d.), na čelu sa radnom grupom koja se bavi pretragama u veb servisima (*Search Web Services Group* - SWSG) razvili su generičku definiciju (opis) protokola za komunikaciju između klijenta i servera prilikom pretrage u veb servisima. SWSG je pomenutom opisu dao naziv *Abstract Protocol Definition* (APD). APD treba da omogući interoperabilnost između različitih protokola za pretragu pružajući unifikovani okvir i terminologiju za opis samih protokola. APD definiše sledeće apstraktne modele:

1. **Data Model** (model podataka) je apstraktni model za opis podataka koji su predmet pretrage,
2. **Query Model** (model upita) je apstraktni model koji opisuje formiranje (konstrukciju) upita pretrage
3. **Processing Model** (model procesa pretrage) je apstraktni model koji sadrži informacije o načinu distribucije upita sa klijenta na server.

4. **Result Set Model** (model rezultata) je apstraktni model strukture rezultata pretrage.
5. **Diagnostic Model** (model za opis grešaka u pretrazi) je apstraktni model u kome se strukturno i nedvosmisleno opisuju greške koje mogu da nastanu prilikom pretrage.
6. **Description and Discovery Model** je apstraktni model koji omogućava opis funkcionalnosti i mogućnosti servisa pretrage.

Kod SRU protokola APD treba biti realizovan na sledeći način. *Processing Model* treba biti baziran na klijent-server paradigmi gde klijent i server komuniciraju pomoću *searchRetrieve* servisa u koji je inkorporiran (podržan) upitni jezik. Po ovom modelu klijent šalje *searchRetrieve* zahtev serveru, koji nakon obrade zahteva šalje *searchRetrieve* odgovor . U zahtevu koji šalje klijent nalazi se upit koji je formiran u skladu sa *Query Model*. Veći deo parametara iz odgovora je definisan u *Result Set Model*. O odgovoru servera se mogu naći i podaci o nastaloj grešci koja se iskazuje upotrebom *Diagnostics Model*. Data Model za CRIS UNS je detaljno opisan u poglavlju 3 Organizacija podataka unutar CRIS UNS sistema

Result Set Model je skup podataka (zapisa) koji nije eksplicitno definisan u SRU standardu. *Result Set Model* je konkretno u CRIS UNS opisan odgovarajućim SRU CRIS profilom. Nakon procesiranja zahteva server odgovara sa nizom (skupom) zapisa, koji odgovaraju upitu. Tako dobijeni set podataka uglavnom sadrži reference (id) ka konkretnim podacima u bazi podataka. Iako bi trebalo da se ovi skupovi budu nepromenljivi, oni se mogu interno (lokalno) modifikovati (npr. sortiranje skupa podataka po određenoj vrsti).

Result Set Model je skup podataka (zapisa) koji nije eksplicitno definisan u SRU standardu. *Result Set Model* je konkretno u CRIS UNS opisan odgovarajućim SRU CRIS profilom (odeljak 5.4.4). Nakon procesiranja zahteva server odgovara sa nizom (skupom) zapisa, koji odgovaraju upitu. Tako dobijeni set podataka uglavnom sadrži reference (*id*) ka konkretnim podacima u bazi podataka. Iako bi trebalo da se ovi skupovi budu nepromenljivi, oni se mogu interno (lokalno) modifikovati (npr. sortiranje skupa podataka po određenoj vrsti).

SRU unifikovano definiše načine zadavanja zahteva pretrage (svaki upit se integriše u okviru URL adrese), slanje zahteva serverima i pribavljanje rezultata pretrage u nekom od predefinisanih XML formata. Implementacija SRU interfejsa je vidno olakšana u poređenju sa Z39.50 protokolom, jer SRU ima manje potrebnih servisa (operacija) za realizaciju interfejsa od Z39.50 protokola. SRU koristi HTTP protokol i njegovu GET metodu (Morgan, 2004) za slanje komandi klijenta ka serveru. Komande se formiraju u obliku parametar/vrednost koji se uključuju u URL. Pomenuti parovi čine ulazne parametre za rad servera.

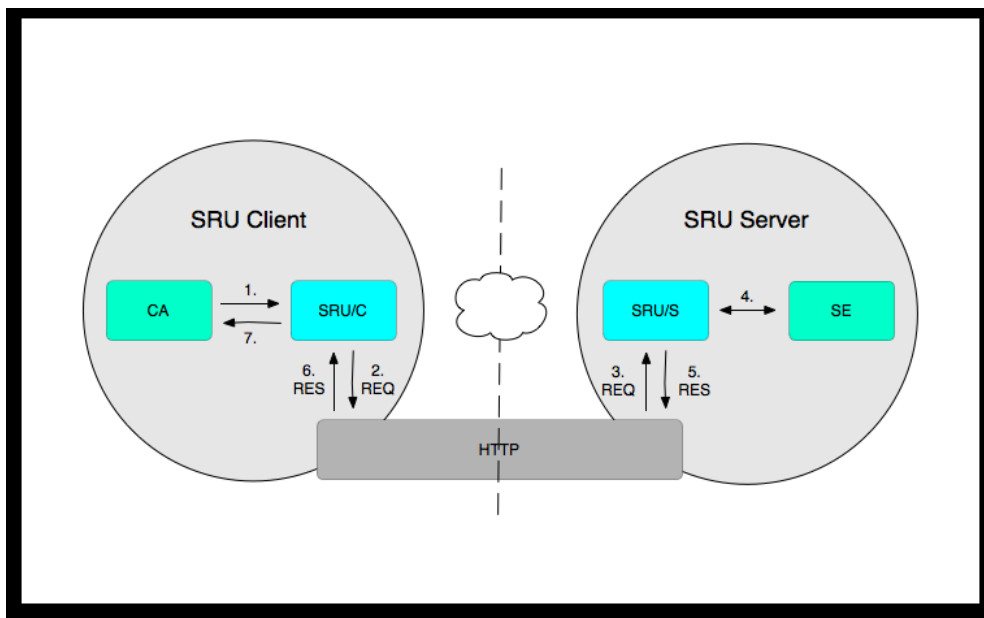
SRW (Search & Retrieve Web Service) je verzija protokola za pretraživanje i preuzimanje informacija koja preuzima principe funkcionisanja SRU, ali za razliku od SRU poruke pakuje u *SOAP Envelope* element (parovi parametar/vrednost su enkodirani u XML). Zbog prethodnog često se u naučnoj literaturi SRU i SRW protokoli nazivaju skraćeno SRU/W. Mada se u novije vreme u literaturi za SRW pojavio termin "SRU via SOAP".

SRU/W svoju pretragu zasniva na indeksima preko kojih se opisuju različiti resursi pretrage. SRU/W za definisanje sintakse upita koristi CQL (*Contextual Query Language*) dok se semantika upita opisuje kontekst setovima koji su organizovani u SRU/W profile. CQL (opisan *Query Model-om*) je opšti upitni jezik za pretragu resursa, a kontekst setovi su koncepti kojima se navode i objašnjavaju dozvoljeni entiteti CQL upita. Pojmovi CQL kontekst set i profil biće detaljnije objašnjeni u narednim odeljcima.

Servisi (operacije) SRU/W nastali su analizom upotrebe servisa Z39.50 protokola. Utvrđeno je da se od 11 dostupnih servisa Z39.50 protokola zapravo intenzivno koriste samo njih nekoliko. SRU/W koristi samo 3 servisa (*Explain, Scan, SearchRetrieve*) i deo su *Processing Model* APD sa koga nazivamo *Protocol Model*. Pomenuti model prikazan je na slici 3.1 i sadrži sledeće komponente:

- Klijentska aplikacija (CA)
- Modul SRU protokola na klijentu (SRU/C)
- Protokol nižeg nivoa (HTTP)
- Modul SRU protokola na serveru (SRU/S)
- interni pretraživač (SE)

Interakcija između pomenutih modula (komponenti) se odvija na sledeći način: Sve počinje tako što SRU/C prihvati zahtev koji je poslao CA, gde se formira *searchRetrieve* zahtev (REQ) koji se prosledi dalje kroz HTTP. Na serverskoj strani iz HTTP zahtev obrađuje SRU/S modul, koji se nakon toga obraća SE kako bi pronašao tražene podatke. Na kraju SRU/S formira *searchRetrieve* odgovor (RES) kroz protokol višeg nivoa. Odgovor poslat kroz HTTP (RES) obrađuje SRU/C kako bi dostavio rezultate klijentu koji je inicirao zahtev (CA).



Slika 3.1 Protokol model SRU/W standarda

3.1.1. SearchRetrive servis

SearchRetrive servis zadužen je za pretraživanje i dobavljanje podataka SRU/W servera. Zahtev za pretragu *SearchRetrieveRequest* i odgovor servera *SearchRetrieveResponse* predstavljaju najvažnije operacije pretrage podataka. Upit definisan CQL jezikom sastavni je deo *SearchRetrieveRequest* poruke i njegovom obradom server generiše odgovor. Složenost SRU/W CQL upita može varirati od jednostavne pretrage u kojoj se npr. zadaje slobodan tekst do kompleksnih logičkih upita sa ugnježenim podupitima i blizinskim kvalifikatorima. Ukoliko se u okviru zahteva pretrage nalaze mogućnosti koje server ne podržava, sever je dužan da pošalje dijagnostiku zašto se upit nije izvršio. Rezultati pretrage se smeštaju

u *SearchRetrieveResponse*. Format rezultata pretrage može biti slobodan tekst ili strukturirani podaci bazirani na XML rečnicima kao što su Dublin Core, MARCXML, MODS (*Metadata Object Description Schema*), CERIF XML, itd.

SearchRetrieveRequest

Poruka *SearchRetrieveRequest* sastoji se od više parametara od kojih je samo parametar koji definiše upit (*query*) obavezani dok su ostali parametri opcioni i ukoliko klijent ne definiše vrednosti za te parametre server će sam postaviti. Iako se podrazumeva da se u SRU/W upotrebljava CQL upitni jezik, na zahtev zajednice omogućeno je da korisnici ako to žele koriste sopstveni jezik za pretragu (neki koji nije CQL). Naime kako je SRU/W u skladu sa APD u kome postoji *Query Model* (deo za definisanje upitnog jezika) u *SearchRetrieveRequest* dodat je parametar *queryType*. Uloga *queryType* je da se u okviru njega navede stringovna oznaka imena protokola (npr. za CQL upitni jezik je „cql“). Serveri po SRU/W standardu često podržavaju i *queryType* sa vrednošću „searchTerms“. U slučaju da *queryType* ima pomenutu vrednost, server očekuje u *query* parametru da pojedinačne reči (termove) budu odvojene praznim mestima (npr. *cat hat rat*), gde server sam odlučuje kako će procesirati reči. Kada definišemo „specifičan“ tip upitnog jezika pored skraćenog imena koje mora biti jedinstveno mora se definisati i proizvoljan URI za jezik. U slučaju da se parametar *queryType* izostavi podrazumevani jezik servera po standardu je CQL.

Klijent prilikom slanja poruke može definisati i maksimalan broj zapisa koje želi da dobije od servera. Ovo je određeno parametrom *maximumRecord*. Server ne sme vratiti veći broj zapisa od onog broja koji je klijent specificirao. Klijent takođe definiše poziciju prvog zapisa (parametar *startRecord*) koji će biti vraćen iz skupa pronađenih zapisa. Ovaj parametar omogućava klijentu da preuzme deo po deo pronađenih zapisa, a ne ceo skup. U situaciji kada za određeni upit, bude formiran set zapisa, klijent može unapred da zada vremensko trajanje samog skupa popunjavanjem polja *resultSetTTL*. Vrednost ovog polja je u sekundama i ono predstavlja maksimalno „predloženo“ vreme postojanja tog skupa (seta) podataka na serveru. Parametrima *stylesheet* i *renderedBy* definišemo oblik prikaza odgovora sa servera. Ovo je pogotovu značajno kada želimo da odgovor koji

je XML prikazemo kao HTML. Tako u okviru parametra definišemo stylesheet konkretnu XSLT transformaciju XML odgovora. Ako je vrednost `renderedBy` parametra „client“, klijent u zahtevu definiše po kojoj XSLT transformaciji se prikazuje odgovor, a ako je vrednost „server“ tada server bira XSLT transformaciju XML odgovora.

Podaci koje klijent dobija od servera mogu biti u formi XML dokumenta što klijentu omogućava da upotrebom XSLT transformacija veoma jednostavno i sa minimalno obrade prikaže rezultate pretrage. Ukoliko klijent želi da podaci budu u formi XML dokumenta on može u zahtevu definisati parametar *recordSchema* kojim je definisao XML šemu po kojoj očekuje da mu server vrati zapise. Postoji skup registrovanih XML šema (SRU Record Schemas, 2013) koje su prihvaćene od Kongresne biblioteke i svaka šema ima naziv i URI identifikator koji je jedinstven. Na primer zapisi mogu biti u Dublin Core formatu i šema koja je definisana za ovaj format ima naziv `dc` i identifikator je `info:srw/schema/1/dc-v1.1`. Ukoliko server ne može da vrati zapise u specificiranoj šemi vratiće odgovarajuću poruku o grešci. Mada SRU standard dozvoljava da se zapisi u rezultatu ipak mogu biti klasifikovani po različitim šemama i to ukoliko je vrednost parametara *recordPacking* „unpacked“. U drugom slučaju vrednost ovog parametra može „packed“ i tada svi zapisi iz rezultata moraju biti klasifikovani istom šemom. Podrazumevana vrednost ovog parametra je „packed“.

Međutim, podaci ne moraju biti u formi XML dokumenta, već mogu biti dati u vidu slobodnog teksta. Da li će podaci biti vraćeni kao slobodan tekst ili kao XML dokument klijent specificira odgovarajućim vrednostima parametra *recordXMLEscaping* (moguće vrednosti su „xml“ ili „string“).

Sortiranje je funkcija SRU/W protokola ali i mogućnost u okviru CQL upita koji je deo standarda SRU/W. U slučajevima kada je funkcija sortiranja pozvana u SRU/W standardu i u CQL upitu server sam odlučuje kako će se izvršiti sortiranje. Sortiranje rezultata u SRU/W standardu klijent može zahtevati upotrebom parametra *sortKeys*. Parametar se sastoji od jednog ili više *sort* ključeva, gde svaki od njih ima set sledećih subparametara:

- *Path* – obavezan parametar koji se koristi prilikom sortiranja, a zadaje u obliku XPath izraza.

- *xsortSchema* – opcioni parametar koji predstavlja kratko ime šeme na koju se primenjuje XPath izraz. Ovo je pomoćna šema koja služi da bi se zapisi sortirali na specifičan način. Ukoliko se ovaj parametar izostavi koristi se šema definisana u parametru *recordSchema*.
- *Ascending* – logički parametar, čija je podrazumevana vrednost „true“. Opisuje u kom redosledu će zapisi biti sortirani.
- *caseSensitive* - logički parametar, čija je podrazumevana „true“. Ukazuje na to da li se vodi računa o malim i velikim slovima
- *missingValue* – koristi se u slučaju da *Path* vrednost ne postoji definisana u zapisu, npr. kada server treba da sortira vrednost po autoru, a u zapisu nedostaje informacija o autoru. Parametar može imati neke od sledećih vrednosti:
 - abort – Inicira se poruka o grešci u kojoj stoji da se sortiranje ne može izvršiti.
 - highValue – Server treba da sortira zapis kao da je vrednost u njemu najveća moguća (pronađena maksimalna vrednost).
 - lowValue – Server treba da sortira zapis kao da je vrednost u njemu najmanja moguća (pronađena minimalna vrednost).
 - omit – server treba da izbacii zapis prilikom prikaza sortiranih rezultata.
 - fixed value – Server treba da sortira zapis kao da vrednost ovog parametra postoji u zapisu.

Parametar *httpAccept* se u zahtevu koristi kako bi se označio željeni internet media format odgovora (mime type). Npr. ako klijent želi da odgovor bude u ATOM (“RFC 4287 - The Atom Syndication Format,” 2005) formatu tada vrednost ovog parametra treba biti „application/atom+xml“. Podrazumevana vrednost ovog parametra je „application/sru+xml“, što je i očekivano, obzirom da akcenat treba da bude na razmeni „sru xml“ poruka. U slučaju da server ne može da vrati poruku u odgovarajućem formatu, vraća se poruka o grešci. Klijent može da zahteva odgovor u više vrsta *mime type* formata, ali server ne mora biti u obavezi da odgovori na sve tražene tipove. Parametar *responseType* parametar definiše šemu odgovora i najčešće se koristi u kombinaciji sa parametrom *httpAccept*. Vrednost ovog parametra je string i

on nije obavezni parametar zahteva. Kada je definisan takav *httpAccept* za koji je unapred poznata šema (u okviru definicije samog standarda), kao što je slučaj sa „application/sru+xml“ tada se ovaj parametar i može izostaviti, međutim ako je upitanju zahtev za nekakav format za koji nije inicijalno po standardu propisana šema (npr. „application/atom+xml“), tada se u okviru *responseType* parametra mora eksplicitno naglasiti koja se šema želi koristiti za navedeni internet media format.

Po SRU/W standardu klijent se može omogućiti pretraga po različitim kategorijama „facet search“ (Tunkelang, 2009), gde kategoriju vezujemo za jedan ili više indeksa. Kako standard (protokol) inicijalno ne podržava multi-serversku pretragu uvođenje aspektne pretrage bi omogućilo da se izvrši pretraga iz više izvora (servera) a da klijent toga nije ni svestan. Klijent u zahtevu pretrage definiše upit i niz specifičnih (*facet*) parametra. Tako parametar *facetLimit* predstavlja maksimalni broj rezultata po aspektu (*facet* polju). Korišćenje parametra se može lako razumeti iz sledećih primera

- *faceLimit:=100*
postavlja limit na 100 zapisa za bilo koje polje (indeks)
- *faceLimit:=100:dc.subject*
postavlja limit 100 za polje *dc.subject* dok je za ostala polja ne postoje limiti
- *facetLimit=10,100:dc.subject,200:dc.title*
podrazumeva da je za *dc.subject* limit 100, za *dc.title* limit 200, dok je za sva ostala polja limit 10

U slučaju da ne želimo da podržavamo *facet*, parametar *facetLimit* treba imati vrednost 0. Iz primera zaključujemo da *facetLimit* može da ima 0 ili jedan nekvalifikovani (onaj koji važi za sva polja) i/ili 0 ili više kvalifikovanih limita. Limit su odvojeni zarezom i njihove vrednosti su celobrojne. Kod kvalifikovanih limita (100:dc.subject) između limita imena indeksa nalazi se karakter „:“. Vrednost limita za server znači sledeće:

- ako je vrednost pozitivna server „može“ (ali nije u obavezi) da obezbedi broj aspekata (*facet*).
- u slučaju da je vrednost nula neće biti rezultatnih *facet*.
- negativna vrednost podrazumeva beskonačan broj rezultatnih *facet*.

Zbog potencijalne dvosmislenosti preporuka za klijente je da ne koriste indekse koji u svom imenu imaju , i/ili :. Na primer, indeks bi teoretski mogao da se zove *dc.subject,200:dc.title*, gde bi smo imali limit od 100 zapisa i tada bi *facetLimit* imao sledeći oblik *facetLimit=100:dc.subject,200:dc.title*. Iz navedenog oblika *facetLimit* uočavamo da bi njega mogli da interpretiramo i kao limit od 100 za *dc.subject* indeks i limit od 200 za *dc.title*. Ako se vrednost *facetLimit* izostavi u potpunosti, server podrazumeva da mu je klijent poslao *facetLimit* sa vrednošću -1.

Definisanje ofseta u rezultatima definišemo je parametrom *facetStart*. Podrazumevana vrednost ovog parametra je 1 (krećemo od prvog elementa). *facetSort* parametar predstavlja sort specifikaciju za *facet* rezultate. *Sort* parametar ima sledeće komponente:

- *sortBy* - koji može imati sledeće vrednosti
 - *recordCount* (sortiranje po broju zapisa koji odgovaraju facet pretrazi)
 - *alphanumeric* (sortiranje po alfanumerčkim vrednostima rezultata)
- *order*
 - *ascending* – rastući redosled rezultata koji se sortiraju. (podrazumevana vrednost ako je *sortBy= alphanumeric*)
 - *descending* - opadajući redosled rezultata koji se sortiraju. (podrazumevana vrednost ako je *sortBy= recordCount*)
- *caseSensitivity* koristi se samo sa parametrom *alphanumeric*. Moguće vrednosti su *caseSensitive* i *caseInsensitive* (koja je i podrazumevana vrednost).

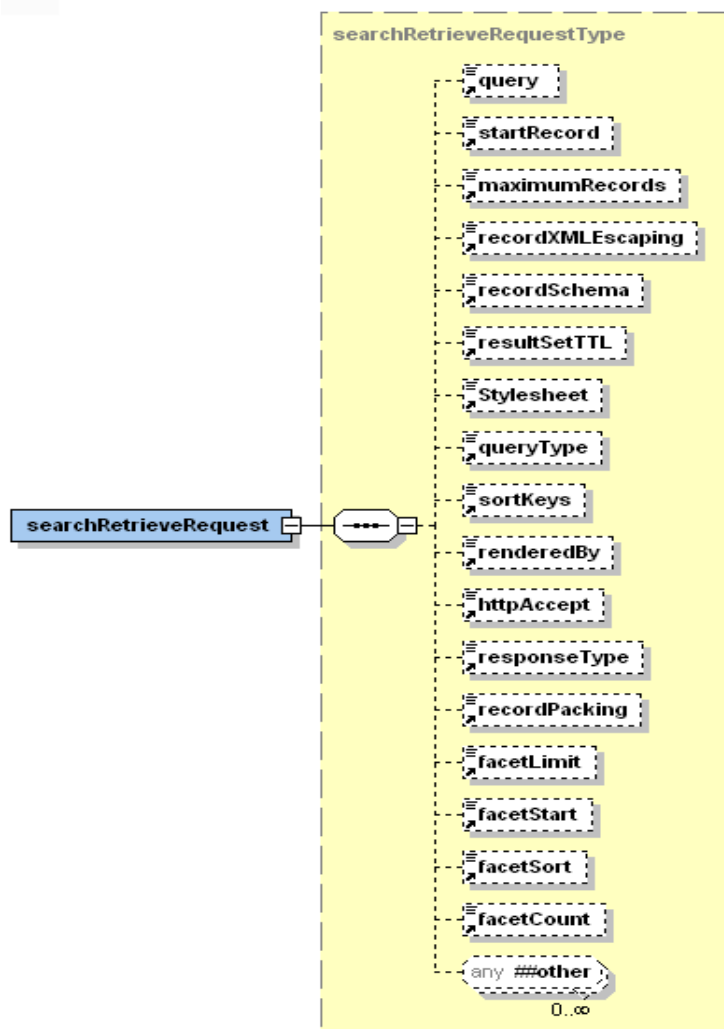
Komponente u *facetSort* parametru su odvojeni zarezom kao npr. *facetSort=alphanumeric,descending,caseSensitive*)

U *facet* pretrazi moguće je dobiti broj *facet* za neki karakteristični term, gde se tada ne smeju upotrebljavati drugi *facet* parametri. Primer upotrebe parametra *facetCount* je sledeći *facetCount:dc.subject=history*

Kako u zahtevu tako i u odgovoru moguće je uvrstiti i niz dodatnih parametara koji nisu inicijalno uvršteni u definiciju SRU/W standarda. Treba imati u vidu da dodatni parametri zahteva ne mogu sugerisati promenu uobičajenog funkcionisanja SRU servera (npr. ne možemo dodatnim parametrima zahtevati da server čita upite po SQL standardu), već samo da tražimo dodatne informacije od servera koje inicijalno nisu podržane SRU/W standardom. Ne postoji poseban parametar u kome se dodaju dodatni parametri niti oni moraju biti definisani u skladu sa nekom predefinisanim šemom. Jedini uslov je da parametri moraju da imaju definisanu šemu koju poznaje server kako bi mogao da dopremi tražene podatke. Moguće je da se parametri i ne nalaze u istom prostoru imena i šemi (čak je i poželjno kako bi se lakše razlikovali) ali se to mora i eksplicitno naglasiti kao na Listingu 3.1. Na slici 3.2 predstavljen je grafički prikaz XML šeme generisan u alatu XMLSpy (Altova XMLSpy XML Editor, 2014).

```
<theo:onSearchFail  
xmlns:theo="info:srw/extension/4/searchextensions">  
scan  
< /theo:onSearchFail>  
<rob:relevancyAlgorithmxmlns:rob="info:srw/extension/2/relevancy">  
CORI< /rob:relevancyAlgorithm>
```

Listing 3.1 Parametri iz različitih prostora imena



Slika 3.2 SearchRetrieveRequest

SearchRetrieveResponse

Server po prijemu *SearchRetrieveRequest* poruke procesira upit koji je dobio od klijenta i vraća *SearchRetrieveResponse* poruku. Broj pronađenih zapisa u bazi server vraća u parametru *numberOfRecords*.

Server može implementirati opciju čuvanja pronađenih rezultata u specifičnim skupovima podataka *resultSet*, i tada se svakom skupu rezultata mora dodeliti jedinstveno ime *resultSetId*. Skup rezultata nastaje prilikom izvršavanja upita i predstavlja svojevrsan način za keširanje rezultata. Sa stanovišta klijenta *resultSet* je skup zapisa referenciranih rednim brojevima

počevši od 1. Kada klijent jednom dobije vrednost *resultSetId*, on ne mora ponoviti isti CQL upit već samo koristi dobijeni *resultSetId* kako bi došao do istog skupa podataka. Treba obrati pažnju da skupovi zapisa nisu inicijalno podržani parametrom u SRU/W standardu već je to deo CQL upitnog jezika (zbog toga treba biti obazriv u slučaju da klijent ne koristi CQL upitni jezik). Zapisi mogu biti obrisani ili postati nedostupni u bazi a da zapisi u *resultSet* i dalje postoje. Ako klijent ipak zatraži takav zapis, server je u obavezi da prikaže poruku o grešci na mestu tog zapisa. Npr. zapis na poziciji 2 u skupu zapisa je obrisani, a klijent zahteva zapise od pozicije 1 do 3. Neophodno je napomenuti da se svaki novi upit formira novi *resultSet* (čak i kada već postoji *resultSet* nastao na osnovu istog upita).

Server može ranije da obriše taj skup ali ne sme nikako da dozvoli da taj skup živi i nakon vremena koje je upisano u *resultSetTTL*. Server može prihvatiti vreme zadato u istoimenom parametru u *SearchRetrieveRequest* ili zadati novu proizvoljnu vrednost u slučaju da proceni da taj skup podataka ne treba da bude toliko dugo dostupan.

Ukoliko poslednji zapis u *SearchRetrieveResponse* nije i poslednji u *resultSet*, odgovor sadrži i element *nextRecordPosition* čija je vrednost rednog broja zapisa koji sledi nakon poslednjeg u generisanom odgovoru (Listing 3.2). Ukoliko su vraćeni svi zapisi iz *resultSet* ovaj parametar mora biti izostavljen.

```

<searchRetrieveResponse>
  <numberOfRecords>10</numberOfRecords>
  <resultSetId>resultA</resultSetId>
  <resultSetTTL>180</resultSetTTL>
  <records>
    <record>
      record 1 ...
    </record>
    <record>
      record 2 ....
    </record>
  </records>
  <nextRecordPosition>3</nextRecordPosition>
</searchRetrieveResponse>

```

Listing 3.2 Generisani searchRetrieveResponse

Server ima mogućnost da proizvede procenu tačnosti broja rezultata parametra *numberOfRecords*), što bi bilo posebno značajno ako se rezultati dopremaju sa različitih fizičkih lokacija (servera). Naime, procena se iskazuje vrednošću parametra *resultCountPrecision* u vidu URI koji identifikuje specifičan term iz kontrolisanog rečnika. Postoji rečnik koji je propisan u okviru SRU/W standarda i sadrži sledeće vrednosti:

- exact – Server garantuje da je vrednost iz *numberOfRecords* tačna.
info:srw/vocabulary/resultCountPrecision/1/exact
- unknown – Server nema informaciju o broju rezultata i ne želi da iskazuje nikakvu procenu o broju rezultata.
info:srw/vocabulary/resultCountPrecision/1/unknown
- estimate - Server nema informaciju o broju rezultata ali daje procenu.
info:srw/vocabulary/resultCountPrecision/1/estimate
- maximum – Server daje procenu da je ovo maksimalna vrednost broja zapisa koja bi se mogla naći u skupu rezultata.
info:srw/vocabulary/resultCountPrecision/1/maximum

- minimum – Server ne zna ukupan broj rezultata, ali je ovo minimum koji može da se očekuje u skupu rezultata.
info:srw/vocabulary/resultCountPrecision/1/minimum
- current – Sever odgovara ovom vrednošću u slučaju kada proceni da je to trenutni broj rezultata koji se može proširivati.
info:srw/vocabulary/resultCountPrecision/1/current

Kao vrednosti *resultCountPrecision* parametra može se koristiti term ili URI, gde je preporuka da se koristi URI. U URI se pored naziva nalazi broj 1 kao oznaka autorizovanog tela (*SRU Maintenance Agency*) koje propisuje URI. Sva registrovana tela navedena su na <http://www.loc.gov/standards/sruBob/resources/infoURI.html#authoritystring>. URI oblika „info-sru“ nije obavezujući oblik URI, već više nekakva konvencija koja je usvojena na nivou SRU/W standardu. Za *resultCountPrecision* element moguće je i konstruisati sopstvene rečnike sa adekvatnim terminima.

Osnovni i najznačajniji parametar *SearchRetrieveResponse* poruke je parametar records koji predstavlja skup pojedinačnih zapisa *record*. Svaki zapis sadrži obaveznu informaciju o XML šemi po kojoj je zapis formatiran (*recordSchema*) čija je vrednost kompletan URI. Obavezni parametar je i *recordXMLEscaping* koji ukazuje da li je zapis enkodovan kao string ili kao XML. Najvažniji element u zapisu, koji je naravno obavezan, predstavlja sam sadržaj zapisa (*recordData*) po nekoj šemi. Zapis može sadržati i poziciju tog zapisa u skupu pronađenih pogodaka (*resultSet*). Parametar *recordPosition* označava poziciju zapisa u *resultSet*. U elementu zapisa (*record*) mogu se transferovati i dodatne informacije o zapisu kao sadržaj podelementa *extraRecordData*. Npr. u ovom elementu se može naći podatak o relevantnosti samog zapisa:

```
<extraRecordData>
<rel:rank xmlns:rel="xmlns:rob="info:srw/extension/2/relevancy">
0.965</rel:rank>
</extraRecordData
```

Listing 3.3 Generisani searchRetrieveResponse

Parametar *extraRecordData* nikako ne treba mešati sa *extraResponseData* koji je deo *SearchRetrieveResponse* poruke a ne pojedinačnog zapisa. Upotrebom *extraResponseData* se omogućuje da se svi dodatni parametri odgovora smeste na jednom mestu bez potrebe da menjanja inicijalne

strukture *SearchRetrieveRequest* poruke. Sama Kongresna biblioteka je uvrstila neke parametre kao deo standarda nove verzije SRU/W, a koji su se u ranijim verzijama koristili samo kao dodatni. U slučaju kada se primenjuje mehanizam dopremanja dodatnih informacija važe sledeća pravila:

- **Ukoliko server ne prepoznaje dodatne podatke iz *SearchRetrieveRequest* poruke, parametri se prosto mogu zanemariti.** Ovo je vrlo slična situacija kao i kada server ne podržava mogućnost definisanu u nekom od opcionih parametara. Pored mogućnosti da server ignoriše parametre, može kao odgovor generisati i poruku o grešci. (odjeljak 3.1.4).
- **Principi rada i funkcionisanja servera ne mogu biti promenjeni upotrebom dodatnih informacija/parametara.** Kao što je navedeno u zahtevu ne možemo tražiti od servera dodatnim parametrom da promeni upitni jezik.
- *extraResponseData* element se **može naći** u *SearchRetrieveResponse* odgovoru samo ga ako ga klijent inicirao kroz *SearchRetrieveRequest*.

Siromašni klijenti poput veb čitača koji rukuju samo stilovima za prikaz (*stylesheet*), zasigurno nemaju mogućnost da i skladište upit koji u trenutku generišu. Iz tog razloga server bi trebao da bude u mogućnosti da u određenom elementu odgovora (*echoedSearchRetrieveRequest*) dostavi i upit koji je klijent poslao. Ne postoji eksplicitan element u *SearchRetrieveRequest* kojom bi se ova funkcionalnost zahtevala od servera, već server sam odlučuje kada i kome će vratiti i zahtev. Element *echoedSearchRetrieveRequest* (Listing 3.4) sadrži podelemente koji reprezentuju korespondentne parametre definisane u *SearchRetrieveRequest*. Za ovu reprezentaciju je zadužena XCQL (XCQL XML šema, 2004), što zapravo je XML predstava CQL.

```

<echoedSearchRetrieveRequest>
  <query>dc.title = dinosaur</query>
  <recordSchema>mods</recordSchema>
  <xQuery>
    <searchClause xmlns="info:srw/xmlns/1/xcql-2-0-v1">
      <index>dc.title</index>
      <relation>
        <value>=</value>
      </relation>
      <term>dinosaur</term>
    </searchClause>
  </xQuery>
  <baseUrl>http://z3950.loc.gov:7090/voyager</baseUrl>
</echoedSearchRetrieveRequest>

```

Listing 3.4 Generisani echoedSearchRetrieveRequest

Korenski element XCQL predstave je *xQuery* i može se naći u SearchRetrieve i Scan operaciji. XCQL reprezentacija donosi sledeće benefite:

- Klijent može da koristi XSLT ili neki drugi način za manipulaciju XML kako bi modifikovao upit bez potrebe da ima nekakav vid parsera za CQL.
- Server može vratiti dodatne informacije o nekoj specifičnoj klauzuli u upitu.

Sever može poslati i svoju URL reprezentaciju u elementu baseURL. Ovakva informacija bi omogućila klijentima rekonstrukciju upita izvršavanjem jednostavne konkatencije.

Po SRU/W standardu server može da generiše rezultate aspektne pretrage (facet) u elementu facetedResults. Rezultati moraju biti u korespondenciji sa facet u definisanim tako da :

- ukoliko nema parametara u zahtevu, server nije u obavezi da generiše rezultate za ovu pretragu ili generiše rezultate po svom nađenju.
- ukoliko je facet parametar u zahtevu 0 (facetLimit=0) tada server ne sme da generiše rezultate.
- u ostalim slučajevima server treba da pokuša da generiše rezultate u skladu sa parametrima navedenim u zahtevu

Jedna od osnovnih prednosti *facet* pretrage u odnosu na klasičnu je pretraživanje više izvora podataka (servera) i nakon toga njihova

konsolidacija u jedinstveni skup zapisa. U Listingu 3.5 je prikazan primer *facet* pretrage sa rezultatima iz Library of Congress Catalog i MELVY baza.

```

<facetedResults>
<datasource>
  <datasourceDisplayLabel>LOC Catalog</datasourceDisplayLabel>
  <datasourceDescription></datasourceDescription>
  <baseURL></baseURL>
  <facets>
    <facet>
      <facetDisplayLabel>Library of Congress
Catalog</facetDisplayLabel>
      <facetDescription>Facet counts for
subject</facetDescription>
      <index>subject</index>
      <relation></relation>
      <terms>
        <term>
          <actualTerm>birds</actualTerm>
          <query></query>
          <count>15</count>
        </term>
        <term>
          <actualTerm>nuthatches</actualTerm>
          <query></query>
          <count>12</count>
        </term>
      </terms>
    </facet>
    ...
  <datasource>
    <datasourceDisplayLabel>MELVYL</datasourceDisplayLabel>
    <datasourceDescription></datasourceDescription>
    <baseURL></baseURL>
    <facets>
      <facet>
        <facetDisplayLabel>MELVYL
Catalog</facetDisplayLabel>
        <facetDescription>Facet counts for
subject</facetDescription>
        <index>subject</index>
        <relation></relation>
        <terms>
          <term>
            <actualTerm>nuthatches</actualTerm>
            <count>18</count>
          </term>
          <term>
            <actualTerm>Sittacarolinensis</actualTerm>
            <count>4</count>
          </term>
        </terms>
      </facet>
    ..
  ..

```

Listing 3.5 Facet pretraga iz više izvora

Kada se klijent serveru obrati sa komplikovanijim upitom moguća je i njegova različita interpretacija, u zavisnosti od toga kako server tumači vezu između samih podupita. Tako da ako uzmemo za primer upit:

```
title=cat and subject=dog or author=frog
```

Podupiti koji nastaju procesiranjem upita su sledeći:

- title=cat
- subject=dog
- author=frog
- title=cat and subject=dog
- title=cat or author=frog
- subject=dog or author=frog

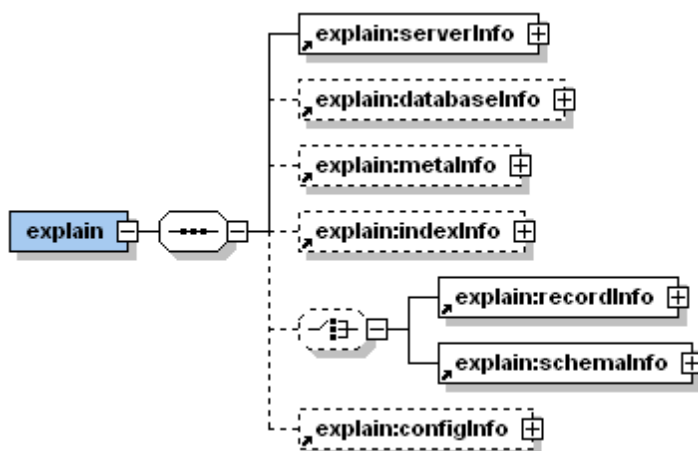
Analiza upita zapravo predstavlja analizu njegovih podupita.. Standardna procedura obrade upita podrazumeva njegovo procesiranje sa leva na desno, tako bi prvo bio obrađen podupit title=cat, nakon toga bi mogao da sledi upit title=cat and subject=dog. Na kraju bi se izvršila disjunkcija (*OR*) između prethodnog upita (title=cat and subject=dog) i upita "author=frog". Neki serveri bi polazni upit mogli tretirati i kao "title=cat and (subject=dog or author=frog)", gde imamo situaciju da ako "title=cat" rezultuje sa 0 pogodaka server može prekinuti sa obradom upita. (nema smisla obrađivati dalje upit ako je jedan operand kod operaciji konjukcija prazan skup). Tako da analiza upita (njegova dekompozicija) može biti vrlo korisna u cilju redefinisanja upita na osnovu informacija dobijenih obradom njegovih podupita.

U slučaju da server ne može da odgovori na klijentov zahtev, poruka koju server šalje će sadržati parametar *diagnostics* koji služi da se u njega smeste poruke o greškama. Postoji lista definisanih (Diagnostics List (SRU: Search/Retrieval via URL -- SRU), 2012) grešaka i za svaku grešku postoji identifikator greške, opis greške i dodatno objašnjenje koje se može prikazati krajnjem korisniku.

3.1.2. Explain Servis

Uloga *Explain* servisa je da SRU/W klijentu pruži informacije o mogućnostima samog SRU/W servera (informacije o bazi podataka, o dostupnim indeksima i o funkcionisanju samog servera), jer server po specifikaciji nije dužan da implementira sve detalje SRU/W protokola. Tako SRU/W serveri ne moraju da implementiraju sve mogućnosti CQL, kao ni sve elemente korišćenih kontekst setova. U ovom servisu se razmenjuju dva tipa poruka *ExplainRequest* i *ExplainResponse*. *ExplainRequest* poruka koju klijent šalje je veoma jednostavna i pored obaveznog parametra koji označava verziju protokla ne mora imati ni jedan drugi parametar. Odgovor servera u vidu *ExplainResponse* poruke sadrži obavezan parametar *record* koji predstavlja XML dokument u skladu sa *ZeeRex* specifikacijom (Z39.50 Explain, Explained and Re-Engineered in XML, n.d.). Odgovor servera se često naziva i „*Explain document*“, koji predstavlja specifikacioni dokument u kome se navode informacije kako pristupiti, pretražiti i prikazati informacije SRU/W servera. Informacije u *Explain document* se odnose na: podržan/e upitni/e jezik/e, kontekst setove, kustomizaciju rezultata (sortiranje, definisanje šeme, enkodiranje, kodne strane, formati).

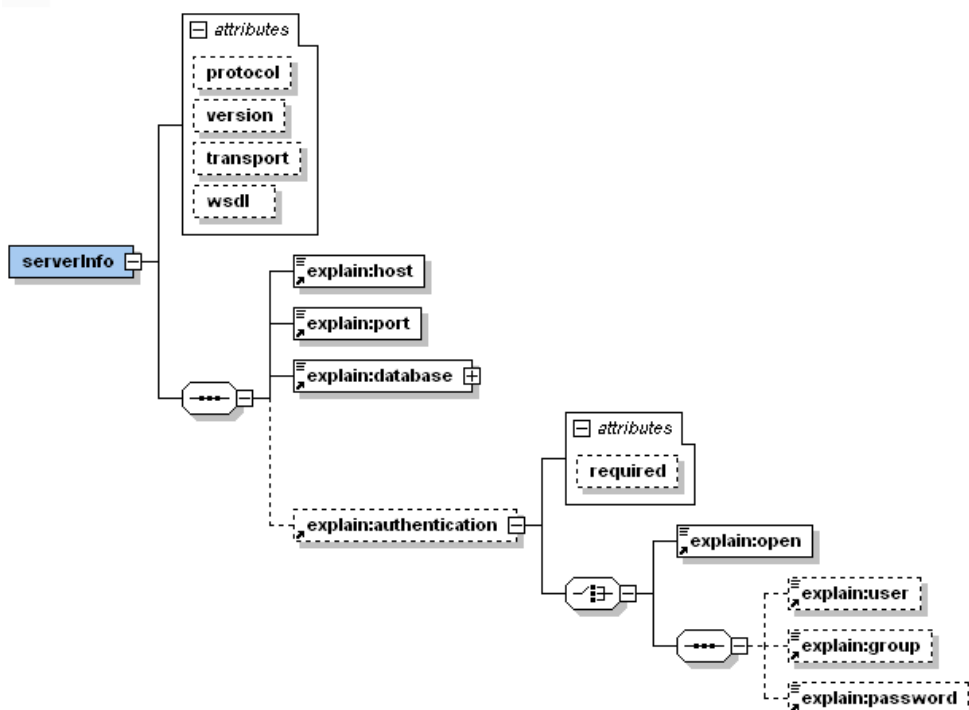
ZeeRex (Z39.50 Explain, Explained and Re-Engineered in XML) specifikacija služi za opis funkcionalnosti servera koji podržava SRU/W standard i klijent može koristiti dobijenu specifikaciju servera za samokonfigurisanje korisničkog interfejsa tj. svoje klijentske aplikacije. Na taj način će korisnik biti sprečen da koristi neke od elemenata standarda koje posmatrani server ne podržava. Prvobitna namena *ZeeRex* specifikacije bila je opis servera koji koriste Z39.50 standard, međutim ubrzo je uočeno da se ova specifikacija može koristiti i za opis drugih protokola za preuzimanje i pretraživanje informacija. Korenski element *explain* definisan *ZeeRex* XML šemom dat je na slici **3.3**



Slika 3.3 Korenski element explain

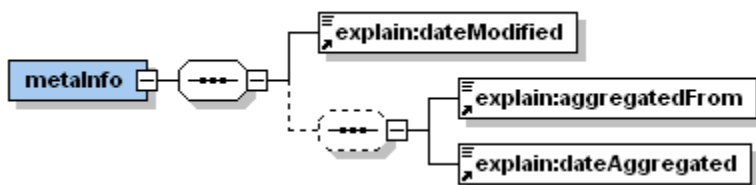
Element *explain* ima dva neobavezna atributa *authoritative* i *id*. Atribut *authoritative* predstavlja opis autorizovanog subjekta kome pripada specifikacioni dokument. Atribut *id* je identifikator osobe ili institucije koja je kreirala ovaj dokument. Korenski element *explain* se sastoji od šest podelemenata: *serverInfo*, *databaseInfo*, *metaInfo*, *indexInfo*, *searchInfo*, *recordInfo* ili *schemaInfo* što zavisi od protokola koji se opisuje i *configInfo*.

Element *serverInfo* sadrži osnovne informacije neophodne da se uspostavi mrežna komunikacija sa serverom. U atributima ovog elementa (*protocol*, *version*, *transport*, *wsdl*) navedene su informacije o protokolu koji server koristi. Predefinisane vrednosti atributa *protocol* su Z39.50, SRU, SRW i SRU/W. Ukoliko je izabrana vrednost SRU/W to znači da su obe verzije servisa dostupne na istoj URL adresi). Aktuelna verziju protokola upisana je atributu *version*, dok se putanja do WSDL fajla koji opisuje server nalazi u atributu *wsdl*. U elementu *serverInfo* nalaze se obavezni podelementi za definisanje adrese servera, porta i naziva baze koja se pretražuje. U okviru elementa naziva baze (*database*) postoje opcioni atributi u kojima se može zadati ukupan broj zapisa u bazi (*numRecs*) i vreme poslednje modifikacije baze po ISO 8601 (YYYYMMDDHHMMSS). Podaci potrebni za autentifikaciju klijenta se nalaze u podelementu *authentication*. Sama autentifikacija se uključuje postavljanjem atribut *required* elementa *authentication*. U slučaju da server treba bude javno dostupan bez ograničenja neophodno je postaviti vrednost *open* u *authentication* elementu. Grafički prikaz ovog elementa dat je na slici 3.4.



Slika 3.4 Element *serverInfo*

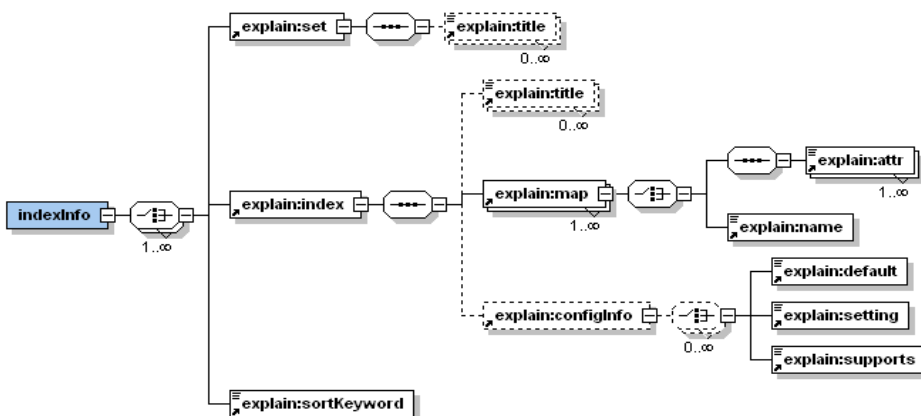
Element *metaInfo* služi za opis samog Explain dokumenta (ZeeRex XML dokumenta). Ovaj element sadrži informacije o datumu kada je dokument kreiran ili modifikovan i na koji način je dobijen sadržaj. Sadržaj dokumenta se može dobiti pretraživanjem posebnih *Explain* baza podataka koje sadrže samo ZeeRex XML dokumente i služe za pretraživanje servera koji podržavaju Z39.50 i/ili SRU/W protokole. Grafički prikaz ovog elementa dat je na slici 3.5.



Slika 3.5 Element *metaInfo*

Element *indexInfo* (slika 3.6) sadrži informacije o prefiksima koji se mogu koristiti u pretraživanju. U ovom elementu bi korisnik mogao da pronađe informacije kako da formira upit. Kada je u pitanju SRU/W server potrebno

je definisati (setovati) *ContextSet* koji server podržava i naravno koje sve prefikse podržava iz posmatranog *ContextSet*-a. Nakon toga u podelementu *index* moguće je navesti i informacije o pojedinačnim indeksima servera. U atributima pomenutog elementa (*id*, *search*, *scan*, *sort*) moguće je definisati: jedinstveni identifikator indeksa, da li je indeks moguće koristiti u servisu *Search/Retrieve* i/ili u *Scan* servisu, kao i da li je omogućeno sortiranje podataka za taj indeks. Sam element *index* sadrži sledeće podelemente: *title* (ime indeksa na serveru koje može biti i višejezično upotrebom njegovog atributa *lang*), *map* (definiše alternativna imena indeksa u drugim setovima) i *configinfo* (predefinisane vrednosti za određeni element).



Slika 3.6 Element *indexInfo*

Element *searchInfo* pruža informacije o načinima za formulaciju same pretrage. Tako se u podelementima (*queryExample*, *queryType*) respektivno nalaze informacije o karakterističnim primerima i tipu upita (koji je upitni jezik podržan, najčešće je to CQL). Element *queryType* ima niz atributa koji se odnose na URI identifikator tipa upita (definicija upitnog jezika), kao i nivo (*supportLevel*) podržane kompleksnosti upitnog jezika. Podelement *facet* sadrži informacije o pretrazi po aspektima (*facet search*), gde se u posebnim atributima ovog elementa smeštaju informacije o aspektima. Tako imamo ime aspekta, opis, dodatne informacije, oblik/tip sortiranja. Enkodovanje unetih karaktera definisano je podelementom *inputEncoding*.

Element *recordInfo* ukazuje na tipove zapisa koje server produkuje kao rezultate. Podelementi se odnose na informacije o XML formatu zapisa.

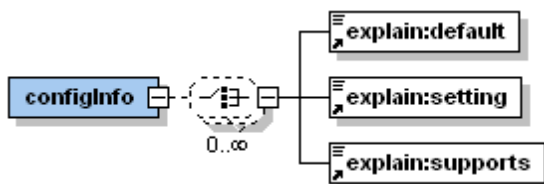
Element *schemaInfo* (slika 3.7) sadrži informacije o XML šemama po kojima mogu biti formatirani zapisi koje će server slati klijentu kao odgovor

na upit. SRU/W server, na primer, može definisati da podržava razmenu zapisa po DublinCore , MARCXML i UNIMARC XML šemi (UNIMARC XML šema, 2004)dok klijent u svom zahtevu može specificirati po kojoj konkretno šemi želi da dobija zapise. U podelementima su navedeni podaci koji se odnose na samu šemu (identifikator, ime, fizička lokacija, dodatni podaci).



Slika 3.7 Element *shemaInfo*

Element *configInfo* definiše konfiguraciju servera, odnosno koji su sve koncepti definisani standardom podržani. Na primer, SRU/W- server može podržavati samo *SearchRetrieve* servis, dok ne mora podržavati *Scan* servis. Slično, može podržavati samo upite sa jednim prefiksom, a da ne podržava upite sa logičkim operatorima, ili da ne podržava imenovane skupove rezultata itd. Tako se u podelementu *default* opisuju elementi standarda koji se smatraju podrazumevanim ukoliko drugačije nije definisano. (npr, klijent kada ne navede *maximumRecords* koliko će biti maksimalni broj zapisa u odgovoru). U elementu zatim *setting* se navode određena podešavanja odnosno vrednosti se ne mogu zameniti vrednostima koje klijent predloži (npr. iako u zahtevu klijent navede broj za *maximumRecords*, vrednost za *maximumRecords* će biti ona koja stoji u elementu *setting*). Podelementom *supports* dat je opis elemenata koji su podržani od strane servera (npr. u slučaju da postoji pretraga po udaljenosti u ovom elementu će to biti i eksplicitno navedeno). Grafički prikaz ovog elementa dat je na slici 3.8.



Slika 3.8 Element *configInfo*

U Element *diagnosticSets* se nalaze informacije o tome da li server podržava dijagnostifikaciju greške. Podelementi *setName*, *URI*, *URL* se respektivno odnose na ime skupa grešaka, definicije skupa grešaka kao i mesta gde se određeni skup može preuzeti.

3.1.3. Scan servis

Scan servis omogućava izlistavanje svih termova vezanih za određeni indeks pretrage. *Scan* servis se obično koristi u kombinaciji sa *SearchRetrieve* servisom kako bi se izbeglo nepotrebno pretraživanje i dobijanje prevelikog skupa podataka. U praksi bi bilo dobro da korisnik prvo pozove servis *Scan* za određeni indeks, dobije listu mogućih vrednosti, broj pogodaka za taj indeks, i zatim pozove servis *SearchRetrieve* sa upitom u kome je definisana jedna od pronađenih vrednosti.

ScanRequest poruka koju klijent šalje serveru sastoji se od obaveznog parametra *scanClause* koji sadrži informaciju o indeksu koji se pretražuje. Vrednost jedinog obaveznog parametra *scanClause* dat je u obliku CQL upita, gde je upit iskazan indeksom mogućom relacijom i termom. Nije dozvoljeno koristiti relacije koje određuju opseg kao što su: $<$, $>=$, *within*. Term u klauzuli će predstavljati početnu tačku (početni term) u uređenoj listi termova rezultata. Npr. u upitu „*dc.title==dragan*“, kao rezultat možemo dobiti termine: „dragan“, „draga“, „dragana“... gde će početni term biti „dragan“. Pored ovog parametra klijent može definisati i parametar *maximumTerms* koji označava maksimalan broj vrednosti koje klijent može prihvatiti, koja mora biti pozitivna celobrojna vrednost. Ukoliko se ovaj parametar ne navede, server definiše maksimalni broj termova u rezultatu. U kombinaciji sa *maximumTerms* često se koristi i parametar *responsePosition*. Klijent navodi ovaj parametar u *Scan* zahtevu, kako bi definisao željenu poziciju početnog terma u listi rezultatnih termova. Podrazumevanu vrednost, ako se ona eksplicitno ne navede, definiše server. Početni term ne mora biti deo indeksa, te se stoga mora uvesti i termin najbliži term. Najbliži term predstavlja početni term ukoliko je on deo indeksa, a ako nije vrednost deo indeksa, poziciju gde bi početni term bio da je u okviru indeksa. Vrednosti parametara početni term, maksimalni broj termova (*maximumTerms*) i najbliži term su međusobno zavisne. Uzećemo da su P i M respektivno vrednosti parametara *responsePosition* i *maximumTerms*. Početni term se definiše u skladu sa sledećim uslovima:

- Ako je $P \leq 0$. Početni term u listi termova se nalazi Q mesta nakon najbližeg, gde je $Q = |P| + 1$, npr. ako je $P = -1$ tada se početni term u listi može naći na udaljenosti od dve pozicije nakon najbližeg terma.

- Ako je $P > 0$, početni term se u listi termova može naći za Q termova pre najbližeg terma, gde je $Q = P - 1$ (Npr. ukoliko je $P = 3$, u ovom slučaju se početni term nalazi dva terma ispred najbližeg terma)
- Ukoliko je $P > M$ tada se početni term ne nalazi u listi, jer svi članovi liste prethode početnom termu.
- Stvarni broj termina u listi ne treba da prelazi M , ali može biti manji, na primer, ako se dostigne kraj liste termina.

U nastavku je dat primer koji jasno opisuje korišćenje termina najbliži term i početni term (Primer 1).

Primer 1

Pretpostvke:

- Indeks se sastoji od termova u sledećem redosledu: A,B,C,D,E,F,G,H
- Najbliži term je D

Karakteristični slučajevi

- najbliži term=-1, lista rezultujućih termova bila bi F,G,H
- najbliži term=0, lista rezultujućih termova bila bi E,F,G
- najbliži term=1, lista rezultujućih termova bila bi D,E,F
- najbliži term=4, lista rezultujućih termova bila bi A,B,C

U oviru *ScanRequest* postoji opcioni parametar *extraRequestData* u kome klijent može da definiše dodatne informacije koje server treba da obradi.

Server vraća poruku *ScanResponse* koja sadrži listu vrednosti koje su pronađene u datom indeksu što je predstavljeno parametrom *terms*. Elementi koji se još mogu naći u odgovoru *Scan* servisa su *diagnostics*, *extraResponseData*, *echoedScanRequest*. Element *diagnostics* predstavlja tip i opis greške koja se dogodila. Server u *extraResponseData* definiše dodatne informacije odgovora i to najčešće ako je to traženo u zahtevu parametrom *extraRequestData*. U parametru *echoedScanRequest* server smešta sve informacije koje opisuju zahtev klijenta. (verziju SRU, CQL upit, maksimalan broj termova ...).

Parametar *terms* se sastoji od sekvence parametara *term*, pri čemu svaki parametar *term* opisuje pojedinačnu vrednost koja je pronađena u datom indeksu. Parametar *term* sadrži u obaveznom elementu *value* informaciju o vrednosti (egzaktno kao u indeksu) terma i poziciji (*whereInList*) te vrednosti u listi ostalih vrednosti za izabrani indeks. Vrednosti za poziciju su predefinisane i pozicija može biti na početku liste (*first*), na kraju liste (*last*), u sredini (*inner*) i na kraju slučaj kada postoji samo jedan term u indeksu (*only*). Parametar *term* sadrži i informaciju o broju pogodaka (*numberOfRecords*) u kojima se posmatrana vrednost nalazi za zadati indeks. U parametru *displayTerm* navodi se vrednost *term* koje se prikazuje krajnjem korisniku. Ovo je posebno značajno kada imamo vrednosti termova koji počinju velikim slovom (Dragan) ili koji imaju neko slovo sa dijakritičkim znakom (Ivanović). U podllementu *extraTermData* moguće je dodati dodatne informacije koje se odnose na term (npr. informaciju na kom jeziku je term).

3.1.4. Obrada grešaka

Sve greške koje se dese prilikom komunikacije klijenta i servera opisane su standardom i predstavljene su elementom *diagnostics*. Server razlikuje dve vrste grešaka: kritične (*fatal*) i nekritične (*non-fatal*). Kada se desi kritična greška server će vratiti samo poruku o grešci i neće ni pokušati da vrati pogotke. Primer kritične greške je kada klijent pošalje upit čija sintaksa nije ispravna i tada server ne može ništa da uradi osim da vrati poruku o grešci. Nekritične greške su one koje, kada se dese, mogu da utiču na rezultat pretrage ali server može da nastavi sa radom. Primer ovakve greške je kada klijent pošalje sintaksno ispravan upit ali upit sadrži neke elemente koje server ne podržava. Server tada može odlučiti da nastavi sa radom a da određene delove upita izostavi prilikom pretrage. Nekritične greške se dalje mogu podeliti u dve podgrupe: *surrogate* i *non-surrogate*. Greške tipa *surrogate* predstavljaju zamenu za zapis koji bi bio odgovoru da nije došlo do greške. Ove greške se mogu nastati, na primer kada server podržava imenovane skupove rezultata i ukoliko klijent posle nekog vremena zatraži zapise iz tog skupa može se desiti da je u međuvremenu neki od zapisa izbrisan iz baze i tada će server vratiti sve zapise koji su i dalje dostupni a umesto onih koji su izbrisani vratiće grešku tipa *surrogate*. Greške tipa *non-surrogate* služe da označe da je došlo do određene greške ali da je server ipak uspeo da vrati pogotke koji možda nisu u skladu sa klijentovim

očekivanjima. Na primer može se desiti da je klijent zahtevao da rezultati pretrage budu sortirani po određenom kriterijumu, a da server to nije mogao da izvrši.

Greške koje server može da vrati su precizno standardom definisane i svaka greška određena je sa tri parametra: *uri*, *details* i *message*. Parametar *uri* predstavlja jedinstveni identifikator greške. Na primer ako je vrednost *uri* parametra *info:srw/diagnostic/1/10* to znači da je greška definisana SRU/W standardom pod rednim brojem 10. U listi registrovanih grešaka može se videti da greška broj 10 znači 'sintaksno neispravan upit'. Parametar *details* može sadržati dodatne informacije koje klijent može da iskoristi kako ne bi ponovo napravio istu grešku. Obično je sadržaj ovog parametra definisan samom vrstom greške. Na primer ukoliko se desi greška gde parametar *uri* ima vrednost *info:srw/diagnostic/1/38*, što znači da je klijent formirao upit koji ima previše operanada, server kao vrednost parametra *details* može vratiti maksimalan broj dozvoljenih operanada u upitu. Parametar *message* sadrži opis greške i eventualno predlog kako da se ta greška izbegne i namenjen je krajnjem korisniku. Jedino je parametar *uri* obavezan dok ostali parametri su opcioni.

3.2. CQL

CQL (Contextual Query Language) je formalni jezik za predstavljanje upita u sistemima za pretragu i pribavljanje najrazličitijih podataka. Sam upitni jezik se zasniva na principima i konceptima iz Z39.50 upitnog jezika. Jezik je dizajniran tako da bude intuitivan i čitljiv za čoveka, i da ujedno poseduje osobinu ekspresivnosti (ekspresivnost je karakteristika kompleksnih upitnih jezika). Prethodno omogućuje da korisnik CQL-om definiše upite proizvoljnih nivoa složenosti. Trenutno je aktuelna verzija 1.2 ovog upitnog jezika.

Upitni jezici mogu se grubo razvrstati u dve kategorije. Klasični predstavnici prve kategorije bi bili SQL, PQF i XQuery. Navedeni jezici su moćni i ekspresivni, mada obični korisnici imaju poteškoća sa njihovom interpretacijom i upotrebom zbog njihove kompleksnosti. U drugu kategoriju spadaju jednostavni i intuitivni upitni jezici (*Common Command Language* ili pretraga na *Google*). Oni su jezici ograničene primene jer je nemoguće izraziti kompleksne upitne koncepte. CQL kombinuje prednosti pomenutih

kategorija, usvajajući jednostavnost i intuitivnost od prve kategorije, i definisanje složenih koncepata iz druge.

3.2.1. Sintaksa CQL upita

CQL upit se može definisati kao pojedinačna klauzula pretrage ili kao kombinacija više klauzula koje su povezane logičkim operatorima. U oba slučaja CQL je omogućeno sortiranje rezultata pretrage, navođenjem sortirajuće naredbe na kraju upita pretrage. Sortiranje se uključuje eksplicitno u upit tako što se na njegovom kraju navede ključna reč *sort by* i zatim se imena indeksa po kojima će se vršiti sortiranje. Moguće je i koristiti i dodatne modifikatore za sortiranje (npr. za definisanje poretka zapisa prilikom sortiranja, sledi primer.

```
dinosaur sortBy dc.date/sort.descending dc.title/sort.ascending
```

Klauzula pretrage može da se sastoji od **indeksa**, **relacije** i **terma** (vrednost traženog izraza), gde je jedino **navođenje** terma **obavezno**. Indeks i relacija se uvek koriste u paru i nemoguće je zadati upit koji bi sadržao samo jedan od pomenutih elemenata. Ukoliko se klauzula pretrage sastoji samo od terma, tada se za indeks i relaciju uzimaju predefinisane vrednosti (indeks "cql.serverChoice" i relacija "="). Moguće oblici klauzula prikazani su sledećim primerima:

- cat
- title = cat
- title = raven and creator = poe
- title = raven sortBy date/ascending

Indeks i relacija uvek pripadaju određenom kontekst setu, te je stoga potrebno ispred njihovih naziva definisati skraćeni naziv kontekst seta (prefiks) praćen simbolom „.“. Indeksi i relacije navedeni bez prefiksa, smatraju se da pripadaju podrazumevanom CQL kontekst setu . U CQL je moguće definisati mapiranje prefiksa, tj. navođenje svih kontekst setova koji se koriste u upitu. Mapiranje kontekst setova realizuje se specifičnom kombinacijom simbol „>“ iza koga sledi skraćeni naziv kontekst seta, simbol „=“ i identifikator kontekst seta (npr. > pro = „http://proffesional.org/“ pro.type = engineer).

Term u klauzuli pretrage predstavlja traženu vrednost indeksa. Ukoliko se term sastoji od više reči ili sadrži neki od specijalnih karaktera „< > = / ()“, tada se on mora definisati između duplih navodnika (npr. “Alexander Graham Bell”). U slučaju da se u samom termu nalazi vrednost „\“, neophodno je koristiti znak “\” ispred vrednosti terma.

Relacija u CQL definiše odnos između indeksa i terma u klauzuli pretrage. CQL podržava standardne relacije (=, >, <, >=, <=, <>, ==). Pored zadavanja relacije kao simbola moguće je relaciju navesti njenim nazivom iz kontekst seta (npr. cql.any). Semantika upotrebe relacija može biti izmenjena upotrebom jednog ili više relacionih modifikatora. Slede primeri:

- dc.title any “fish frog” - Pronalazi sve zapise čiji naslov sadrži bar jednu od reči “fish“ ili „frog” .
- dc.title all “fish frog” - Pronalazi sve zapise gde se u naslovu nalaze i reč “fish“ i „frog”.

CQL definiše dve vrste modifikatora: relacione i logičke. Relacionim modifikatorima dodatno se proširuje značenje relacije, dok su logičkim modifikatorima opisane veze udaljenosti dva terma. Svi modifikatori moraju pripadati određenom kontekst setu, te je za njih potrebno navesti prefiks. Izostavljanjem prefiksa podrazumeva se upotreba podrazumevanog CQL kontekst seta. Modifikatori se navode simbolom “/” iza koga sledi naziv modifikatora i kombinacija relacije i vrednosti. Upotreba modifikatora zavisi od njegove definicije u okviru kontekst seta. Sledi primer:

title =/relevant cat – Pretraga naslova po termu *cat* potrebom *relevancy* algoritma implementiranog na serveru.

title =/relevant cat – Pretraga naslova po termu *cat* potrebom *relevancy* algoritma implementiranog na serveru.

Složeni CQL upiti definišu se upotrebom logičkih operatora, čiji su operandi klauzule pretrage ili složeni podupiti. CQL jezikom definisano je postojanje četiri binarna logička operatora: AND, OR, NOT i Prox. Semantika upotrebe AND i OR logičkih operatora ekvivalentna je onoj u Bulovoj algebri. Tako se AND operatorom povezuju dve klauzule gde je rezultat međusobni presek skupa rezultata, dok je kod operatora OR rezultat unija dva skupa rezultata. NOT operator se ne definiše kao negacija operanda, već kao varijanta AND operatora. Primenom NOT operatora dobija se skup rezultata koji

zadovoljava levi operand, a isključuje skup rezultata definisan u desnom operandu logičkog upita. Operator Prox definiše relativnu udaljenost između termova pretrage. Pošto se termini navode u okviru klauzula pretrage, operandi logičkog upita moraju ukazivati na isti indeks pretrage. Udaljenost između termova određuje se upotrebom logičkih modifikatora. CQL kontekst set definiše modifikatore: jedinica udaljenosti, broj elemenata udaljenosti i poredak termova kao u primeru koji sledi.

dc.title = raven prox/unit=word/distance>3 dc.title = crow - Tražimo reč *raven* u naslovu koja je udaljena najmanje tri reči od reči *crow*.

Svi navedeni segmenti CQL jezika su osetljivi na upotrebu malih i velikih slova. Čitava sintaksa CQL jezika je prikazana u ABNF (*Augmented Backus-Naur Form* - *Augmented Backus-Naur Form*, <http://tools.ietf.org/pdf/rfc5234.pdf>) notaciji. Karakter '=' odvajava element od njegove definicije, znak '/' razdvaja alternativne mogućnosti za element, uglastim zagradama '[']' su uokvireni opcioni elementi dok je znakom '*' predstavljeno višestruko pojavljivanje nekog elementa. Ključne reči koje su deo sintakse uokvirene su dvostrukim navodnicima. Sledi ABNF za sintaksu CQL jezika

; A. Elementi Upita

cql-query = query [sort-spec]

; B. Elementi klauzule za pretragu

query = *prefix-assignment search-clause-group

search-clause-group = search-clause-group Boolean-modified subquery / subquery

subquery = "(" query ")" / search-clause

search-clause = [index relation-modified] search-term

search-term = simple-string / quoted-string

; C. Elementi specifikacije sortiranja

sort-spec = sort-by *index-modified

sort-by = "sortby"

; D. Prefiks elementi

prefix-assignment = ">" [prefix "="] uri

prefix = simple-name

uri = quoted-uri-string

; E. Elementi indeksa

index-modified = index [modifier-list]

index = simple-name / prefix-name

; F. Relacije

relation-modified = relation [modifier-list]

relation = relation-name / relation-symbol

relation-name = simple-name / prefix-name

relation-symbol = "=" / ">" / "<" / ">=" / "<=" / "<>" / "=="

; G. Logički elementi

Boolean-modified = Boolean [modifier-list]

Boolean = "and" / "or" / "not" / "prox"

; H. Modifikatori

modifier-list = *modifier

modifier = "/" modifier-name [modifier-relation]

modifier-name = simple-name

modifier-relation = relation-symbol modifier-value

modifier-value = simple-string / quoted-string

; I. Pseudonimi

prefix-name = prefix "." simple-name

; Prefiks (obično ime) i ime odvojeno tačkom, gde nije dozvoljeno pojavljivanje ;razmaka, pre i posle karaktera "."

quoted-uri-string = URI pod navodnicima

; (RFC 3986 (STD 66) (Uniform Resource Identifier, 2005) određuje dozvoljene znakove za URI

quoted-string = sekvenca karaktera pod navodnicima. Unutar navodnika se može pojaviti znak za navodnike ali onda ispred tog znaka mora stojati znak \.

; J. opšti termini

simple-name = simple-string

simple-string =

;Bilo koja sekvenca karaktera koja ne sadrži (,),=>,<,"/, ili prazan karakter.

Kao što smo videli CQL upitni jezik definiše različite koncepte i veliki spektar mogućnosti za formiranje upita, ali SRU/W standardom nije propisano da server mora da implementira sve te koncepte. Stoga su definisana tri nivoa koji se odnose na stepen složenosti CQL upita koji server može da parsira i obradi. Podrazumeva se da usklađenost sa tri nivoa složenosti CQL upita mora postojati i na strani klijenta.

Nivo 0 podrazumeva da je klijent u stanju da formira upit koji se sastoji samo od terma koji se traži, odnosno klijent ne definiše ni indeks ni relacije već to server sam određuje. Term je najčešće jedna reč ili niz reči pod navodnicima. Server treba da bude u stanju da procesira pristigli upit ili da u slučaju da klijent pošalje upit koji server ne podržava vrati odgovarajuću poruku o grešci.

Nivo 1 podrazumeva da i klijent i server podržavaju pravila definisana u nivou 0. Nivo 1 obuhvata pravila definisana nivoom 0 i proširuje ta pravila tako da server može da parsira:

- a) upite koji se sastoje od indeksa, relacije i terma,
- b) upite kod kojih su operandi povezani logičkim operatorima. Pod operande na ovom nivou podrazumevamo samo termove a ne i kombinaciju indeks relacija term koju server na ovom nivou nije dužan da podrži.

Bitno je istaći da server mora da podrži parsiranje upita navedenog pod a) i upita pod b), ali da ne mora da podrži obradu oba tipa upita već može podržati samo jedan od ta dva tipa.

Sa stanovišta klijenta nivo 2 podrazumeva potpunu podršku nivoa 1 kao i mogućnost da se iskažu svi elementi iz CQL upitnog jezika i njegovih

kontekst setova. Na nivou 2 server mora biti u stanju da parsira bilo koji upit po CQL upitnom jeziku, ali i dalje ne mora implementirati sve neobavezne koncepte, gde u tim slučajevima mora vratiti poruku o grešci.

3.3. Context Set

CQL (*Contextual Query Language*) nosi pomenuto ime jer se koncept pretrage zasniva više na semantici nego na sintaksi. CQL jezik svoju pretragu zasniva na metapodacima tj. na pretrazi po kontekstu koja ne zavisi od tipa podataka. Upotrebom metapodataka omogućuje se da se isti CQL upiti različito interpretiraju nad određenim skupovima i bazama podataka. Nazivi meta podataka u CQL jeziku navode se unutar kontekst setova, tj. u indeksima CQL jezika. Kontekst setovi su okarakterisani sledećim konceptima:

- indeksi,
- relacije,
- modifikatori relacija i
- modifikatori logičkih operatora.

Postojanje kontekst seta omogućuje da nezavisni korisnici (zajednice) definišu sopstvenu semantiku upotrebe CQL čime dobijamo potencijalnu interoperabilnost kod različitih domena primene. Prilikom definisanja novog kontekst seta ne moraju se obavezno definisati svi njegovi koncepti. Štaviše, u većini slučajeva za kontekst setove se navode samo indeksi, dok su ostali elementi kontekst seta izostavljeni, gde naravno postoje jasna pravila šta se uzima kao podrazumevana vrednost za izostavljene elemente koncepta. Korisnicima je dozvoljeno definisanje svih postojećih koncepata u sopstvenom kontekst setu, bez bojazni da se isti nazivi indeksa, relacija ili modifikatora, mogu naći u drugim kontekst setovima (potencijalna dvosmislenost CQL upita). Prethodno je prevaziđeno definisanjem identifikatora i skraćenog naziva svakog kontekst seta. Identifikator (URI) i/ili skraćeni naziv se koriste unutar CQL upita određujući pripadnost elementa kontekst setu. Najčešće su URI oblika „http“ ili „info“.

Pripadnost elementa kontekst setu definiše se prefiksom samog elementa (indeksa, relacije ili modifikatora) u CQL upitu. Na primer ako je definisan PROFESIONAL kontekst set i njegov skraćeni naziv pro, tada se u CQL

upitu može koristiti struktura *pro.type*, što znači da se pretražuje po indeksu *type* čija je semantika određena kontekst setom PROFESSIONAL. CQL dozvoljava da se dva istoimena indeksa različito upotrebljavaju, samo kada oni pripadaju različitim kontekst setovima. Ukoliko u upitu nije eksplicitno definisan *Context Set* za prefikse koji se pojavljuju u upitu smatra se da pripadaju CQL *Context Set*. Kontekst setovi se mogu registrovati u okviru Kongresne biblioteke i time bivaju dostupni svim korisnicima (Registered Context Sets, 2011). Asocijacija prefiksa za URI je definisana u *Explain* servisu ili direktno u upitu kao npr.

> *dc = "info:srw/context-sets/1/dc-v1.1" dc.title = cat*

U navedenom upitu se za URI *dc = "info:srw/context-sets/1/dc-v1.1"* asocira ime prefiksa *dc* što ukazuje na Dublin Core kontekst set. Treba biti obazriv da je vezivanje prefiksa za URI u samom upitu lokalnog karaktera, odnosno da ne važi u narednim upitima.

Neki od poznatih registrovanih kontekst setova su: CQL verzije 1.2, Dublin Core verzije 1.1, ZeeRex, Bath 2.0, Bibliographic verzije 1 itd. U nastavku će biti detaljno opisani kontekst setovi koji su bili značajni za CRIS UNS sistem pretrage naučno-istraživačkih podataka.

3.3.1. CQL Context Set

CQL Context Set (CQL Context Set, version 1.2, 2009) je podrazumevani kontekst set koga server po SRU/W mora da podrži. Pomenuti kontekst set se najviše koristi obzirom da su njime definisani indeksi, relacije kao i modifikatori koji imaju opštu primenu (mogu se lako koristiti u više različitih konteksta/domena). Trenutno su aktuelne verzije 1.1, 1.2, 2.0. Rezervisano ime ovog konteksta seta je „cql“, dok je identifikator *info:srw/cql-context-set/1/cql-v2.0*. CQL Context Set definiše sledeće indekse:

serverChoice – Ukoliko korisnik definiše upit koji sadrži indeks *serverChoice* to znači da server ima slobodu da pretraži po onim indeksima koje on izabere i pri tome ne mora prilikom svakog upita koristiti iste indekse kao podrazumevane. Ukoliko korisnik prilikom kreiranja upita ne definiše indeks, server će smatrati da je zadati indeks *serverChoice*. *serverChoice* indeks se može i eksplicitno navesti u formi

cql.serverChoice=ime_zadatog_terma i u tom slučaju se za zadati term pretražuje po indeksima koje odabere server.

resultSetId – ovaj indeks se koristi kad se u upitu koriste imenovani skupovi, tada je vrednost ovog indeksa u stvari naziv imenovanog skupa koji je server kreirao u nekom od prethodnih upita. Ovaj indeks se može kombinovati samo sa relacijom '=' dok sa ostalim relacijama poretka nema definisano značenje. Takođe, ovaj atribut se ne može koristiti sa *Scan* servisom. Inicijalno se ovaj indeks koristi kao jednostavna klauzula pretrage gde indeks *cql.resultSetId* = identifikator skupa koji prethodno generiše server, kao što je navedeno u primeru a. Treba imati naumu da je moguće povezati dva skupa podataka logičkim operatorom kao u primeru b, gde se u tom slučaju kao rezultat logičkog poređenja dobija novi skup zapisa. Kad želimo da dobijemo samo određene rezultate (filtriramo) iz postojećeg seta podataka iskoristili bili oblik kao u primeru c.

- a) *cql.resultSetId* = "5940824f-a2ae-41d0-99af-9a20bc4047b1" – Dobijamo sve zapise iz skupa sa navedenim identifikatorom.
- b) *cql.resultSetId* = "a" AND *cql.resultSetId* = "b" – Nastaje novi skup zapisa (treći) koji je presek postojeća dva.
- c) *cql.resultSetId* = "a" AND *dc.title=cat* – Primenjuje se *dc.title=cat* na skup podataka pod nazivom a.

allRecords – specijalni indeks koji se koristi ukoliko je potrebno vratiti sve zapise koji postoje u bazi i pri tome nije bitno koje su vrednosti navedene za term i/ili relaciju. Preporučena sintaksa za upotrebu ovog indeksa je *cql.allRecords=1* pri čemu u samoj pretrazi term „1“ se neće uzeti u obzir, umesto broja 1 mogao je da stoji bilo koji drugi term. Moguće je kombinovati ovaj indeks sa drugim upitima upotrebom logičkih operatora (primer A). Ovaj indeks se ne može koristiti sa *Scan* servisom.

- A. *cql.allRecords* = 1 NOT *dc.title* = dog – Pretražuje sve zapise koji nemaju term dog u naslovu.

Relacija nije jednako „<>“ omogućava da se dobiju svi zapisi koji ne sadrže navedeni term iz relacije, kao npr.

dc.date <> 2004-01-01 – Svi zapisi koji u polju date nemaju vrednost 1. januar 2004. godine.

U nastavku teksta slede niz relacija koje deo *CQL Context Set*.

adj – ova relacija se koristi prilikom pretraživanja fraza. Odnosno sve reči koje se pojavljuju u termu se u istom redosledu moraju naći i u zapisu. Relacija *adj* se izvršava sa implicitnim modifikatorom ('*cql.word*'), čija vrednost može biti promenjena prilikom definisanja upita. Treba naglasiti da se relacija *adj* može iskazati i PROX logičkim operatorom, tako upit *title adj "a b c"* biva ekvivalentan sa *title=a prox/distance=1/ordered title=b prox/distance=1/ordered title=c*

Prazan karakter je inicijalni delimiter kod terma koji sadrži više reči i koristi se kao vrednost za *adj* relaciju. Može se definisati i drugi delimiter, ali je neophodno da se ta informacija publikuje kroz *Explain* servis.

all, any – ove relacije se mogu koristiti kada term sadrži više reči i tada se sa relacijom *all* može definisati da sve reči moraju biti pronađene u izabranom indeksu ali da redosled reči u indeksu nije bitan. Slično, relacija *any* definiše da indeks može sadržati bilo koju od datih reči. Relacija *all* se može zameniti odgovarajućim upitom pri čemu svi operandi moraju biti povezani logičkim operatorom AND, dok se relacija *any* može zameniti odgovarajućim upitom pri čemu svi operandi moraju biti povezani logičkim operatorom OR. Ove relacije implicitno sadrže relacioni modifikator '*cql.word*' koji se može promeniti prilikom definisanja upita. Relacija *all* se može koristiti sa relacionim modifikatorom „*windowSize*“, kako bi zahtevali da se reči u *all* zadate u „prozoru“ (opsegu) kao npr.

dc.title all/windowSize=6 "cat hat rat" – Tražimo reči "cat", "hat", i "rat" u prozoru od 6 reči.

within – ova relacija se koristi kada je potrebno definisati neki opseg kom može pripadati term. Posebno može biti korisna ukoliko se ne zna tačna vrednost za neki indeks ali se zna opseg vrednosti, na primer potrebno je pronaći knjigu čija je godina izdavanja između 2008 i 2014. (*dc.date within "2008 2014"*)

encloses – relacija koja grubo predstavlja inverznu relacije za *within*. Ova relacija se koristi kada se želi istaći da rezultati pretrage uključuju dati term. Na primer, ukoliko se pretražuju neki geografski podaci i postoji indeks *area*, tada bi upit *area encloses 45.3,19.0* gde term predstavlja koordinate neke

tačke, trebao da vrati sve rezultate kod kojih vrednost indeksa *area* uključuje datu tačku.

geo.area encloses "45.3 19.0" – Tražimo koordinatu koja se nalazi između tačaka 45.3, 19.0.

Modifikatori relacija

Modifikatori se koriste da relaciji daju novo značenje, u cilju da se skup rezultata pretrage što više suzi kako bi korisnik dobio samo one rezultate koji su njemu relevantni. Modifikatori relacija se mogu podeliti na funkcionalne modifikatore i modifikatore koji određuju format terma koji se traži. Funkcionalni modifikatori su:

relevant – upotrebom ovog modifikatora server bi trebao da vrati sve rezultate koji su relevantni sa traženim termom. Na primer ako je korisnik zadao upit sa termom 'voće' tada bi kao rezultat trebalo vratiti i pogotke koji sadrže reč 'banana', 'jabuka', i slično. Algoritam kojim će se odrediti relevantnost definiše server.

fuzzy – upotrebom ovog modifikatora server vraća sve pogotke koji sadrže izraz sličan traženom termu. Algoritam za određivanje sličnosti definiše server. Ovaj modifikator može biti koristan kada imamo term u kome postoji neka slovna greška tj. postoji nekakva permutacija pojedinih karaktera ka npr.

dc.title=/fuzzy "Satrac i more"

partial – ovaj modifikator se može koristiti sa relacijama *within* i *encloses*. U tom slučaju server bi trebao da vrati i one podatke koji delimično zadovoljavaju datu relaciju.

ignoreCase, respectCase – ovim modifikatorima se serveru daju uputstva da li da zanemari ili ne, mala i velika slova prilikom pretrage. Ovi modifikatori nalaze posebno nalaze svoju primenu prilikom sortiranja, kada je neophodno voditi računa o velikim i malim slovima.

ignoreAccents, respectAccents – ovim modifikatorima se serveru daju uputstva da li da zanemari ili ne, dijakritike koji se mogu pojaviti u nekim pismima (na primer srpska latinica, nemački alfabet i slično). Ovi modifikatori nalaze svoju primenu prilikom sortiranja gde je neophodno voditi računa o rezultatima sa dijakriticima.

locale – modifikator se koristi u slučajevima kada se upit treba tretirati kao da je iz nekog specifičnog lokaliteta. Lokaliteti su specifični identifikatori koji ukazuju na niz opcija u skladu sa vremenskom zonom, jezikom, zemljom, jedinicama mere. Identifikator lokaliteta se koristi u opštem obliku modifikator *locale=identifikator*. Vrednosti identifikatora su propisane od *Unicode Consortium* u okviru projekta pod nazivom *Common Locale Data Repository* (CLDR) (“CLDR - Unicode Common Locale Data Repository,” 2014), gde se spisak svih registrovanih vrednosti nalazi na (<http://www.iana.org/assignments/language-subtag-registry>). Sve vrednosti su dvoslovne skraćenice koje reprezentuju zemlju lokaliteta (npr. es za Španiju, en za Englesku). Mada postoji situacija kada se jedan isti jezik razlikuje u različitim državama, te se tada dvoslovnna karakteristika proširuje po ISO 3166 (ISO 3166 - Country codes - ISO, 2012) i tako na npr. za Veliku Britaniju je skraćenica en-UK ili za Sjedinjene Američke Države en-US.

dc.title within/locale=fr "l m" 162 – pronađi sve naslove između l i m, gde će lokalitet fr odrediti šta je između l i m.

weight – ovaj modifikator se koristi kako bi se dodala težina za klauzulu pretrage koja je relativna u odnosu na druge klauzule. Vrednosti težina su pozitivne celobrojne vrednosti, a inicijalna vrednost je 1.

Format terma koji se pretražuje definiše se specifičnim modifikatorima. Modifikatori koji određuju format terma su:

word – ukoliko se term sastoji od više reči treba ga tako i posmatrati (kao skup reči).

string – ukoliko se term sastoji od više reči treba ga posmatrati kao jednu jedinstvenu reč, odnosno kao frazu.

isoDate – term predstavlja datum po ISO 8601 specifikaciji.

number – term predstavlja broj. Ovaj modifikator posebno dolazi do izražaja kada se koriste relacije poretka <, >, >=, >=.

uri – kompletan term predstavlja URI identifikator

oid – term predstavlja identifikator objekta definisan po ITU-T's ASN.1 - X.690 standardu.

masked – ovo je podrazumevani modifikator relacije i označava da specijalni znaci '*', '?' i '^' imaju posebnu ulogu. Eksplicitna upotreba ovog modifikatora navodi se upotrebom '*cql.masked*' kao relacioni modifikator. Znak '*' zamenjuje nula ili više karaktera u termu, a znak '?' zamenjuje tačno jedan karakter. Znak '^' se koristi ukoliko se term sastoji od više reči a data je jedna od relacija all, any, ili adj. Pomenuti znak se može pojaviti na početku reči ili na kraju reči i označava da se reč uz koju stoji mora pojaviti na početku indeksiranog sadržaja, odnosno na kraju. Na primer ako je dat upit *dc.title all ^cat dog*, to znači da se traže oni zapisi koji u naslovu sadrže i reč *cat* i reč *dog* ali se reč *cat* mora naći na početku naslova, nijedna druga reč ne može biti ispred nje. Ukoliko je neki od navedenih karaktera deo terma i želimo da ga preskočimo upotrebljava se karakter '\\'. U nastavku su navedeni i primeri korišćenja za modifikator *masked*:

*dc.title = c*t* – Pronalazi sve reči koje počinju sa c i završavaju se sa t bez obzira koliko reči slova imaju te reči.

*dc.title adj "*fish food"* – Pronalazi sve naslove čija je pretposlednja reč *fish* i gde nakon nje sledi reč koja počinje sintagmom *food*.

dc.title = c?t - Pronalazi reči u naslovu koje imaju tačno tri slova gde reč počinje karakterom c a završava s karakterom t.

dc.title adj "^cat in the hat" - Pronalazi sintagmu *cat in the hat* koja se mora nalaziti na početku naslova.

dc.title any "^cat ^dog rat^" - Pronalazi sintagmu *cat in the hat* koja se mora nalaziti na kraju naslova.

unmasked – ovaj modifikator se koristi ukoliko je potrebno da se specijalni znaci '*', '?' i '^' tumače kao i obični karakteri, znači bez posebnog značenja. Isti efekat se može postići ukoliko se ispred svakog specijalnog znaka stavi znak '\\'.

honorWhitespace - namena ovog modifikatora je uzimanje u obzir belih (nevidljivih) karaktera u zadatom termu i najčešće se koristi u kombinaciji sa relacijom ==. U slučaju da se ne navede ovaj modifikator na serveru je da odluči da li će se nevidljivi karakteri uzimati u obzir.

substring – ovaj modifikator se može koristiti za definisanje opsega (prvog i poslednjeg) karaktera koji se posmatraju u indeksu koji se traži. Odnosno

traženi term može biti podskup vrednosti indeksa po kom se pretražuje. Sintaksa modifikatora je: *substring=početak* : kraj, pri čemu su početak i kraj celi brojevi koji označavaju respektivno prvi i poslednji karakter koji se posmatraju u datom indeksu. Prilikom definisanja ovog modifikatora važe sledeća pravila:

- Pozitivne celobrojne vrednosti se računaju "unapred", počevši od 1.
- Negativne celobrojne vrednosti računaju se "unazad", tako što se pozicija-1 smatra poslednjim karakterom
- početak i kraj u sintaksi se smatraju kao inkluzivne pozicije
- Ako se izostave početak i kraj, podrazumevana vrednost da se za početnu vrednost pozicije uzima 1 a za kraj -1

Na primer ako je postavljen upit *marc.008 =/substring="1:6" 920102*, pri čemu je 920102 term koji se traži, to znači da se pretražuje po indeksu 008 koji pripada *Marc Context Set*-u ali se posmatra samo prvih 6 karaktera sadržaja u datom indeksu.

regex – modifikator označava da je dati term predstavljen u vidu POSIX regularnog izraza [] i server bi trebalo da poseduje algoritam koji je u stanju da parsira taj regularni izraz.

*dc.title adj/regex "(lord|king|ruler) of th[ea] r.*s"* - Pronalazi u naslovu reči "lord" ili "king" ili "ruler" nakon čega sledi reč *of*, zatim *the* ili *tha* i na kraju r i 0 ili više pojava karatera s.

Modifikatori logičkih operatora

CQL Context Set definiše samo pet modifikatora logičkih operatora i ovi modifikatori se mogu koristiti samo sa logičkim operatorom *Prox*.

distance – označava udaljenost između dva terma. Opšti oblik upotrebe ovog modifikatora je *distance* simbol vrednost. Simbol predstavlja neki od relacionih operatora < > <= >= = <>. U slučaju da se simbol izostavi podrazumevana vrednost je <=. Vrednost bi trebala da pozitivna celobrojna vrednost. Kada se ovaj deo modifikatora ne navede podrazumevana vrednost je 1.

container - ovaj modifikator se koristi kada želimo da pretražujemo "kontejner". Kontejner predstavlja skupinu više indeksa, tako npr. kontejner

pod nazivom autor može da sadrži indekse "ime" i "datum". U tom slučaju server omogućava postavljanje upita za pretragu autora sa specifičnim imenom i datumom. Ovakvom konstrukcijom upita ne bi smo dobili neželjene rezultate (više autora sa željenim imenom ali ne odgovarajućim datumom i obrnuto) kao kod logičkog (bulovskog) upita. Ukoliko server podržava ovu funkcionalnost mora je i publikovati putem *Explain* servisa, gde se tačno mora navesti koji indeksi su vezani za koji kontejner.

unit – definiše format terma. Predefinisane vrednosti za ovaj modifikator su *paragraph*, *sentence*, *word* i *element*, s tim da je *word* podrazumevana vrednost modifikatora. Svaku od ovih vrednosti server može interpretirati na sebi svojstven način.

unordered – ovim modifikatorom je definisano da redosled termova koji se traže nije bitan.

ordered – ovim modifikatorom je definisano da je redosled termova koji se traže mora biti identičan kao i u upitu.

Sledeći primeri prikazuju upotrebu logičkih modifikatora:

cat prox/unit=word/distance>2/ordered hat - Pronalazi reč "cat" koja se nalazi na minimalno dve reči pre reči "hat"

cat prox/unit=paragraph hat - Pronalazi zapis koji ima paragraf u kome se nalaze reči "cat" i "hat"

name=jones prox/container=author date=1950 - pronalazi ime "jones" u vezi sa godinom 1950, gde su indeksi *name* i *date* vezani za kontejner *author*.

3.3.2. Dublin Core Context Set

Dublin Core Context Set (Dublin Core Context Set Version 1.1, 2007) je nastao na bazi Dublin Core formata (Dublin Core® Metadata Initiative (DCMI) Format, 2014) koji je najčešće korišćeni format za razmenu najrazličitijih informacija (video materijali, slikem veb stranice ...) u informacionim sistemima. Format definiše listu elemenata i atributa za opis određenog resursa. Listu čine 15 meta termova poznatiji kao *Dublin Core Metadata Element Set* (<http://dublincore.org/documents/dces/>) koji je detaljno opisan sledećim dokumentima: IETF RFC 5013, ISO Standard 15836-2009 i NISO Standard Z39.85. Za razvoj pomenutog formata

zadužena je DCMI (*Dublin Core Metadata Initiative*). Opšta semantika i preporuke za korišćenje Dublin Core indeksa data je na (<http://dublincore.org/documents/usageguide/elements.shtml>). *Dublin Core Context Set* je jedan od najčešće korišćenih skupova prilikom pretraživanja upotrebom SRU/W protokola. Ovaj skup definiše samo indekse, stoga se najčešće kombinuje sa *CQL Context Set*. Indeksi *Dublin Core Context Set* su preuzeti iz specifikacije Dublin Core standarda za opis metapodataka. Stoga treba naglasiti da se *Dublin Core Context* sastoji od 15 indeksa. Trenutno aktuelna verzija Dublin Core Context je **1.1**, dok je URI ovog konteksta seta *info:srw/cql-context-set/1/dc-v1.1*, a preporučeno skraćeno ime je **dc**. Dublin Core standard za opis metapodataka se može koristiti za opis različitih resursa, i jedan od tih resursa mogu biti i naučno-istraživačke informacije. U poglavlju **3.4** dat je spisak korišćenih indeksa koji pripadaju *Dublin Core Context Set* skupu kao i njihova semantika kada se *Dublin Core Context Set* koristi za opis naučno-istraživačkih podataka. Indeksi koji ulaze u sastav *Dublin Core Context Set* su:

1. Title
2. Creator
3. Subject
4. Description
5. Publisher
6. Contributor
7. Date
8. Type
9. Format
10. Identifier
11. Source
12. Language
13. Relation
14. Coverage
15. rights

3.3.3. Bib Context Set

Bib context set je skup koji bi bio najpogodniji za predstavljanje pretraživih podataka iz bibliografskog domena. Tako se ovim kontekst setom definišu neophodni bibliografski indeksi i modifikatori. Indeksi i modifikatori se baziraju na MODS standardu (Metadata Object Description Schema: MODS (Library of Congress), 2013), gde je MODS iskorišćen za definisanje semantike, što svako ne podrazumeva da podaci moraju biti u skladu sa MODS standardu. Jedinstveni identifikator kontekst seta je *info:srw/cql-context-set/1/bib-v1*, dok je preporučeno skraćeno ime **bib**. U tabeli 3.1 navedeni su indeksi i dodati modifikatori za *Bib Context Set*. Primetno je da su modifikatori i indeksi podeljeni u niz kategorija. Kategorija opisuje skup indeksa i modifikatora sa vrlo bliskom semantikom. Tako se kategorija *Title* donosi indekse koji predstavljaju nekakav vid naslova. U pomenutoj kategoriji su svrstani sledeći indeksi *bib.titleAbbreviated* (skraćeni naziv resursa), *bib.titleUniform* (radna verzija naslova – ona koja se trenutno koristi), *bib.titleTranslated* (prevedeni oblik naslova), *bib.titleAlternative* (alternativni oblik naslova), *bib.titleSeries* (naslov serijala). Za kategoriju *Title* vezuju se modifikatori indeksa *bib.portion* i *bib.titleAuthority*. Modifikator *bib.portion* dodatno klasifikuje vrstu naslova sledećim vrednostima: *main* (glavni naslov), *sub* (podnaslov), *partNum* (broj poglavlja), *partName* (naziv poglavlja).

Kategorija	Indeksi	Modifikatori
Title	bib.titleAbbreviated bib.titleUniform bib.titleTranslated bib.titleAlternative bib.titleSeries	bib.portion bib.titleAuthority
Name	bib.name bib.namePersonal bib.namePersonalFamily bib.namePersonalGiven bib.nameCorporate bib.nameConference	bib.date bib.nameAuthority bib.role bib.roleAuthority
Subject	bib.subjectPlace bib.subjectTitle bib.subjectName bib.subjectOccupation	bib.subjectAuthority
Identifier		bib.identifierAuthority

Date	bib.dateIssued bib.dateCreated bib.dateValid bib.dateModified bib.dateCopyright	bib.dateAuthority
Resource Type		bib.typeAuthority
Format		bib.formatAuthority
Genre	bib.genre	bib.genreAuthority
Target Audience	bib.audience	bib.audienceAuthority
Classification	bib.classification	bib.classAuthority
Place of Origin	bib.originPlace	bib.geoUnit bib.placeAuthority
Language		bib.languageAuthority
Edition	bib.edition	
Part	bib.volume bib.issue bib.startPage bib.endPage	
Issuance	Issuance	

Tabela 3.1 Sadržaj *Bib Context Set*

Modifikator *bib.titleAuthority* se koristi samo sa indeksom *titleUniform* i njegova vrednost je *lcnaf* (Library of Congress Authorities File, 2012). *Library of Congress Name Authority File* sadrži autorizacione podatke o imenima lica, organizacijama, mestima i titulama. Njegova svrha je identifikacija ovih subjekata i korišćenje takvog kontrolisanog rečnika da bi se obezbedio uniforman pristup bibliografskim izvorima. Kategorija *Name* grupiše indekse koji se odnose na neki oblik imena resursa. U grupu indeksa *Name* svrstavaju se sledeći indeksi: *bib.name* (puno ime resursa), *bib.namePersonal* (lično ime resura), *bib.namePersonalFamily* (porodično ime), *bib.namePersonalGiven* (prezime), *bib.nameCorporate* (koorporativno ime) i *bib.nameConference* (ime događaja-konferencije na kojoj je publikovan zapis). Relacioni modifikatori iz grupe dodatno opisuju indekse iz grupe. Tako modifikatorom *bib.date* možemo definisati pretragu po nekom imenu koje je direktno vezano za određeni datum, kao npr

bib.namePersonal=/bib.date="1835-1913" "Albert Babeau"

Modifikator *bib.nameAuthority* gde ime dovodimo u kontekst sa nekim autorizacionim telom, kao npr.

bib.NamePersonal=/bib.nameAuthority=lcnaf/bib.role=composer/bib.roleAuthority=marcrelator "Beethoven, Ludwig van, 1770-1827"

Modifikatorom *bib.role* definišemo vezu imena i uloge (kao u prethodnom primeru gde je *Beethoven* vezan za ulogu „composer“). Modifikator *bib.roleAuthority* definiše autorizaciono telo koje propisuje moguće uloge (*bib.role*). Podrazumevana vrednost ovog parametra je „marcrelator“ što je samo jedna od mogućih klasifikacija rola koje su definisane na (<http://www.loc.gov/marc/sourcecode/relator/relatorlist.html>).

Kategorija *Subject* se odnosi na indekse koji resurs mogu pretraživati po različitim aspektima meta atributa predmet. Tako indeksom *bib.subjectPlace* dopunjujemo upit sa mestom za koji je neki resurs vezan npr.

bib.name="Roman o Londonu" AND bib.subjectPlace=london

Indeksi *bib.subjectTitle* *bib.subjectName* se odnosi na naslov i ime predmeta pretrage respektivno. Indeksom *bib.subjectOccupation* se navodi profesija (zanimanje) subjekta. Za ovu grupu indeksa vezuje se samo modifikator *bib.subjectAuthority* čije moguće vrednosti su: *marcgac* (<http://www.loc.gov/marc/geoareas/gacshome.html>), *marccountry* (http://www.loc.gov/marc/countries/cou_home.html), *iso3166* (http://www.iso.org/iso/country_codes.htm), *lcs* (<http://id.loc.gov/authorities/subjects.html>), *lcnaf*, čime definišemo autorizacioni okvir za neki kod indeksa kao npr.

bib.subjectName=/bib.subjectAuthority=lcnaf "Williams, Ted, 1918-2002"

Kategorija *Identifier* takođe sadrži samo jedan modifikator pod nazivom *bib.typeAuthority*. Pomenuti modifikator može imati sledeće vrednosti: *hdl*, *doi*, *isbn*, *isrc*, *ismn*, *issn*, *local*, *lcn*, *stock-number*, *uri*. Za svaku od navedenih vrednosti respektivno se vezuju sledeći URI:

- [info:/srw/1/vocabulary/identifierType/hdl](http://info:srw/1/vocabulary/identifierType/hdl)
- [info:/srw/1/vocabulary/identifierType/doi](http://info:srw/1/vocabulary/identifierType/doi)
- [info:/srw/1/vocabulary/identifierType/isbn](http://info:srw/1/vocabulary/identifierType/isbn)
- [info:/srw/1/vocabulary/identifierType/isrc](http://info:srw/1/vocabulary/identifierType/isrc)
- [info:/srw/1/vocabulary/identifierType/ismn](http://info:srw/1/vocabulary/identifierType/ismn)
- [info:/srw/1/vocabulary/identifierType/issn](http://info:srw/1/vocabulary/identifierType/issn)

- info:srw/1/vocabulary/identifierType/local
- info:srw/1/vocabulary/identifierType/lccn
- info:srw/1/vocabulary/identifierType/stock-number
- info:srw/1/vocabulary/identifierType/uri

Podjednako se može koristiti term ili URI, gde je preporuka da se uvek ako je moguće pre koristi URI nego term. Kad god se ne navede URI nego term (prefiks), tada se prefiks tretira iz URI koji ima sledeći oblik „info:srw/resultCountPrecision/1/“. U navedenom URI deo putanja „1“ se odnosi na autorizaciono telo, konkretno sa navedenim brojem misli se na *SRU Maintenance Agency*. Druga autorizaciona tela se registruju posebnim zahtevom, za detaljnije informacije pogledati na (<http://www.loc.gov/standards/sru/resources/infoURI.html>). Treba napomenuti da je korišćenje „info“ tipa URI nije obaveza već samo preporuka.

Kategorija *Date* sadrži indekse koji definišu indekse za specifične vrste datuma. Tako se u ovoj kategoriji nalaze indeksi: *bib.dateIssued* (datum izlaska periodike), *bib.dateCreated* datum kreiranja bibliotečkog resursa, *bib.dateValid* (datum validnosti resursa), *bib.dateModified* (poslednje izvršena modifikacija na resursu), *bib.dateCopyright* (datum važenja autorskih prava resursa). Kategoriji *Date* pridružen je jedan modifikator pod nazivom *bib.dateAuthority*. Pomenuti identifikator ima dve moguće vrednosti *w3cdtf* (<http://www.w3.org/TR/NOTE-datetime>) i *edtf* [<http://www.loc.gov/standards/datetime/>], gde oba ukazuju na autorizaciono telo koje propisuje format datuma.

Kategorija *Resource Type* ne sadrži nijedan indeks već samo jedan modifikator pod nazivom *bib.typeAuthority*, tako da se ovaj modifikator može koristiti i sa drugim indeksima. Vrednost koja se koristi sa ovim modifikatorom je „*modsResource*“, koja ukazuje na to da se koristi lista tipova resursa (<http://id.loc.gov/vocabulary/resourceTypes.html>) iz MODS šeme.

*dc.type=/bib.typeAuthority=modsResource text AND
bib.namePersonal="matilda plews"*

Kategorija *Format* sadrži samo modifikator *bib.formatAuthority* koji definiše oblik format nekog autorizacionog tela i često se koristi i sa indeksima iz drugih kontekst setova kao npr.

*dc.format=/bib.formatAuthority=modsPhysicalForm print AND
bib.namePersonal="matilda plews"*, gde se *modsPhysicalForm* odnosi na listu sa (<http://www.loc.gov/marc/sourcecode/form/formlist.html>).

Klasifikacija *Genre* sadrži po jedan indeks (*bib.genre*) i modifikator (*bib.genreAuthority*). Vrednosti za indeks *bib.genre* se navode iz kontrolisanog rečnika koji vezujemo za autorizaciono telo koje ga propisuje i ta vrednost se navodi u *bib.genreAuthority*, kao u primeru koji sledi.

*bib.genre=/bib.genreAuthority=modsGenre "humor, satire" AND
bib.namePersonal="dan jenkins"*

Podrazumevana vrednost indeksa *bib.genre* u slučaju da se ne navede eksplicitno može biti regulisano i kroz *Explain* servis. Dok je podrazumevana vrednost modifikatora *bib.genreAuthority modsGenre* i ona se odnosi na listu žanrova propisanu na (<http://www.loc.gov/marc/sourcecode/genre/genrelist.html>).

Klasifikacija *Target Audience* sadrži indeks *bib.audience* i modifikator *bib.audienceAuthority* za pretragu koji se odnosi na konzumente tog bibliografskog resursa. Indeks *bib.audience* navodi tip publike kojoj je bibliografski resurs namenjen, dok se modifikatorom *bib.audienceAuthority* navodi autorizaciono telo koje propisuje tipove publike kao u primeru koji sledi:

*bib.audience=/bib.audienceAuthority=modsAudience adolescent AND
bib.namePersonal="matilda plews"*

Incijalna vrednost modifikatorom *bib.audienceAuthority* je *modsAudience* i ona dolazi iz kontrolisanog rečnika koji je dostupan na (<http://www.loc.gov/marc/sourcecode/target/targetlist.html>).

Kategorija *Classification* definiše kako se klasifikuje bibliografski resurs. Ova klasifikacija sadrži indeks *bib.classification* i modifikator *bib.classAuthority*. Vrednosti indeksa *bib.classification* su iz kontrolisanog rečnika koji zavisi od autorizacionog tela koja ga propisuje, a navodi se kao vrednost modifikatora *bib.classAuthority* kao u primeru koji sledi:

bib.classification=/bib.classAuthority=lcc RF110-320

lcc je autorizaciono telo koje se navodi u listi (<http://www.loc.gov/marc/sourcecode/classification/classificationsource.html>) i zapravo ukazuje na organizaciju "Library of Congress classification". Što se tiče vrednosti indeksa *bib.classification*, RF110-320 gde R klasa odnosno RF podklasa Otorinolaringologija sa brojem koji bliže određuje da su u pitanju bolesti uha.

Kasifikacijom *Place of Origin* se omogućuje pretraga bibliografskog zapisa po mestu njegovog nastanka. Klasifikacija sadrži indeks *bib.originPlace* i modifikatore *bib.geoUnit* i *bib.placeAuthority*. Tako se vrednosti indeksa *bib.originPlace* moraju uskladiti sa vrednostima navedenim u modifikatorima. U nastavku je dat primer koji pokazuje upotrebu indeksa i modifikatora iz ove klasifikacije

*bib.originPlace=/bib.geoUnit=country/bib.placeAuthority=marcCN cuba
AND bib.namePersonal="livan hernandez"*

Naime, ukoliko je vrednost *originPlace cuba* ona se detaljnije specificira modifikatorom *bib.geoUnit* koji ima vrednost *country* što ukazuje da je pojam **cuba** Država i da je autorizacionim telom (*bib.placeAuthority=marcCN*) potvrđeno da je vrednost **cuba** definisana u nekom kontrolisanom rečniku (<http://www.loc.gov/marc/countries/>). Vrednost relacionog modifikatora *geoUnit* takođe može biti i **city**. Vrednost *marcCN* modifikatora *bib.placeAuthority* propisuje da se kao vrednost indeksa *originPlace* mora koristiti pun naziv države (**cuba**). Da je kao vrednost modifikatora *bib.placeAuthority* stajalo *marcCC* tada bi u našem primeru umesto vrednosti indeksa *originPlace cuba*, stajao skraćeni naziv države **cu** propisan pomenutim kontrolisanim rečnikom.

Za kategoriju Language se ne vezuje ni jedan indeks već samo modifikator *languageAuthority*. Prakasa je da u ovom kontekst setu ne definiše konkretan jezik bibliotečkog zapisa već standard (autorizaciono telo) koji propisuje kako se definiše korišćeni jezik. Tako vrednosti *languageAuthority* (*rfc3066* (<http://www.ietf.org/rfc/rfc3066.txt>), *so639-2b* (http://www.loc.gov/standards/iso639-2/php/code_list.php) de fakto predstavljaju ime standarda. Podrazumevana vrednost modifikatora je ona koju server propiše. Modifikator *languageAuthority* se najčešće koristi kao dopuna *dc.language* indeksu, kao npr:

*dc.language=/languageAuthority=iso639-2b car AND
bib.subjectPlace=caribbean*

Kategorija *edition* sadrži samo indeks *edition*, koji opisuje ediciju (verziju) bibliografskog zapisa. Edicija može često da ukazuje na tržište za koje resurs bio namenjen, kao npr. *bib.edition=canadian*

Kategorija *Part* sadrži četiri indeksa *bib.volume* (broj volumena), *bib.issue* (broj izdanja), *bib.startPage* (početna strana), *bib.endPage* (poslednja stranica). Indeksi iz ove kategorije bliže opisuju i ukazuju na određeni deo bibliografskog objekta.

dc.title="neurology now" AND bib.volume=1 AND bib.issue=2

Poslednje navedena kategorija *Issuance* uključuje samo jedan indeks *issuance* koji opisuje vremenski period izdavanja bibliografskog objekta. Moguće vrednosti ovog indeksa su: *continuing* i *monograph*. Pomenute vrednosti su proizašle iz opšte podele bibliografskog sadržaja po vremenu na: vremenski omeđene publikacije (monografska građa) i vremenski neomeđene publikacije (periodika i kontinuirane).

dc.title="neurology now" AND bib.issuance=continuing

3.3.4. Bath Context Set

Bath Context Set je još jedan u nizu kontekst setova čiji je najčešći domen primene bibliografski podaci. Jedinstveni identifikator je <http://zing.z3950.org/cql/bath/2.0/>, dok je skraćeni naziv **bath**. Bath ne uvodi dodatne relacije i modifikatore. U tabeli 3.2 su navedeni svi indeksi *Bath Context Set* i njihov kratak opis.

Indeksi	Opis
bath.keyTitle	ključni naslov zapisa
bath.possessingInstitution	Institucija koja je poseduje bibliografski resurs na koji ukazuje zapis
bath.name	Ime autora zapisa
bath.personalName	Lično ime autora
bath.corporateName	Korporativno ime
bath.conferenceName	Ime događaja (konferencije) na kojoj je bibliografski resurs prezentovan
bath.uniformTitle	Naslov koji se dodeljuje resursima koji nemaju definisan naslov ili koji se pojavljuju pod više različitih naslova

Indeksi	Opis
bath.issn	Issn broj serijske publikacije
bath.geographicName	Geografski naziv za koji je vezan posmatrani bibliografski zapis
bath.notes	Beleške (dodatne informacije, najčešće nestrukturirane) o bibliografskom resursu)
bath.topicalSubject	Podaci o temama koji obrađuje bibliografski resurs
bath.genreForm	Žanr kome pripada bibliografski resurs

Tabela 3.2 Sadržaj *Bath Context Set*

3.4. SRU/W profili pretrage

Kod SRU/W protokola interoperabilnost između klijentske i serverske strane obezbeđena je upotrebom SRU/W profila. SRU/W profili su dokumenti kojima se opisuju funkcionalne mogućnosti i specifikacioni zahtevi servera. SRU/W profili objedinjuju postojeće kontekst setove i šeme zapisa podataka, navodeći listu podržanih elemenata iz pomenutih entiteta. SRU/W serveri nisu u obavezi da podrže sve elemente iz kontekst seta. Profili se ne smeju mešati sa kontekst setovima, šemama zapisa ili SRU ekstenzijama. Profili ne uvode „ništa novo“ već samo definišu listu mogućnosti koju server poseduje. Listu mogućnosti čine kontekst setovi ili njihovi delovi i/ili šeme zapisa. Svaki SRU/W profil karakteriše jedinstveni identifikator. Opis profila tj. podržanih elemenata kontekst setova na SRU/W serveru dobija se pozivom *Explain* servisa. Spisak svih registrovanih profila nalazi se na (Registered SRU/W Profiles, 2009). Autorizaciono telo koje registruje i održava pomenute profile je Kongresna biblioteka. U nastavku su opisani profili čiji su elementi relevantni za ovu disertaciju.

3.4.1. SRU/W Base Profile

Bazni profil propisuje minimalni skup zahteva koji treba da ispune klijent i/ili server kako bi se ostvarila interoperabilnost. Ovaj profil ne propisuje upotrebu niti jednog kontekst seta ili šeme po kojoj moraju biti predstavljeni podaci. Jedinstveni identifikator ovog profila je *info:srw/profile/1/base-profile-v1.1* a trenutna verzija je 1.1. Početni nivo interoperabilnosti podrazumeva da klijent/server podržava sledeće:

- SearchRetrieve operaciju (servis), gde i klijent i server moraju da imaju podršku za pomenutu operaciju
- Podrška za CQL upitni jezik, gde i klijent i server mora da podrži neki od tri nivoa složenosti CQL upita (poglavlje 3.2.1 - Sintaksa CQL upita)
- Podrška za *Explain* servis. Sa stanovišta servera on mora da obezbedi postojanje i funkcionisanje servisa, dok klijent mora biti sposoban da pošalje zahtev i prihvati odgovor od servera. Što se tiče samog *Explain* zapisa mora biti ispoštovano sledeće:
 - Element *serverInfo* mora biti prisutan i mora sadržati atribute *host*, *port* i *database*
 - U elementu *serverInfo* atribut *protocol* mora biti postavljen na vrednost SRU ili SRU/W. Atribut *version* mora biti postavljen na najvišu vrednost (verzija protokola koju podržava server).
 - *indexInfo* mora sadržati bar jedan prefiksa i minimalno jedan indeksni element namapiran na prefiks.

3.4.2. CQL Profile for Bibliographic Searching (Bib Profile)

Ovaj profil se koristi u kontekstu pretrage najrazličitijih vidova bibliotečkih podataka. Profil podrazumeva kombinaciju elemenata iz sledećih kontekst setova:

- *Bib Context Set*
- *DC (Dublin Core) Context Set*
- *CQL Context Set*

Prilikom korišćenje ovog profila preporuka je da se prvo upotrebljavaju indeksi, relacije i modifikatori iz opštijih kontekst setova (*CQL Context Set* i *Dublin Core Context Set*) pa da se tek nakon toga koristi specifičniji *Bib Context Set*. Tako npr. u *Bib Context Set* ne postoji opšti oblik naslova već samo specifični (*titleAbbreviated*, *titleUniform*, *titleTranslated*, *titleAlternative*, *titleSeries*), te je u tom slučaju za indeks pretrage bolje koristiti naslov iz *Dublin Core Context Set* (*dc.title*). *Bib Profile* možemo

posmatrati i iz ugla *Bib Context Set* koji je najveći po pitanju broja indeksa, relacija i modifikatora, i gde se u okviru njegovih kategorija dodaju indeksi i modifikatori iz preostala dva kontekst seta (*CQL Context Set* i *Dublin Core Context Set*). Tako se kategorija *Naslov* iz *Bib Context Set* proširuje indeksom *dc.title* iz *Dublin Core Context Set* i modifikatorom *lang* iz *CQL Context Set*. Kategorija *Name* se proširuje indeksima *dc.contributor*, *dc.creator* i *dc.publisher*. Kategorija *Subject* je proširena samo sa *dc.subject* iz *Dublin Core Context Set*. U klasama *Identifier*, *Resource Type*, *Format* i *Language* koje nisu imale indekse pridodati su sledeći indeksi respektivno šp kategorijama: *dc.identifier*, *dc.type*, *dc.format*, *dc.language* svi iz *Dublin Core Context Set*.

3.4.3. Bath Profile

Ime *Bath Profile* se inicijalno vezivalo za profil nastao pod ISO *Internationally Registered Profile* (IRP) za Z39.50 protokol (<http://zing.z3950.org/srw/bath/2.0/>). Profil dobija ime *Bath* po istoimenom mestu u Engleskoj, u kome se okupila radna grupa koja je radila na razvoju profila. Ovaj profil se najčešće koristi kada se žele napraviti preduslovi za interoperabilnost između bibliotečkih klijenata i servera koji komuniciraju putem SRU/W standarda. U sastav ovog profila ulaze indeksi iz *Dublin Core*, *CQL*, *Bath* i *Rec* kontekst setova navedeni u tabeli 3.3.

Aspekt Pretrage	Context Set	Index
Author, including Creator	Dublin Core	creator
Title	Dublin Core	title
Subject	Dublin Core	subject
Any	CQL	anywhere
Standard Identifier	Dublin Core	identifier
Date of Publication, including Date of Publication Range	Dublin Core	date
Key Title	Bath	keyTitle
Format/Type of Material	Dublin Core	format
Language	Dublin Core	language
Possessing Institution	Bath	possessingInstitution
Name	Bath	name

Aspekt Pretrage	Context Set	Index
Personal Name	Bath	personalName
Corporate Name	Bath	corporateName
Conference Name	Bath	conferenceName
Uniform Title	Bath	uniformTitle
ISSN	Bath	issn
Remote System Record Number	Rec	id
Geographic Name	Bath	geographicName
Notes	Bath	Notes
Topical Subject	Bath	topicalSubject
Genre/form Subject	Bath	genreForm

Tabela 3.3 Sadržaj *Bath* profila

U tabeli **3.3** se vidi da postoje izdvojeni aspekti(domene) pretrage i da se tu ne nalaze mapirani svi indeksi iz kontekst setova koji ulaze u sastav *Bath Profile*. Ukupan broj indeksa iz *Bath Profile* sa 96 je tako sužen na 21 indeks. Zaključeno je da se vrednosti ostalih indeksa mogu dobiti kombinacijom izdvojenih indeksa i korišćenjem relacija i modifikatora. U *Bath* profilu se korišćenje relacija i modifikatora može navesti i upotrebom imena **aspekta pretrage** iz tabele **3.3** nakon koje sledi **ime „tipa pretrage“** navedenog u tabeli **3.4**. U tabeli **3.4** se pored tipa pretrage nalaze i relacije i modifikatori koji se koriste kako bi se ostvario „tip pretrage“.

Tip pretrage	Relacija	Modifikatori
Keyword with Right Truncation	=	* na kraju terma pretrage
Exact Match	==	
First Words in Field	=	^ na početku terma
First Characters in Field	=	* na kraju terma pretrage i ^ na početku terma

Tabela 3.4 Tipovi pretrage u *Bath* profilu

Ostali tipovi pretrage koji se ne navedu sa eksplicitnim imenom pretražuju se kao da je u pitanju relacija = bez ijednog modifikatora. *Bath* profil propisuje sledeće izlazne formate podataka dobijenih pretragom: DC Schema for SRU (<http://www.loc.gov/standards/sruBob/resources/dc-schema.html>) i Marcxml (<http://www.loc.gov/standards/marcxml/schema/MARC21slim.xsd>).

3.4.4. CRIS Profil pretrage

Na osnovu rezultata analize (poglavlje 2) ustanovljeno je da pretraga naučno-istraživačkih podataka bez sumnje zahteva uvođenje niza standarda. Uvođenje standarda u sisteme pretrage bi pomoglo kako onima koji ih razvijaju (programeri, dizajneri) tako i onima koji ih koriste. Sa stanovišta

programera standardizacija bi doprinela njihovom bržem razvoju i lakšem održavanju, kao i pojednostavljenoj integraciji sa sličnim sistemima pretraga. Standardizacija bi korisnicima obezbedila jasno definisan pristup pretrazi koji bi se mogao primeniti i u budućnosti na sličnim sistemima. Sama standardizacija treba da donese neophodna pravila i koncepte koji bi bili opšte prihvaćeni.

Poslednjih godina, podaci naučno-istraživačke delatnosti često se stavljaju u kontekstu sa bibliotečkim podacima, iz razloga što bibliotečki sistemi skladište podatke o naučno-istraživačkim rezultatima. Samim tim, moguće je povezati bibliotečke standarde sa standardima za prikaz naučno-istraživačkih podataka, kao što je opisano u (Ivanović et al., 2011).

Bibliotečki sistemi baziraju se na precizno definisanim standardima, što je segment koji nedostaje ili nije u potpunosti razvijen u CRIS sistemima. Posebno je to slučaj sa standardima u domenu pretrage podataka. Standardi pretrage u bibliotečkim sistemima su jasno i precizno definisani, jer je njihovom uspostavljanju prethodio dugi niz godina verifikacije i razvoja. Iako postoji mnoštvo bibliotečkih standarda u domenu pretrage (Z39.50, SRU/W, OAI-PMH...), ne može se sa sigurnošću reći koji je najbolji i/ili najefikasniji. Odabir standarda zavisi najviše od konkretnih zahteva i potreba samih korisnika sistema. U disertaciji je predloženo da se sistem pretrage temelji na standardu SRU/W. SRU/W standard je odabran jer:

- dozvoljava definisanje servisa preko kojih je omogućena pretraga koja ne zavisi od interne reprezentacije podataka pretraženog sistema, već se ona zasniva na metapodacima.
- omogućuje definisanje i publikovanje rečnika metapodataka pretrage (SRU/W profili, kontekst setovi).
- omogućuje pretragu udaljenih sistema putem interneta.
- njegov upitni jezik pretrage (CQL) dozvoljava zadavanje izrazito složenih upita.
- CQL sintaksom je podržano kreiranje svih tipova pretrage identifikovanim i prikazanim u poglavlju 2 (logički upiti, približna pretraga, pretraga po udaljenosti, pretraga nedovršenih reči/rečenica i pretraga fraza).

- dozvoljava eksternim sistemima da odaberu format zapisa rezultata pretrage.
- ima pojednostavljenu implementaciju u odnosu na ostale protokole pretrage.
- koristi nove informacione tehnologije (WSDL, SOAP, HTTP i XML).

Uzevši u obzir rezultate analize (poglavlje 2), razmatrana je mogućnost odabira postojećeg ili kreiranja novog SRU/W profila koji bi svoju primenu našao u CRIS sistemima. U prvoj iteraciji pokušano je da za najčešće korišćene indekse (identifikovani u sekciji rezultati analize) pronađu ekvivalenti u postojećim SRU/W profilima i kontekst setovima.

Naime, analizirani su postojeći profili i njihovi kontekst setovi koji bi svoju primenu mogli da nađu u domenu pretrage podataka naučno-istraživačke delatnosti. Poseban akcenat analize je bio na Bath 2.0 profilu i Bib 1.0 profilu jer eksplicitno sadrže indekse koji se odnose na deo podataka naučno-istraživačke delatnosti. Oba profila sadrže Dublin core i CQL kontekst setove. Bath 2.0 pored navedenih sadrži bath i rec (Record Metadata Context Set, 2007) kontekst setove, dok Bib 1.0 ima i bib kontekst set. Analizom je utvrđeno da navedeni profili i njihovi specifični kontekst setovi sadrže mnoštvo različitih indeksa uglavnom pogodnih za opis bibliografskih podataka, što dovodi do zaključka da se pomenuti profili ne mogu direktno upotrebiti za opis svih entiteta u CRIS sistemima. Domen CRIS sistema pored bibliografskih podataka čine i podaci o osobama, institucijama, događajima, projektima ..., stoga je bilo potrebno definisati novi profil i kontekst set koji bi u potpunosti opisao sve relevantne podatke iz CRIS sistema.

Za potrebe sistema pretrage naučno-istraživačkih podataka definisan je novi profil pod nazivom CRIS. Predloženi identifikator novog profila je *info:srw/cql-context-set/1/cris-v1.0*. CRIS profil obuhvata postojeće kontekst setove *Dublin Core* i *CQL*, kao i novo kreirani **Cris** kontekst set. U tabeli 3.5 su navedeni indeksi iz odgovarajućih kontekst setova CRIS profila i opis semantike samih indeksa. Zapisi koji se pretražuju pomoću CRIS profila, odnose se na podatke iz naučno-istraživačke delatnosti kao što su: Institucije, Organizacije, Autori (Istraživači), Konferencije, Zbornik radova, Događaji, Radovi na konferencijama, Časopisi, Radovi u časopisima, Monografije,

Poglavlja u monografiji, Teze i Disertacije, Patenti i Proizvodi istraživanja. Predloženi profil ni na koji način ne uslovljava korisnike da u potpunosti implementiraju sve stavke iz navedenog profila. Svako može implementirati samo deo profila (indeksa) da bi ostao u skladu sa podacima iz sopstvenog sistema. Stoga sistemi koji žele da implementiraju predloženi profil dobijaju sve benefiti (funkcionalnosti) samog profila bez potrebe za izmenom arhitekture kao i strukture podataka samih sistema.

Predloženi identifikator Cris kontekst seta je *info:srw/cql-context-set/1/cris-v1.0*. Kratko ime koje se predlaže za definisanje prefiksa pomenutog kontekst seta je **cris**. Ključna reč **cris** koristi se za referenciranje indeksa i relacija konkretnog kontekst seta u okviru CQL upita. U tabeli 5.5 Indeksi CRIS profila su navedeni svi indeksi koji ulaze u sastav CRIS profila, sa imenom, oznakom kontekst seta kome pripadaju i njihovim konkretnim značenjem.

Index name	Context set	Description
allIndexes	CQL	Pretraga svih dostupnih indeksa.
anyIndexes	CQL	Pretraga skupa indeksa odabranog od strane tekst servera.
keywords	CQL	Ključne reči kojima se opisuju svi zapisi (istraživači, institucije, časopisi, radovi u časopisu, konferencije, zbornici radova konferencije, radovi na konferenciji, monografije, poglavlja monografija, knjige, poglavlja knjige, teze i disertacije, patenti, proizvodi i projekti ...).
title	Dublin core	Naslov i skraćeni oblik naslova za: publikaciju (rad u časopisu, rad na konferenciji, poglavlje monografije, poglavlje knjige, teze i disertacije), zbornik radova konferencije, monografiju, knjigu, časopis, projekat, patent i proizvod, instituciju ili istraživače.
creator	Dublin core	Lično ime istraživača koji je autor, editor ili recenzent publikacija, monografija, knjiga i zbornika radova konferencije. Lično ime istraživača koji je učestvovao u nastanku zapisa patent, proizvod i projekat.
subject	Dublin core	Ključne reči za sve zapise i oblast istraživanja za istraživača i instituciju.
description	Dublin core	Opis koji predstavlja apstrakt i/ili opšte napomene zapisa: publikacija, časopisa, konferencija, zbornika radova sa konferencija, knjiga, monografija, projekata, patenata i proizvoda.
publisher	Dublin core	Lično ime izdavača za zapis: časopis, zbornik radova sa konferencije, monografija, knjiga, patent i proizvod.
contributor	Dublin	Entitet koji je doprineo pri kreiranju zapisa. Lično ime istraživača

Index name	Context set	Description
	core	u funkciji editora za zbornik radova sa konferencije, knjigu, monografiju i časopis. Naziv institucije koja je doprinela stvaranju zapisa: publikacije, projekta, patenta i proizvoda. Lično ime istraživača koji je recenzent monografije. Institucija koja je izdavač zapisa: časopis, zbornik radova sa konferencije, monografija, patent i proizvod.
date	Dublin core	Datum publikovanja, nastanka ili osnivanja zapisa za sve zapise: publikacija, časopisa, konferencija, zbornika radova sa konferencije, monografije, knjiga, patenata, proizvoda, projekta, istraživača i institucija.
identifier	Dublin core	Identifikator resursa koji može biti URI, ISSN, ISBN, DOI i local (lokalni identifikator).
source	Dublin core	Naziv zapisa konferencija, zbornika radova sa konferencije, knjiga, časopis i monografija za radove u pomenutim entitetima.
language	Dublin core	Definiše jezik posmatranog zapisa.
coverage	Dublin core	Geografska lokacija (ime mesta i/ili države) za resurse institucije, istraživača i konferencije.
abstract	Cris	Deo sadržaja zapisa publikacije, patenta i projekta koji se odnosi na apstrakt.
firstAuthor	Cris	Lično ime istraživač koji je prvi autor publikacije.
type	Cris	Naziv naučno-istraživačkog zapisa (tip resursa) koji se pretražuje.
conference	Cris	Naziv konferencije za zapise: konferencija, zbornik radova sa konferencije i rad na konferenciji.
journal	Cris	Naziv časopisa za zapise časopis i rad u časopisu.
monograph	Cris	Naziv monografije za zapise monografija i poglavlje u monografiji.
institution	Cris	Naziv institucije za zapise: istraživač, institucija i projekat.
researchArea	Cris	Oblast istraživanja za resurse instituciju i istraživača.
researcherTitle	Cris	Titula/zvanje istraživača.
editor	Cris	Lično ime editora za zbornik radova konferencije, monografije, knjige i časopisa.
ISSN	Cris	ISSN identifikator
ISBN	Cris	ISBN identifikator
researcher	Cris	Lično ime istraživača za resurse istraživač i instituciju.
fullText	Cris	Celokupni sadržaja zapisa publikacije.

Tabela 5.5 Indeksi CRIS profila

CRIS profil za potrebe pretrage podržava sledeće relacije iz CQL kontekste seta: "=", "==", ">", ">=", "<", "<=", "<>", ">=", "any", "all", "adj" i "within". Modifikatori relacija podržani iz CQL kontekst seta su: "stem", "fuzzy", "locale", "word", "string", "number", "masked", "unmasked" i "within". Podržani su svi logički modifikatori iz CQL kontekst seta. Za potrebe "unit" logičkog modifikatora definisane su jedinice udaljenosti koje mogu biti neke od navedenih vrednosti: "word", "sentence" i "paragraph". Navedene relacije i modifikatori iz CQL kontekst seta bili su sasvim dovoljni da se u potpunosti definiše CRIS profil, i apsolutno nije bilo potrebe za dodavanjem/kreiranjem novih. Takođe, na osnovu zaključaka analize utvrđeno je da su postojeće relacije u CQL kontekst setu sasvim dovoljne da se na osnovu njih kreiraju i najzahtevniji upiti pretrage podataka naučno istraživačkog domena.

Radi boljeg razumevanja kako će indeksi iz CRIS profila mogu iskoristiti za pretragu naučno-istraživačkih zapisa, u nastavku su dati odgovarajući primeri. Prvi primer pokazaće kako se indeks *creator* mapira na elemente CERIF 1.5 modela podataka. Dok se drugi primer odnosi na mapiranje istoimenog indeksa na model podataka CRIS sistema Univerziteta u Novom Sadu. Indeks *creator* se koristi za pribavljanje zapisa koji u sebi sadrže istraživače u funkciji autora, editora, recenzenta ili participanta.

U prvom primeru pomoću indeksa *creator* odrediće se skup istraživača čiji entiteti *cfPersName* ili *cfPersName_Pers* zadovoljavaju vrednost indeksa. Sistem pretrage vratiće kao odgovor sve zapise entiteta *cfResPubl* (knjiga, poglavlje knjige, monografija, poglavlje monografije, časopis, rad u časopisu, zbornik radova konferencije, rad na konferenciji, teza i disetracija), *cfProj* (projekta), *cfResPat* (patenta) i *cfResProd* (proizvoda) koji sa pronađenim istraživačima grade specifičnu semantičku vezu. Semantička veza koja mora biti zadovoljena je definisana u entitetima *cfPers_ResPubl*, *cfProj_Pers*, *cfPers_ResPat*, *cfPers_ResProd* i obuhvatila je sledeću klasifikaciju:

- *Author, Editor* ili *Reviewer* za vezu *cfPers_ResPubl*
- *Participant* za vezu *cfProj_Pers*
- *Author* za vezu *cfPers_ResPat* ili *cfPers_ResProd*

Navedene klasifikacije `cfPers_ResPubl`, `cfProj_Pers`, `cfPers_ResPat` i `cfPers_ResProd` su apostrofirane, jer se one pojavljuju kao entiteti (indeksi) u analiziranim sistemima. CERIF model sadrži mnoštvo klasifikacija za postojeće entitete i relacije. Klasifikacije u CERIF modelu su opisane u CERIF rečniku (CERIF 1.5 Semantics, 2010), gde su one grupisane u pojedinačnim šemama. Npr. *Participant* predstavlja samo jednu od mogućih klasifikacija iz šeme *Person Project Engagements* (`cfClassificationSchemeName`) za entitet `cfProj_Pers`.

Ukoliko je neophodno da rezultati pretrage sadrže osobe koja su npr. koordinatori projekata, potrebno je da se prethodno navedeni podaci o ovoj ulozi i opišu zasebnim indeksom. Obzirom da CRIS profil nema ograničenja u dodavanju novih indeksa, a u cilju dobijanja semantički bogatije pretrage, moguće dodati neophodne indekse koji su u skladu sa CERIF semantičkim vokabularom (semantičkim šema i njihovim klasifikacijama). Tako se u CERIF za definisanje istraživača u ulozi koordinatora projekta koristi `cfProj_Pers`, *Person Project Engagements* klasifikaciona šema sa klasifikacijom *Coordinator*. Prethodno omogućava da kreiramo indeks *projectCoordinator* koji bi omogućio pretragu podataka koji se odnose samo na "osobe koji su svojstvu koordinatora projekta".

U drugom primeru vrednost indeksa `creator` mapiraće se na određena polja MARC 21 zapisa. Obzirom da je naveden indeks *creator*, iz entiteta `Marc21Record_Class` pribavljaju se identifikatori svih zapisa koji su klasifikovani kao: knjige, poglavlja knjige, monografije, poglavlja monografije, časopisi, radovi u časopisu, zbornici radova konferencije, radovi na konferenciji, teze i disertacije, projekti, patenti i proizvodi. Za svaki pribavljeni identifikator očitava se tačno jedan MARC 21 zapis uskladišten u entitetu `Marc21Record`. U odnosu na tip naučno-istraživačkog zapisa, indeks `creator` može biti mapiran na više različitih MARC 21 polja. Tako se za istraživače u funkciji *Editor* ili *Reviewer* posmatra MARC 21 700a (Added entry - Personal Name) za zapise tipa knjiga, časopisa, zbornik na konferenciji i monografija. U slučaju da je istraživač u funkciji *Author* posmatra se MARC 21 100a (Main entry - Personal Name) za zapise tipa: knjiga, poglavlje knjige, rad u časopisu, rad na konferenciji, monografija, poglavlje monografije, teza i disertacija, patent, proizvod i projekat.

4. Organizacija podataka unutar CRIS UNS sistema

Sistem za pretragu naučno-istraživačkih rezultata je nastao kao proširenje postojećeg sistema za unos istih (CRIS UNS). Stoga je neophodno da se prethodno opiše model koji se koristi za skladištenje podataka. Model je bio predmet istraživanja (Ivanovic, 2010) gde je i detaljno opisan.

Više informacija o arhitekturi i implementaciji CRIS UNS može se pronaći u radovima (Ivanović et al., 2010; Milosavljević et. al., 2011; Ivanovic et al., 2009; Ivanović and Milosavljevic, 2009; Penca and Nikolić, 2012; Surla et al., 2013). Automatska ekstrakcija metapodataka iz publikovanih rezultata skladištenih u ovom sistemu je opisana u radu (Kovačević et al., 2011). Komponenta za izveštavanje opisana je u radu (Dimić-Surla & Ivanović, 2012). Mogućnost razmene podataka sa drugim sistemima diskutovana je u radu (Ivanović, 2011a). CRIS UNS ontologije su tema radova (Ivanović, 2012, Dimić-Surla et al., 2012).

U radovima (Ivanović et al., 2010; D. Ivanovic et al., 2012) dato je proširenje CERIF modela podataka za vrednovanje publikovanih naučno-istraživačkih rezultata istraživača. Proširenje je bazirano na semantičkom sloju CERIF modela pomoću kojeg je moguće entitete i njihove veze klasifikovati po različitim klasifikacionim šemama. Ovaj model je verifikovan na pravilima koja su u skladu sa Pravilnikom o vrednovanju i kvantitativnom iskazivanju naučno-istraživačkih rezultata istraživača Univerziteta u Novom Sadu (Surla et al., 2012). Servis za vrednovanje naučnih rezultata objavljenih u časopisima je opisan u radovima (Nikolić et al., 2012; Nikolić et al., 2013; Nikolić et al., 2014) i dostupan je na adresi: <http://cris.uns.ac.rs/evaluation/evaluationStrucnoVeceTT.jsf>.

U okviru CRIS UNS implementirana je i digitalna biblioteka teza i disertacija koja je opisana u (L. Ivanovic et al., 2012; Ivanović et al., 2012b; Ivanović and Surla, 2012, Ivanovic et al., 2013; Ivanović, 2013). Kreirana digitalna biblioteka se može koristiti za potrebe različitih naučnih institucija, a verifikovana je na tezama i disertacijama Univerziteta u Novom Sadu

U ovom poglavlju će biti navedeni samo najvažniji segmenti iz pomenutog CRIS UNS modela koji se tiču sistema pretrage i opisan je proces indeksiranja podataka predstavljenih modelom.

4.1. CRIS UNS model podataka

U modelu podataka CRIS UNS sistema deo CERIF modela podataka koji je namenjen skladištenju podataka o publikovanim rezultatima je mapiran na model podataka MARC 21 formata.

4.1.1. CERIF

Digitalizacijom podataka naučno-istraživačke delatnosti nastalo je mnoštvo različitih sistema za skladištenje istih. Još na samim počecima korišćenja pomenutih sistema, uviđala se potreba za razmenom digitalnih zapisa između sistema koji sadrže slične podatke. Iz navedenog se zaključuje da je sistemima naučno-istraživačke delatnosti prevashodno bio potreban formalni opis semantike samih podataka, čime bi se smanjila njihova neodređenost i povećao stepen dostupnosti i distribucije. Jedan od oblika formalnog opisa koji bi se mogao iskoristiti za predstavu podataka naučno-istraživačke delatnosti je CERIF standard. CERIF je standard koji opisuje model podataka na bazi koga se mogu razmeniti podaci naučno-istraživačke delatnosti.

CERIF se u praksi često mapira na standarde koji takođe omogućuju prikaz podataka naučno-istraživačke delatnosti, kao što je to opisano u radu (Ivanović et al., 2011).

Entiteti CERIF modela podataka su podeljeni u sledeće grupe:

- *Core entities* - Ovu grupu entiteta čine bazni entiteti koji se odnose na: naučno-istraživačke projekte (*cfProj*), istraživače (*cfPers*) i organizacije/institucije (*cfOrgUnit*).
- *Result entities* – Entiteti iz obe grupe su namenjeni za skladištenje naučno-istraživački rezultata kao što su: publikacije (*cfResPubl*), produkti/proizvodi (*cfResProd*) i patenti (*cfResPat*).
- *2nd level entities* - U ovu grupu entiteta spadaju entiteti namenjeni skladištenju sadržaja značajnih za naučno-istraživačku delatnost, a koji ne pripadaju grupama entiteta *Result entities* i *Core entities*
- *Multilingual entities* – U ovu grupu entiteta spadaju entiteti koji su namenjeni skladištenju podataka o nekom entitetu iz grupe *Core*

entities, Result entitites, 2nd level entities или *Semantic layer entities* na proizvoljno mnogo različitih jezika.

- *Semantic layer entities* - CERIF model podataka ima semantički sloj pomoću kojeg entiteti i veze između entiteta mogu biti klasifikovane po raznim klasifikacionim šemama.
- *Link entitites* - Entiteti iz ove grupe su namenjeni povezivanju entiteta koji pripadaju grupama *Result entities, Core entitites, Semantic layer entities, 2nd level entities*.

4.1.2. MARC 21

MARC 21 format je standard koji propisuje format zapisa za skladištenje podataka značajnih za bibliotečke sisteme. Standard MARC 21 je definisan formatima zapisa sledećih vrsta podataka:

- MARC 21 format za bibliografske podatke (*MARC 21 Format for Bibliographic Data: Table of Contents (Network Development and MARC Standards Office, Library of Congress)* - <http://www.loc.gov/marc/bibliographic/ecbdhome.html>)
- MARC 21 format za normativne podatke (*MARC 21 Format for Authority Data: Table of Contents (Network Development and MARC Standards Office, Library of Congress)* - <http://www.loc.gov/marc/bibliographic/ecbdhome.html>)
- MARC 21 format za podatke o fondu odnosno lokacijske podatke (*MARC 21 Format for Holdings Data: Table of Contents (Network Development and MARC Standards Office, Library of Congress)* - <http://www.loc.gov/marc/holdings/echdhome.html>)
- MARC 21 format za klasifikacione podatke (*MARC 21 Format for Classification Data: Table of Contents (Network Development and MARC Standards Office, Library of Congress)* - <http://www.loc.gov/marc/classification/eccdhome.html>)
- MARC 21 format za informacije o zajednici (*MARC 21 Format for Community Information: Table of Contents (Network Development and MARC Standards Office, Library of Congress)* - <http://www.loc.gov/marc/community/eccihome.html>)

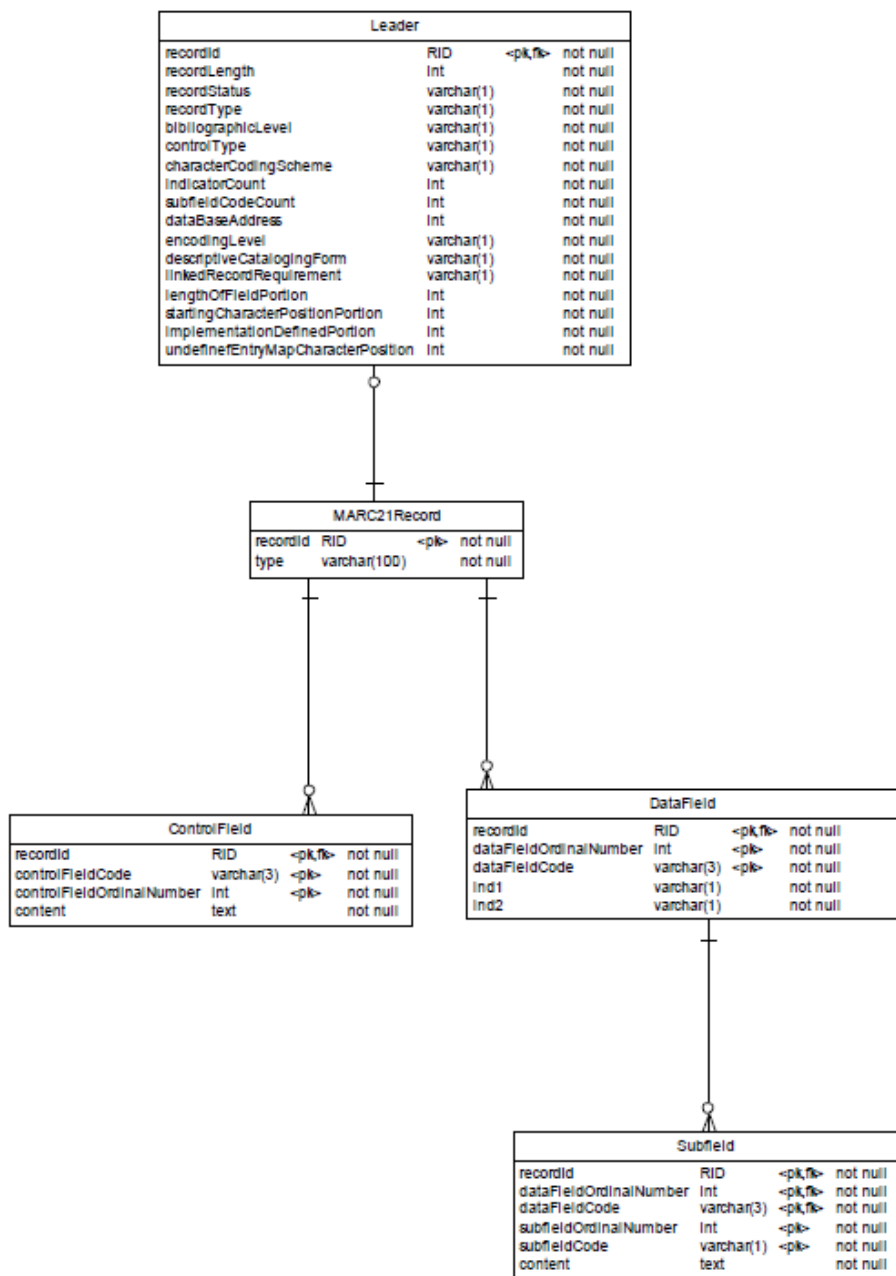
Prema MARC 21 standardu, zapis predstavlja kompoziciju tri elementa: strukture zapisa, oznake sadržaja i podataka koje zapis sadrži. Pod oznakom sadržaja podrazumevaju se nazivi polja, potpolja, šifarnici i ostale konvencije uvedene kako bi se podaci o publikacijama na jedinstven način mogli okarakterisati i identifikovati. Struktura MARC 21 zapisa sastoji se iz četiri dela, zaglavlja zapisa (Leader), direktorijuma (Directory), skupa kontrolnih polja (Control Fields) i skupa polja sa podacima (Data Fields). Zaglavlje zapisa javlja se kao prvi element zapisa i fiksne je veličine od 24 karaktera. Svaki podatak u zaglavlju zapisa je šifrirana vrednost na odgovarajućoj poziciji karaktera i nosi informaciju neophodnu za obradu zapisa. Direktorijum sadrži podatke o oznaci, dužini i početnoj lokaciji za svako kontrolno polje i za svako polje sa podacima koje se javlja u okviru zapisa. Svako kontrolno polje i svako polje sa podacima ima oznaku polja. Kontrolna polja mogu sadržati ili jedinstven podatak o publikaciji ili skup podataka fiksne dužine pri čemu se svaki pojedinačni podatak javlja na tačno određenoj poziciji. Polja sa podacima sadrže najviše dva indikatora. Uloga indikatora je da omoguće precizniju interpretaciju ili bliže odrede podatke koji se javljaju kao sadržaj pripadajućeg polja sa podacima. Vrednosti indikatora interpretiraju se odvojeno, odnosno dva indikatora u polju nemaju međusobne veze. Polja sa podacima sadrže skup potpolja. Potpolja sadrže oznaku potpolja i sadržaj. Za oznaku potpolja u zapisu rezervisana su dva karaktera, prvi je oznaka dolar (\$), a drugi oznaka potpolja koja može biti malo latinično slovo ili numerički simbol. Sadržaji potpolja u okviru zapisa predstavljaju osnovne nosioce podataka o publikaciji na koju se zapis odnosi.

4.1.3. CERIF kompatibilni model podataka baziran na MARC 21 formatu

U ovom modelu podataka svi atributi CERIF entiteta koji se odnose na publikacije, proizvode, patente, osobe, projekte, događaje i određeni skup atributa entiteta koji se odnose na organizacione jedinice zamenjeni su modelom MARC 21 formata. Model podataka MARC 21 formata je prikazan na slici 4.1.

U modelu podataka prikazanom na slici 4.1 nema entiteta koji predstavlja direktorijum zapisa jer se on može generisati na osnovu vrednosti upisanih u zaglavlju, kontrolnim poljima, poljima sa podacima i potpoljima.

Entitet `MARC21Record` ima samo jedan atribut `recordId` u kome se nalazi jedinstveni identifikator MARC 21 zapisa. Entitet `Leader` predstavlja zaglavlje, entitet `ControlField` predstavlja kontrolno polje, entitet `DataField` predstavlja polje sa podacima, a entitet `Subfield` predstavlja potpolje zapisa. Svaki podatak u zaglavlju zapisa koji je šifrirana vrednost na odgovarajućoj poziciji karaktera je modeliran posebnim atributom entiteta `Leader`. Po MARC 21 formatu kontrolna polja i polja sa podacima mogu imati osobinu ponovljivosti u okviru zapisa. Atribut `controlFieldOrdinalNumber` entiteta `ControlField` predstavlja redni broj kontrolnog polja u nizu kontrolnih polja sa istim imenom. Atribut `dataFieldOrdinalNumber` entiteta `DataField` predstavlja redni broj polja sa podacima u nizu polja sa podacima sa istim imenom. Takođe i potpolja u okviru polja sa podacima mogu biti ponovljiva. Atribut `subfieldOrdinalNumber` entiteta `Subfield` predstavlja redni broj potpolja u nizu potpolja istog imena koji pripadaju istom polju sa podacima. Entiteti `ControlField` i `Subfield` u atributima `content` sadrže podatke o bibliografskoj građi. Entitet `MARC21Record` sadrži atribut `type`. Ovaj atribut služi da se ustanovi pojava kog entiteta CERIF modela podataka je skladištena u MARC 21 formatu. Atribut `type` može imati sledeće vrednosti: `project`, `person`, `organizationUnit`, `resultPublication`, `resultProduct`, `resultPatent`, `event`.



Slika 4.1 MARC21 model

4.2. Indeksiranje podataka

Da bi podaci skladišteni u CRIS UNS sistemu lako pretraživali, bilo je neophodno obaviti niz operacija nad samim podacima. Jedna od ključnih je

svakako bila i indeksiranje postojećih naučno-istraživačkih podataka. Postoji mnoštvo definicija indeksiranja u oblasti pretrage podataka. U nastavku će biti navedene neke od najpoznatijih i najznačajnijih. Tako se u literaturi često navodi da **indeksiranje** predstavlja **proces sakupljanja, parsiranja i skladištenja** podataka u cilju izgradnje efikasnog sistema pretrage. Bez indeksiranja sistem bi morao da prolazi kroz svaki pojedinačan dokument u skladištu što bi zahtevalo mnogo vremena i računarskih resursa. Tako se npr. 1000 indeksiranih dokumenata može pretražiti u milisekundi, dok pretraga ne obrađenih 1000 dokumenta može trajati satima, što predstavlja neprihvatljivo vreme za onog ko očekuje rezultate pretrage.

U (Baeza-Yates, 1999) se navodi da je indeksiranje proces konstrukcije niza ključnih reči (indeksa), koji inicira potreba korisnika za pretragom.

Po (Manning et al., 2008) indeksiranje predstavlja operaciju (proces) konstrukcije invertovanog indeksa, koji predstavlja de fakto najčešće korišćenu vrstu indeksa. U (McCandless, 2010) se pod pojmom indeksiranje podrazumeva nezaobilazan proces kako bi se obezbedila efikasna pretraga za korisnike.

Samo indeksiranje ponajviše zavisi od vrste i tipa podataka koji se indeksiraju. Indeksiranje podataka u sistemu pretrage naučno-istraživačkih podataka je izvršeno korišćenjem *Lucene* biblioteke. U sledećoj literaturi (Konchady, 2008; Morville, 2010; Gospodnetić, 2005; Moczar, 2014; White, 2013; Richardson, 2004; Carnell et al., 2003) se navodi kocepti i moguće upotrebe *Lucene* biblioteke. Razlog za odabir pomenute biblioteke leži u činjenici da je čitav sistem za unos naučno-istraživačkih podataka implementiran tehnologijama otvorenog koda baziranih na Java programskom jeziku kao i dobra iskustva iz drugih sistema prilikom korišćenja ove biblioteke.

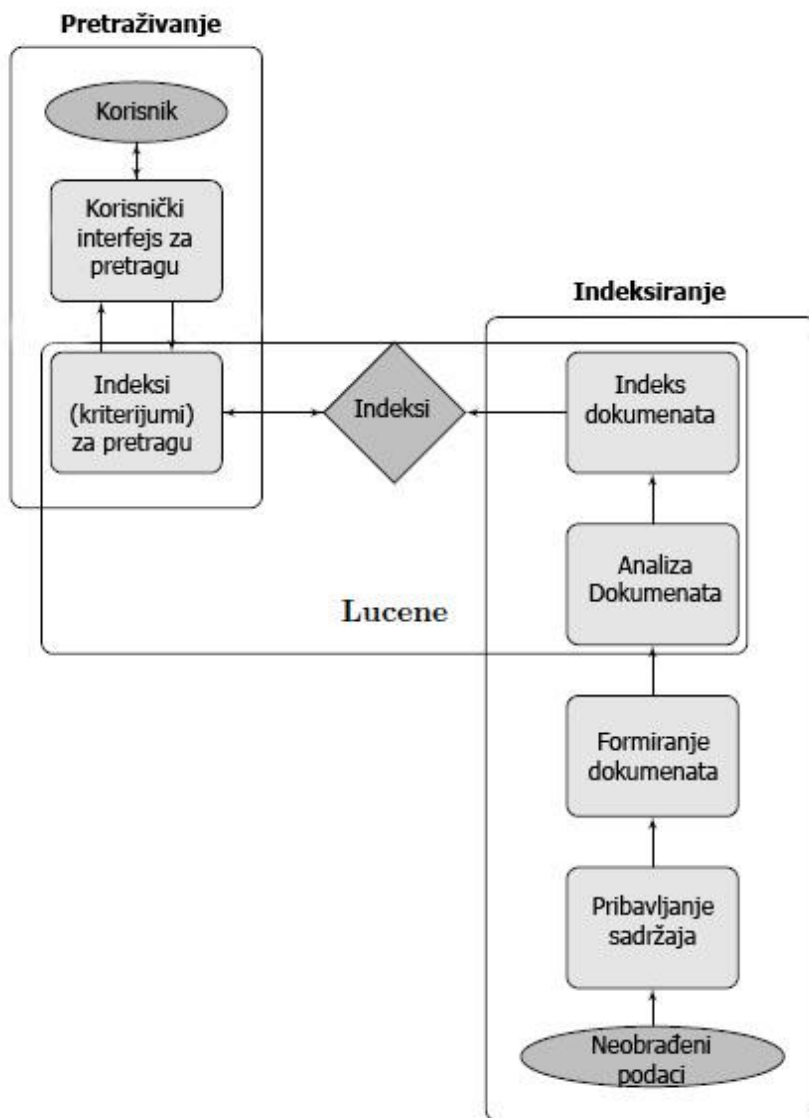
4.2.1. Lucene

Lucene je softverska biblioteka za pronalaženje informacija koja je napisana u Javi. Besplatna je i može se preuzeti sa (*Lucene Download* <http://lucene.apache.org/core/downloads.html>). Ova biblioteka obezbeđuje tzv. *Application Programming Interface* (API) za potpuno tekstualno indeksiranje i pronalaženje. Ova softverska biblioteka dostigla je visoku popularnost kod korisnika zbog svoje jednostavnosti. Jednostavnost se pre

svoga oglada u lakoći korišćenja API, gde korisnici ne moraju u potpunosti da poznaju sve teorijske osnove iz domena pretraživanja i indeksiranja. *Lucene* pomaže programerima da kod funkcija indeksiranja i pretraživanja pretvore bilo koji tip podatka u pretraživu tekstualnu reprezentaciju. Podaci se indeksiraju u specifičnu strukturu indeksnog fajla i pretražuju se po zadatom upitu korisnika. Samo logika aplikacije koja se odnosi na upravljanje indeksiranjem, pretraživanjem i prihvatanjem upita od korisnika ostaje u nadležnosti programera.

4.2.1.1. Lucene Arhitektura

Na slici **4.2** je prikazana arhitektura aplikacije za pretragu zasnovana na Lucene biblioteci. U okviru arhitekture uokviren je deo koji obavlja Lucene biblioteka. Sekvenca (redosled) koraka (akcija) kod aplikacije za pretragu je bitan. Radi lakšeg razumevanja čitavog procesa sistema za pretragu informacija sve korake grupišemo u dve kategorije. Prva četiri koraka, od prikupljanja podataka do indeksiranja dokumenata su deo segmenta **Indeksiranje**, dok su koraci od prosleđivanja upita do prezentovanja rezultata deo pretrage informacija (**Pretraživanje**).



Slika 4.2 Arhitektura aplikacije za pretragu bazirana na Lucene

Prvi korak u indeksiranju je *Pribavljanje sadržaja*. Za ovaj korak su najčešće zadužene eksterne aplikacije kao što su “pauci” ili “kravleri” (Levene and Poulouvasilis, 2004). Mada sadržaj mogu da generišu i korisnici koji unose vrednosti metapodataka ili npr. vrše *upload* datoteka. Postupak je jednostavan ukoliko je sadržaj već tekstualne prirode. U suprotnom, neophodno je angažovati posebne parsere koji bi ekstrahovali tekstualni sadržaj iz različitih tipova dokumenata.

Sledeći korak je *Formiranje dokumenata* od prikupljenog sadržaja. Pre indeksiranja, neophodno je transformisati sadržaj u tzv. atomičke jedinice (dokumente) kojima operira sistem za pretragu. Dokumenti obično sadrže određena imenovana polja sa vrednostima kao što su npr. naslov, apstrakt, autor, URL. Kreiranje adekvatnih dokumenata i polja je esencijalna stvar za pretraživanje podataka

Tekstualna polja se ne mogu direktno indeksirati, već je nužno da se tekst dekomponuje (izparsira) u niz bazičnih struktura pod nazivom tokeni. Ovaj postupak se dešava prilikom procesa analize dokumenta. Grubo bi se svaki token mogao okarakterisati kao reč iz nekog jezika. Analizator određuje kakvi će se tokeni smeštati kao vrednosti polja u dokumentu. Glavni koraci kod analize su svođenje reči na osnovni oblik, uklanjanje opštih reči, provera sinonima, ponderisanje dokumenata. Detaljnije o procesu analize u poglavlju **4.2.1.5**. Segment **Pretraživanje** sa slike **4.2** opisano je u poglavlju **5** Pretraživanje podataka unutar CRIS sistema

4.2.1.2. Lucene indeksiranje

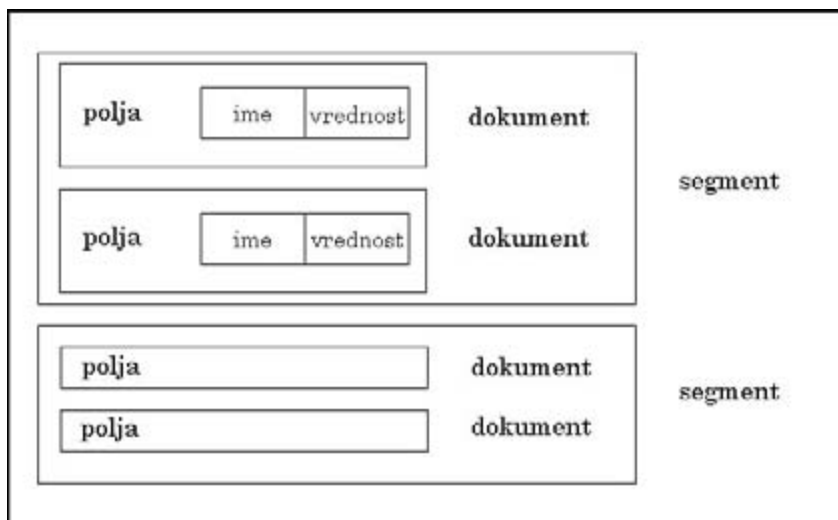
Indeksiranje je esencijalni deo (komponenta) svakog sistema pretrage. Jedan od glavnih koncepata kod Lucene API funkcija je indeksiranje. Indeksiranje je pretvaranje bilo kog tipa podataka u format pretraživog indeksa, gde pretvaranje defakto obavljaju analizatori. Prvo se dokumenti obrađuju tako da se sav nekoristan tekst kao što su tzv. stop reči, sufiksi i prefiksi odbacuju. Detaljan opis procesa analize dat je kasnije u posebnom odeljku. Kao rezultat procesa indeksiranja kreira se indeks.

U nastavku je ovog odeljka stavljen akcenat na **Struktura indeksa u Lucene**. Fundamentalni koncepti Lucene su indeksi, dokumenti, polja i termovi. Između pomenutih entiteta uspostavljena je sledeća hijerarhija:

- Indeksi sadrže sekvencu dokumenata
- Više dokumenata može biti organizovano u segment
- Dokument je niz polja
- Polje je imenovana sekvenca termova
- Term je string

Dokument predstavlja osnovnu jedinicu kod indeksiranja u Lucene. Sa stanovišta implementacije Lucene indeks se sastoji od instanci Lucene klase

Document koje definišu indeksne dokumente. Svaki dokument sadrži polja (Fields) koja se sastoje od parova ime i vrednost. Vrednosti su termini koji će biti predmet pretrage. Primer indeksa dat je na slici 4.3.



Slika 4.3 Struktura Lucene indeksa

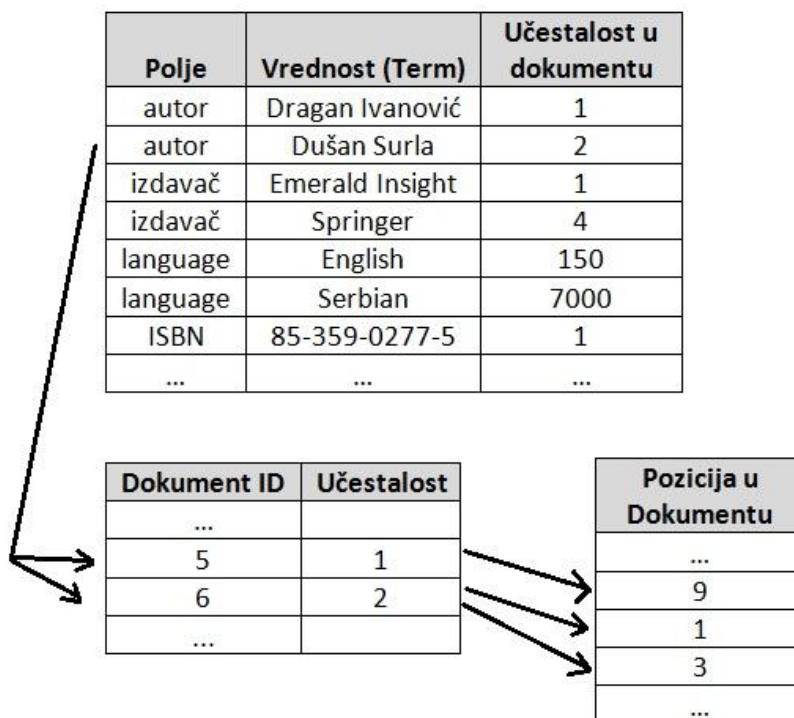
Kao što se vidi sa prethodne slike, Lucene indeks sadrži više segmenata. Segmenti su posebne strukture koje Lucene koristi prilikom indeksiranja, kako bi se ostvarile maksimalne performanse. Usled toga svaki segment ima u sebi smešteno više dokumenata. Dokumenti se sastoje od indeksiranih polja. Kao što je već rečeno, polja sadrže najmanje delove u strukturi indeksa koji su parovi ime i vrednost. Ova polja se koriste za izračunavanje težinskih faktora (pondera) i rangiranje rezultata pretrage.

Lucene često upoređuju sa bazama podataka jer može da skladišti i pribavlja podatke. Međutim, postoje velike razlike između pomenutih koncepata. Prva je svakako „fleksibilna” šema podataka kod Lucene. U bazama podataka, svaka tabela je predstavljena šemom koja je definisana imenima atributa (kolona), domena kao i različitim ograničenjima. Podaci koji nisu u skladu sa strukturom tabele ne mogu biti pohranjeni u istu. Zahvaljujući fleksibilnoj šemi, pojedinačni indeks u Lucene može da sadrži dokumente najrazličitijeg sadržaja. Druga suštinska razlika između Lucene i Baza podataka je da Lucene prilikom formiranja indeksa ne dozvoljava rekurzije i ugnježdavanje. Dok kod relacionih baza podataka se omogućuje lako povezivanje tabela preko primarnih i stranih ključeva. Potencijalne rekurzije i veze se moraju

ukloniti prilikom kreiranja indeksa, što može da dovede do višestruke replikacije pojedinih informacija.

4.2.1.3. Invertovani indeks

Struktura Lucene indeksa je poznata kao invertovani indeks. Invertovani indeks znači da se sadržaj dokumenata analizira i da se važni termini indeksiraju kao parovi ime polja i vrednost. Invertovani indeks nastaje kada Lucene skenira čitavu kolekciju dokumenata i izdvaja jedinstvene termine koji će biti predmet pretrage. Pomenuti termini bi se mogli smatrati kao neki vid rečnika. Za svaki entitet u rečnicima se formira tzv. „lista pojava”. U listi pojava se evidentiraju dokumenti u kojima se sreće određeni term. Invertovani indeks je koncept sličan indeksu u knjigama. Knjiški indeks sadrži reči i broj strana na kojima se te reči nalaze. Primer struktura invertovanog data je na slici 4.4.



Slika 4.4 Struktura Lucene invertovanog indeksa

Prilikom kreiranja invertovanog indeksa Lucene čuva i sve neophodne informacije za implementaciju *Vector Space Model*. (Luhn, 1957). Pomenuti

model podrazumeva da se u listi pojava doda i automatski ili manuelno definisana težina za termine u dokumentu. Težine se koriste prilikom izračunavanja mere relevantnosti nekog upita. Model zahteva izračunavanje broja pojavljivanja svakog terma u dokumentu. Najčešće implementacije ovog modela skladište i mesta (pozicije) pojavljivanja svakog terma u dokumentu. Pozicija terma je vrlo značajna kako bi se omogućila blizinska pretraga i pretraga fraza.

Svako polje sadrži jedan ili više termova koji pokazuju na odgovarajuće dokumente u repozitorijumu dokumenata. Invertovani indeks olakšava pronalaženje dokumenata u sistemu i koristi ga jezgro sistema za pretraživanje. Stoga se dokumenti traže u poljima i njihovim vrednostima.

4.2.1.4. Lucene API

Kod procesa indeksiranja uključene su osnovne Lucene klase:

- `IndexWriter`
- `Analyzer`
- `Directory`
- `Document`
- `Field`

`IndexWriter` je glavna klasa Lucene operacije indeksiranja. Ona kreira inicijalni indeksni fajl za zadatu putanju na računaru. Kao parametar prihvata putanju, tip analizatora i bulov parametar (ako je vrednost bulovog parametra `true`, tada se kreira novi indeks i brišu se prethodni indeksi, a ako je `false`, tada ga dodaje postojećem indeksu na toj putanji). `IndexWriter` je klasa koja ima samo pristup za pisanje u indeks i koristeći svoje metode korisnici mogu dodavati dokumente u indeks u cilju kreiranja indeksa za pretraživanje.

Klasa `Analyzer` implementira pretprocesiranje teksta, odnosno pripremu teksta za indeksiranje. Ona analizira dokumente i odbacuje tekst koji nije koristan u aplikaciji pretraživanja. Lucene poseduje više različitih vrsta analizatora. Oni su detaljnije opisani u poglavlju **4.2.1.6**.

Klasa `Directory` pokazuje mesto gde je indeks kreiran. Ona ima dve podklase, `FSDirectory` i `RAMDirectory`. Programer u skladu sa svojim

potrebama odlučuje koju će koristiti. `FSDirectory` egzistira u sklopu fajl sistema, a `RAMDirectory` egzistira u operativnoj memoriji računara. Očigledno je da `RAMDirectory` ima prednost u odnosu na `FSDirectory` jer je njen indeks u operativnoj memoriji i time je brže indeksiranje i pretraživanje dokumenata u indeksu nego kada se radi sa diskom. Međutim, kada se završi pretraživanje ili izađe iz aplikacije, indeks u operativnoj memoriji će biti obrisan. Prema tome, važno je da se indeks čuva na disku ukoliko želimo da se ponovo koristi u budućnosti.

Klasa `Document` reprezentuje jedan dokument u indeksu koji može biti rezultat pretrage. Sastoji se od polja koja čuvaju parove ime, vrednost. Polja su zapravo meta podaci koji se odnose na dokument. Kod Lucene biblioteke se meta podaci kao što su autor, naslov, datum kreiranja se respektivno smeštaju kao vrednosti polja u dokumentu. Takođe, u cilju indeksiranja bilo kog dokumenta, dokument mora biti u tekstualnom formatu ili pretvoriv u tekst.

Najznačajnija klasa za indeksiranje je `Field`. Ona reprezentuje osnovnu strukturu polja u indeksu. Svako od polja prikazuje informacije o dokumentima koji su sa njima povezani i ona se pretražuju u toku operacije pretrage indeksa. Polja su veoma značajna obzirom da ona sadrže konkretne vrednosti koje se indeksiraju. Postoji mnoštvo parametara koji se mogu specificirati u *Lucene* prilikom dodavanja polja u indeks. Najznačajnije su svakako opcije koje obezbeđuju pretraživost i dostupnost polja. Opcije u `Field.Index.*` definišu kako će se pretraživati vrednosti polja u invertovanom indeksu. Dok opcije u `Field.Store.*` definišu da li će konkretna vrednost polja biti sačuvana ili ne. Čuvanje vrednosti polja povećava veličinu indeksa. Pomenute opcije se mogu kombinovati na najrazličitije načine. U tabeli 4.1 se navode najkorišćenije kombinacije pomenutih opcija, kao njihova moguća upotreba.

Field.Index.*	Field.Store.*	Primeri korišćenja
NOT_ANALYZED	YES	Identifikatori (imena fajlova, primarni ključevi), telefonski broj i JMBG, URL, lična imena i datumi.
ANALYZED	YES	Naslovi dokumenata, apstrakt dokumenta.
ANALYZED	NO	Sadržaj (telo) dokumenata.

Field.Index.*	Filed.Store.*	Primeri korišćenja
NO	YES	Tip dokumenta, primarni ključ iz baze (u slučaju kada ne treba da se koristi za pretragu).
NOT_ANALYZED	NO	Skrivene ključne reči.

Tabel 4.1 Struktura Lucene invertovanog indeksa

Index.ANALYZED - podrazumeva korišćenje analizatora za dekomponovanje vrednosti polja u niz pretraživih tokena, što podrazumeva predprocesiranje teksta, prebacivanje velikih u mala slova, itd.

Index.NOT_ANALYZED - indeksira polje bez prethodne analize. Čitavo polje se tretira kao jedan token. Ova opcija se koristi kod polja čija vrednost treba da ostane ista kao o dokumentu.

NO - ukazuje da nije moguće izvršiti pretragu polja sa ovom opcijom. Podrazumeva polja čiji sadržaj ne treba da bude dostupan (pretraživ) za korisnike.

STORE.YES - ukazuje da vrednost iz polja treba da se skladišti. Ova opcija se koristi ako će se skladišti vrednosti iz dokumenta koji ne iziskuju puno memorijskog prostora kao što su npr. naslovi dokumenata.

STORE.NO - korišćenjem ove opcije se ne skladišti vrednost polja. Ova opcija se vrlo često uparuje sa Index.ANALYZED, jer se tada najčešće indeksiraju obimna tekst polja koja se ne moraju pribaviti u originalnom obliku, kao što su npr. sadržaji (tela) veb stranica i slično.

4.2.1.5. Indeksna Analiza

Analiza je proces pretvaranja tekstova dokumenata u osnovne i „indeksibilne“ termine. Ovde se izvode koraci predprocesiranja teksta u koje spadaju svođenje reči na osnovni oblik, odbacivanje opštih i nekorisnih reči, pretvaranje ispisa teksta u ispis malim slovima, stemming i/ili lematizaciju. Za svaku vrstu reči moguće je inicijalno odrediti kanonski, polazni oblik - lemu. Lema neke imenice je njen nominativ jednine, a lema glagola je infinitiv. Na primer, neki od oblika leme hodati su: hodam, hodao, hodamo, itd. Ponekad nije moguće jednoznačno odrediti lemu neke reči. Na primer, reč vode ima

sledeće leme: vod, voda, voditi. Postupak nalaženja leme zovemo lematizacija.

Lucene izvršava analizu prilikom indeksiranja ili prilikom obrade upita za pretragu. Ni jedan tekst server, pa ni *Lucene*, ne indeksira tekstualni sadržaj bez prethodnog raščlanjivanja teksta u niz individualnih bazičnih jedinica pod nazivom tokeni, nakon čega se oni dodatno obrađuju. Token se najčešće poistovećuje sa osnovnom jedinicom iz svakog jezika, a to je reč. Niz obrađenih tokena (termovi) je proizvod procesa analize. Kad ukombinujemo termine sa imenom polja za koji su vezani dobijamo stukturalne termine (Manning et al., 2008). Strukturalni term se može smatrati fundamentalnom indeksnom reprezentacijom originalnog teksta iz dokumenta.

Analizator ima ključnu ulogu u procesu analize, jer određuje strukturu i oblik tokena u tekstualnim poljima. Analizator treba da se pozabavi stvarima poput velikih i malih slova, složenica, množine i jednine imenica. Analizator je nadležan za niz operacija u procesu analize, od kojih su najznačajnije: ekstrakcija reči, eliminisanje interpunkcije, uklanjanje akcenata, prevođenje u mala slova, uklanjanje učestalih reči (predloga, veznika ...), stemovanje), zamena reči osnovnim oblikom iste (lematizacija). Čitav niz pomenutih operacija se neretko naziva proces analiza (McCandless, 2010) .

Upotreba adekvatnog analizatora je ključan faktor prilikom procesa indeksiranja. Jedan od bitnih faktora za odabir analizatora je jezik tekstualnih podataka. Drugi bitan faktor može biti različita terminologija zbog činjenice da podaci pripadaju nekom domenu. Ne postoji globalni analizator koji bi bio adekvatan prilikom indeksiranja bilo koje vrste i tipa podataka. Postoji mnogo raspoloživih analizatora u *Lucene*, međutim ukoliko je potrebno, moguće je relativno jednostavno kreirati i sopstveni analizator. Sam proces analizatora se odvija u pozadini i nakon izvršene analize se dobijeni termovi direktno smeštaju u indeks.

4.2.1.6. Ugrađeni analizatori

Analizatori se razlikuju po broju operacija koje primenjuju nad incijalnim tekstom. Koji god analizator da se odabere treba imati na umu koje će termine odabrani analizator kao rezultat da generiše. U nastavku će biti ukratko opisani pet najkorišćenijih analizatora iz *Lucene* biblioteke.

`WhitespaceAnalyzer`

Ovaj analizator obavlja samo jednu operaciju analize koja se može naslutiti iz samog imena ovog analizatora. Naime, `Whitespace Analyzer` samo dekomponuje ulazni tekst u odnosu na „bele“ karaktere (razmake).

`SimpleAnalyzer`

Sličan je `Whitespace Analyzer` stin što se splitovanje teksta vrši na tzv. „netekstualnih karaktera“ (*,-,?). Nakon toga se izvršava pretvaranje svih velikih slova u mala.

`StopAnalyzer`

Ima svu funkcionalnost kao i `Simple Analyzer`, stin što odbacuje opšte i učestale reči. Treba imati u vidu da se u ovom analizatoru za opšte i učestale reči podrazumevaju one koje potiču iz engleskog jezika (the, a, an ...), ali se lista opštih i učestalih reči može konfigurirati.

`StandardAnalyzer`

Predstavlja najsofisticiraniji predefinisani *Lucene* analizator. `Standard Analyzer` obavlja sve operacije kao i `Stop Analyzer` ali i dodaje niz novih. Ovaj analizator poseduje mogućnost da prilikom tokenizacije prepozna konstrukcije kao što su e-mail ili IP adrese.

Način funkcionisanja pomenutih analizatora je prikazan i u sledećem primeru. Između znakova " " je naveden dokument za analizu, dok su između [] navedeni rezultujuć termovi.

Primer za analizu

"Dušan Surla, PMF-Novu Sad, a email je surla@uns.ac.rs"

Rezultati pojedinačnih analizatora

`WhitespaceAnalyzer`:

[Dušan] [Surla,] [PMF-Novu] [Sad,] [a] [email] [je] [surla@uns.ac.rs]

`SimpleAnalyzer`

[dušan] [surla] [pmf] [novu] [sad] [a] [email] [je] [surla] [uns] [ac] [rs]

`StopAnalyzer`

[dušan] [surla] [pmf] [novu] [sad] [email] [je] [surla] [uns] [ac] [rs]

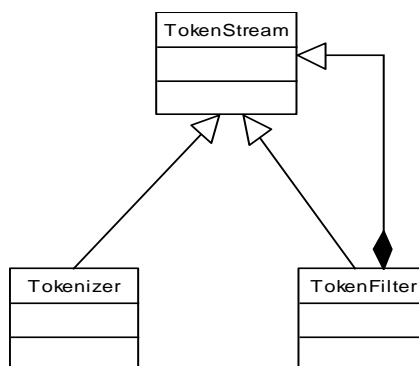
`StandardAnalyzer`

4.2.1.7. Sopstveni Analizator

Iz prethodnog izlaganja je jasno da će se često prilikom indeksiranja nametnuti potreba za kreiranjem sopstvenog (*custom*) analizatora. Proces kreiranja sopstvenog analizatora počinje nasleđivanjem bazične apstraktne klase `Analyzer`. Preduslov za kreiranje sopstvenog analizatora je implementacije metode :

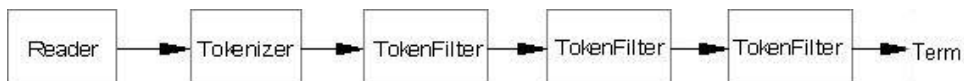
```
public TokenStream tokenStream(String fieldName, Reader reader).
```

Klasa `TokenStream` se nakon toga koristi za iteriranje kroz sve tokene. Za ovu klasu su vezane dve klase naslednice `Tokenizer` i `TokenFilter`. (slika 4.5).



Slika 4.5 Hijerarhija klase koje učestvuju u produkciji tokena

Sa slike se uočava veza kompozicije između klase `TokenStream` i `TokenFilter`, te je očigledno da `TokenFilter` enkapsulira drugi `TokenStream`, koji naravno može biti i drugi `TokenFilter`. Klasa `TokenFilter` služi pre svega za modifikaciju dobijenih tokena. Tako npr. primenom `LowerCaseFilter` sva slova u tokenu postaju mala slova ili `StopFilter` koji uklanja sve reči koje su definisane kao nepotrebne u ovom filteru. Klasa `Tokenizer` čita karaktere iz java čitača (`java.io.Reader`) i kreira tokene. Sam analizator svoj proces započinje klasom `Tokenizer`, gde se nakon toga nadovezuje proizvoljan broj željenih filtera (`TokenFilter`). Na slici 4.6 je prikazan tzv. "lanac analize teksta".



Slika 4.6 Lanac analize teksta

4.3. Proces indeksiranja u sistemu pretrage CRIS UNS

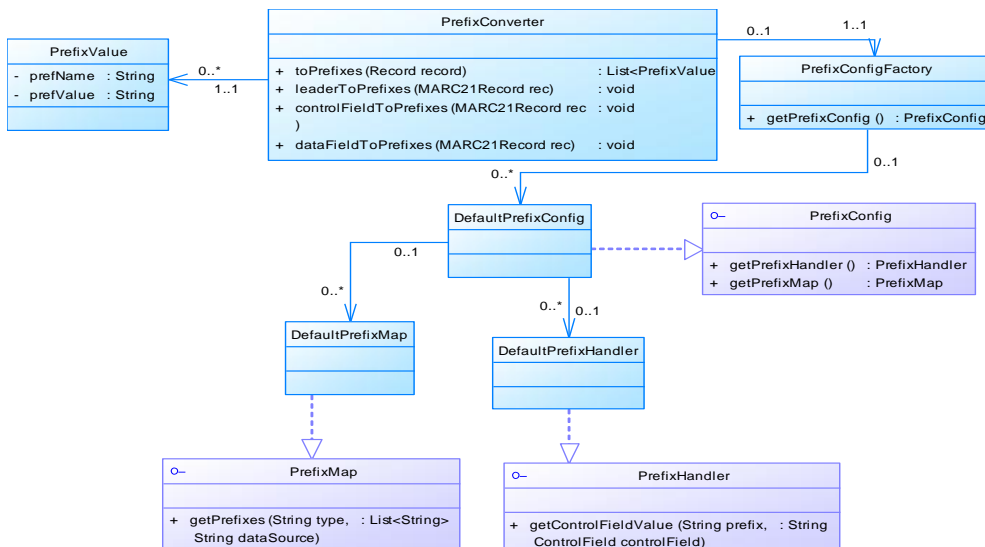
Iz prethodnih poglavlja evidentno je da je preduslov za pretragu podataka uz pomoć *Lucene* biblioteke konstruisanje struktura *Lucene* dokumenta. U konkretnom slučaju se moraju naučno-istraživački podaci iz CRIS UNS sistema predstavljeni CERIF kompatibilnim modelom skladišteni u MySQL bazi podataka, transformisati u odgovarajući *Lucene* dokument. Pomenuta transformacija se sastoji od niza koraka.

Kako su podaci smešteni kao zapisi u MySQL bazi, neophodno je da se prethodno uspostavi konekcija sa bazom. U slučaju da se ne može uspostaviti konekcija proces indeksiranja se prekida. Indeksi se moraju skladištiti u nekoj od memorija, što će u ovom slučaju biti direktorijum na tvrdom disku. Snimanje indeksa na tvrdi disk je jedino suvisno rešenje zbog količine podataka koja se indeksira iz CRIS UNS sistema. Kako bi se proces indeksiranja uspešno izvršio neophodno je proveriti dostupnost pomenute lokacije. Nakon toga sledi kreiranje posebnih java klasa koje u sebi enakapsuliraju metode za pribavljanje uskladištenih zapisa (*RecordDB*). Prilikom poziva metoda ovih klasa, neophodno je definisati koji tip naučno-istraživačkog podatka želimo da dobijemo kao rezultat selekcije nad bazom podataka. Nakon toga je neophodno da se dobijene vrednosti iz baze smeste u objektnu reprezentaciju zapisa (*Record*). *Record* u sebi sadrži string reprezentaciju MARC21 zapisa koju je neophodno transformisati u objektnu reprezentaciju MARC21 zapisa. Za ovaj proces je zadužena klasa *MARC21RecordFactory*. Pojedini zapisi imaju veze ka drugim zapisima (npr. organizacija ima podređene odnosno nadređene organizacije) te je neophodno da se i te relacije upišu u objektnu reprezentaciju zapisa. Svi pribavljeni zapisi se smeštaju u listu zapisa

Za indeksiranje je neophodno da se iz zapisa kreira i odgovarajući DTO (*Data Transfer Object*, 2003) objekta za taj zapis. DTO objekti nemaju poslovnu logiku već samo čuvaju podatke o određenom tipu zapisa. U zavisnosti od tipa naučno-istraživačkog podatka poziva se odgovarajući

konvertor (npr. za tip podatka institucija poziva se `InstitutionConvertor`) koji podatke za DTO dobija iz MARC21 reprezentacije koja se nalazi u zapisu. Ujedno je moguće da u slučaju da se vrednosti već nalaze u DTO objektu, one i iskoriste kako bi se popunila MARC21 objektna reprezentacija. Za indeksiranje je neophodno da se određeni delovi zapisa (DTO ili MARC21) upišu u *Lucene* dokument. eophodno je formirati praznu *Lucene* Document klasu u kojoj će smeštati *Lucene* polja, koja će našem slučaju biti popunjena vrednostima „prefiksa“ reprezentovano klasom `PrefixValue`. Prefiks je struktura koja se sastoji od dva tekstualna podatka: oznake (`prefName`) i vrednosti (`prefValue`). Naime neophodno je izvršiti transformaciju korišćenjem `PrefixConverter` i metoda `toPrefixes(record)` na željenim zapisom. Kada je u pitanju DTO objekat proverava se koji je tip naučno-istraživačkog podatka (organizacija, osoba ...) pa se za određenu oznaku vezuje vrednost atributa iz DTO objekta. (npr. `SOMENAME` prefiks se vezuje za ime organizacije dobijene iz DTO objekta).

Što se tiče MARC21 zapisa moguće je indeksirati zaglavlje (`Leader`), kontrolna polja (`ControlFields`) i polja sa podacima (`DataFields`). Određivanje prefiksa za MARC21 zapis u formi stringa se svodi na njegovo pronalaženje u mapi prefiksa. Mapa prefiksa sadrži tekstualnu oznaku uparenu za odgovarajućim tipom podataka (vrstom naučno-istraživačkog podataka) i oznakom polja i potpolja podatka u MARC21 zapisu. Tako se npr. za ime autora (*type author*) koristi polje 100a i vezuje za tekstualnu oznaka "AU". Moguće je vezati **više polja za istu tekstualnu oznaku**, što rezultuje činjenicom da je moguće pretražiti **više polja istim indeksom**. Tako bi se u oznaku prefiksa upisalo "AU" dok bi se vrednost polja i potpolja iz objektna reprezentacije MARC21 upisala u vrednost prefiksa. U *Lucene* polje se oznaka prefiksa upisuje kao ime polja dok se vrednost prefiksa upisuje kao vrednost polja. Na slici **4.7** se vide set klasa iz sistema pretrage koje predstavljaju mini model koji je zadužen za rukovanje prefiksima. Java interfejsi iz modela (`PrefixMap`, `PrefixHandler`, `PrefixConfig`) pružaju mogućnost polimorfizma, tako npr. možemo imati više različitih mapa prefiksa.



Slika 4.7 Deo modela za upravljanje prefiksima

U prethodnom segmentu je istaknut značaj opcija koje se koriste prilikom indeksiranja vrednosti polja. Kako bi se za svako polje odredilo način analize i indeksiranja, prefiksi (oznake) su svrstane u četiri grupe *notAnalyzedNotStored*, *notAnalyzed*, *notIndexed*, *rest*. Prilikom kreiranja polja proverava se kojoj grupi polja pripada prefiks. Grupa *notAnalyzedNotStored* ne uključuje analizu vrednosti polja i ne čuva njegovu vrednost (`Field.Store.NO`, `Field.Index.NOT_ANALYZED`), gde bi adekvatan primer bio kontrolni broj (identifikator) zapisa autor ("AUCN"). Grupa *notAnalyzed* ne uključuje analizu vrednosti polja ali čuva njegovu vrednost (`Field.Store.YES`, `Field.Index.NOT_ANALYZED`), gde bi adekvatan primer bio serijski broj publikacija ISSN. Grupa *notIndexed* samo čuva vrednosti polja ali nije moguće pretražiti polja iz ove grupe (`Field.Store.YES`, `Field.Index.NO`), gde bi adekvatan primer bio *FILEURL* (URL do fajla). Grupa *rest* predstavlja najveći broj indeksa iz sistema i ona uključuje upotrebu odgovarajućeg analizatora ali ne skladišti vrednosti termina (`Field.Store.NO`, `Field.Index.ANALYZED`). U CRIS UNS sistemu postoje višejezični podaci (npr. Naslov teze ili disertacije), što je značajno prilikom analize istih, jer se različite operacije analize primenjuju u zavisnosti od jezika samog podatka. Analiza polja se izvršava neposredno pre samog smeštanja polja u indeks. Za potrebe analize razvijen je poseban analizator *CrisAnalyzer*. Deo kod implementiranog

CrisAnalyzer nalazi se na listingu 4.1. Pomenuti analizator obavlja sledeće operacije:

- Vršiti se tokenizacija po praznom mestu
- Prebacivanje sadržaja u mala slova
- Konvertovanje sadržaja u latinično pismo
- Primena stop filtera u zavisnosti od jezika podataka. Svaki od stop filtera definiše set “stop” (nepotrebnih, učestalih) reči. Tako bi učestale za engleski jezik bile: *a, an, the* ... Dok su za srpski jezik učestale reči npr. *ne, sam, si, je*.
- Svođenje reči na korenski oblik (stemovanje – *stemming*). U zavisnosti od jezika koristi se odgovarajući stemer. Tako se za stemovanje engleskih reči koristi čuveni *Porter stemming* algoritam (Willett, 2006), dok se za srpski jezik koristi poseban *steming* algoritam.
- Prebacivanje reči iz hrvatskog jezika na srpski jezik
- Uklanjanje akcenata sa pojedinih slova (npr. č, ć ...)

Nakon ove operacije vrši se smeštanje analiziranog polja u *Lucene* dokument i na kraju se nad dokumentom pozove *IndexWriter* klasa kako bi se dokument i dodao u indeksni fajl.

```
public class CrisAnalyzer extends Analyzer {
    private Set<String> serbianStopSet;
    private Set<String> englishStopSet;
    private static final String[] SERBIAN_STOP_WORDS = {"biti",
"ne", "jesam", "sam", "jesi", "si",
...
private static final String[] ENGLISH_STOP_WORDS ={"a", "an", "and",
"are", "as", "at", "be", "but",
    public CrisAnalyzer() {
        super();
serbianStopSet = StopFilter.makeStopSet(SERBIAN_STOP_WORDS);
        englishStopSet
StopFilter.makeStopSet(ENGLISH_STOP_WORDS);
    }
    public TokenStream tokenStream(String fieldname, Reader reader) {
        if(fieldname.endsWith("_SRP")){
            return new RemoveAccentsFilter( new
(CroSerTranslateFilter)
...
new WhitespaceTokenizer(reader)), serbianStopSet, new
SerbianStemmer()
else if(fieldname.endsWith("_ENG")){
```

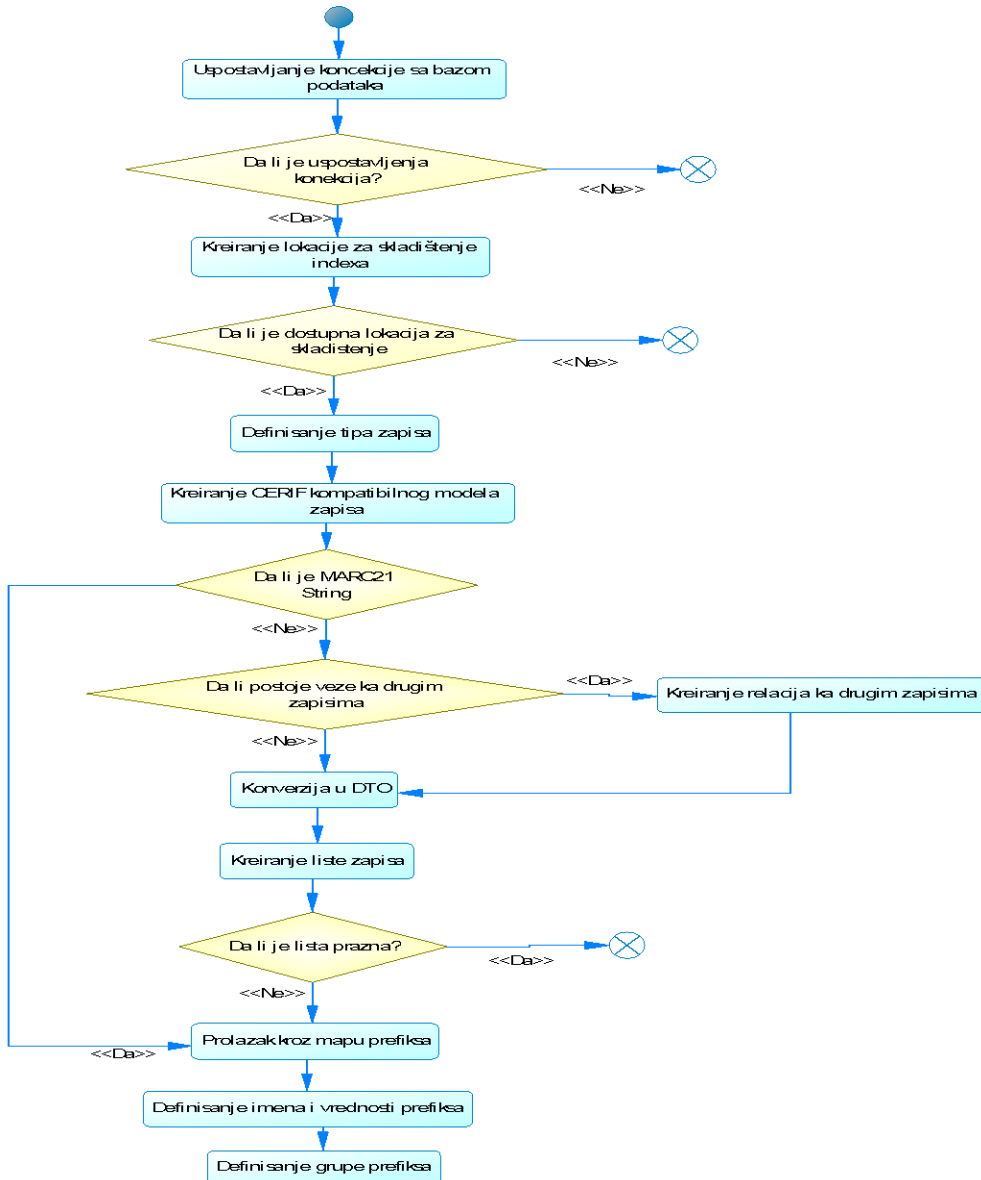
```

return new RemoveAccentsFilter(
    new WhitespaceTokenizer(reader)), englishStopSet, new
PorterStemmer())

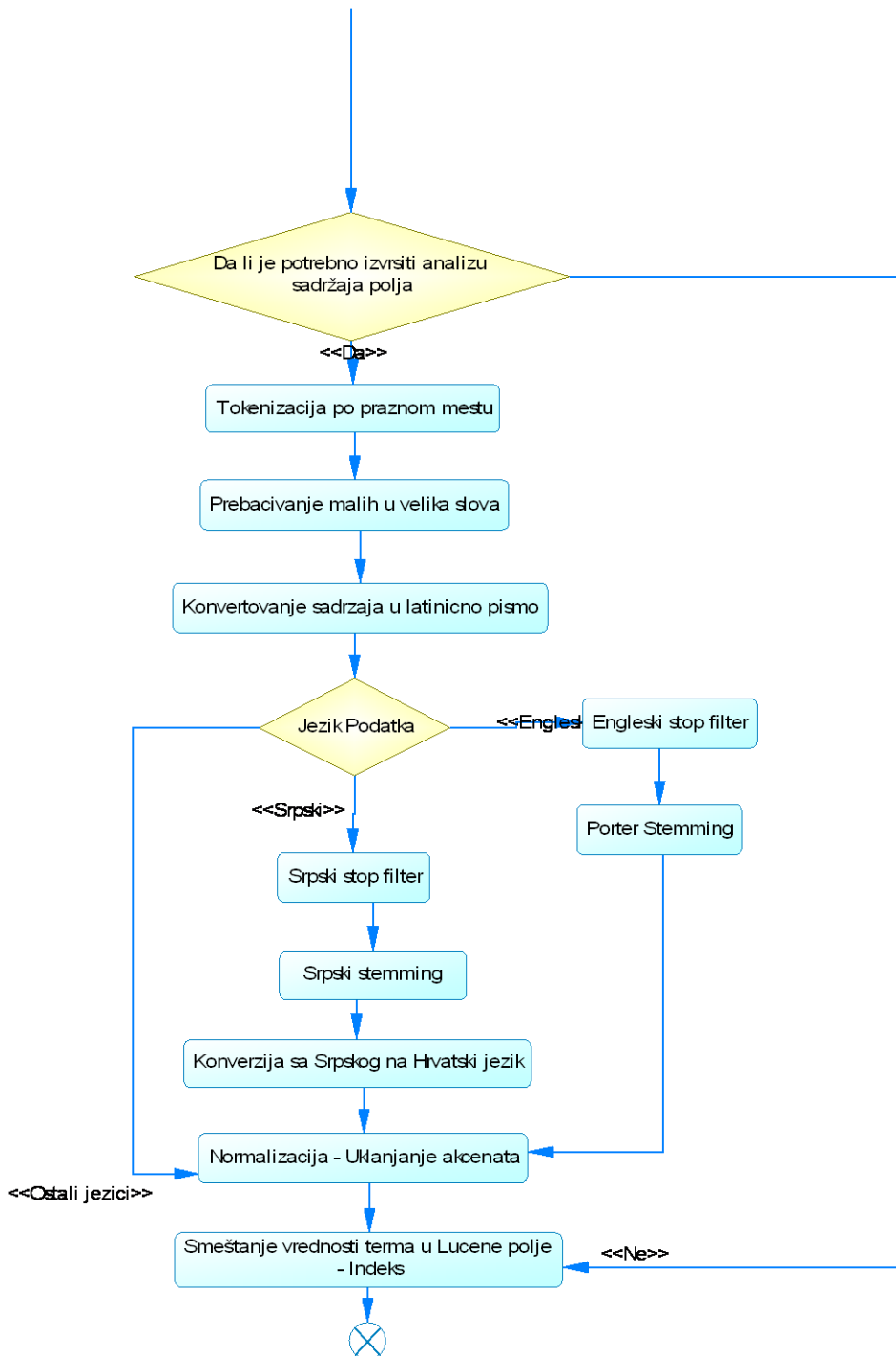
```

Listing 4.1 Deo CrisAnalyzer klase

Komponenta za indeksiranje se stara da se indksi ažuriraju novim prilikom svakom unosa novog zapisa u CRIS UNS sistem.Čitav tok indekiranja dat je na dijagramu aktivnosti na slikama 4.8 i 4.9.



Slika 4.8 Dijagram aktivnosti toka indeksiranja prvi deo



Slika 4.9 Dijagram aktivnosti toka indeksiranja drugi deo

5. Pretraživanje podataka unutar CRIS sistema

Cilj istraživanja u ovoj disertaciji bio je da se razvije proširenje postojećeg informacionog sistema naučno-istraživačke delatnosti opisanog u (Ivanović et al., 2010). Navedeno proširenje je zapravo sofisticiran podsistem koji služi za pretragu prethodno pohranjenih podataka. U ovom poglavlju data je relevantna specifikacija informacionih zahteva i arhitektura informacionog sistema za pretragu metapodataka naučno-istraživačke delatnosti. Osnovna ideja je da se pretraga zasniva na bibliotečkima standardima SRU/W i CQL, a da korisnici mogu da pretražuju sistem bez potrebe da prethodno poznaju strukture podataka u sistemu (MARC21, CERIF) kao i pomenute standarde pretrage. Takođe, korisnicima je potrebno omogućiti pretragu po terminima koji su nezavisni sa stanovišta korišćenog pisma (latinični i ćirilčni unosi tretiraju se ravnopravno).

5.1. Specifikacija informacionih zahteva

Na osnovu korisničkih zahteva i dobijenih rezultata u (Penca et al., 2012) i (Penca et al., 2014b) formulisana je struktura i funkcionalne mogućnosti sistema pretrage naučno-istraživačkih podataka. Potrebno je da se obezbedi pretraga sledećih naučno-istraživačkih podataka:

- publikacija (radovi u časopisima, radovi u zbornicima sa konferencija, monografije, teze i disertacije)
- autora/istraživača (prvi autor, koautori, editori)
- organizacija (naučne institucije, univerziteti, fakulteti, departmani, katedre)
- časopisa
- konferencija (skupovi, seminari)
- patenta
- produkta
- projekta

Specifikacija pretrage podataka naučno-istraživače delatnosti prezentovana je u formi UML (Unified Modeling Language (UML), 1997) dijagrama slučajeva korišćenja. Prikazani dijagrami slučajeva korišćenja opisuju ponašanje tj. funkcionalnost sistema bez definisanja njegove interne strukture. U nastavku su prikazani sledeći dijagrami slučajeva korišćenja:

- pretraga publikacija
- pretraga autora
- pretraga tekstualnim zadavanjem upita
- obrada korisničkog zahteva

Nakon prikazanih dijagrama dati su opisi slučajeva korišćenja. Učesnici slučajeva korišćenja su *User* (Korisnik) koji inicira pretragu i *System* (Sistem) koji obrađuje upit zadat CQL jezikom i vraća skup pogodaka. U opisu navedenih segmenata primenjena su sledeća pravila:

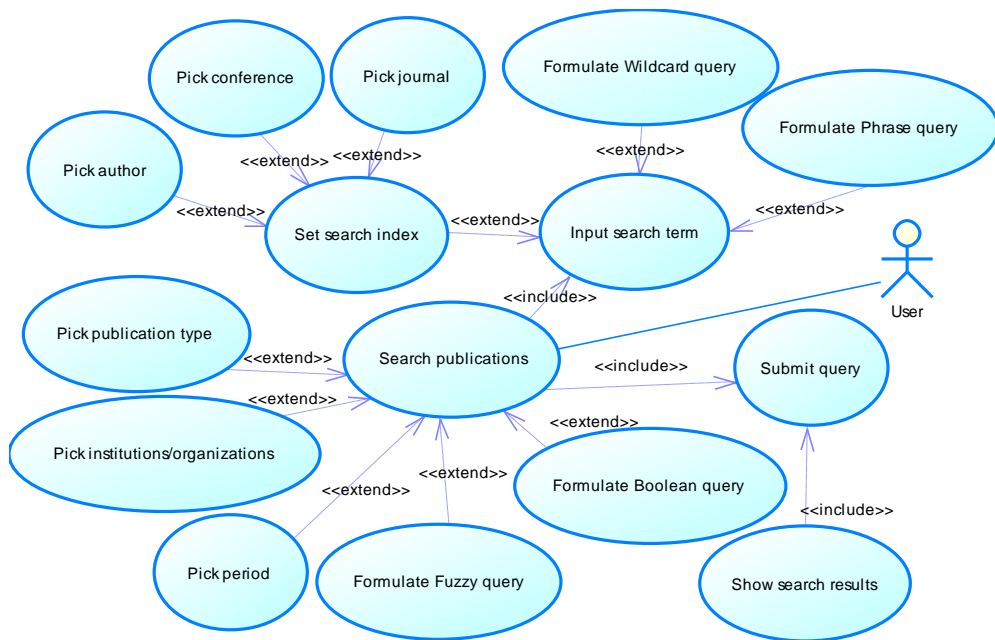
- Naziv slučaja korišćenja navodi se između znakova "<" i „>”.
- Tekst (INCLUSION: <naziv slučaja korišćenja>) označava obavezno izvršenje navedenog slučaja korišćenja u okviru posmatranog.
- Tekst (EXTENSION: <naziv slučaja korišćenja>) označava da učesnik može (ali ne mora) inicirati izvršenje navedenog slučaja korišćenja koji proširuje funkcionalnost slučaja korišćenja koji se opisuje.

5.1.1. Pretraga publikacija

Dijagram slučajeva korišćenja koji se odnosi na pretragu naučnih publikacija prikazan je na slici **5.1**. U nastavku biće dato objašnjenje slike **5.1**.

<Search publications>. Korisnik odabire sekciju “Pretraga publikacija”. Korisnik unosi tekst za pretragu (INCLUSION: <Input search term>). Uneseni tekst može se tretirati kao približna pretraga (EXTENSION: <Formulate Fuzzy query>). Korisnik može definisati više termova za pretragu (EXTENSION: <Formulate Boolean query>) povezujući ih logičkim operatorima. Pretraga se može limitirati odabirom vremenskog perioda (EXTENSION: <Pick period>), tipa publikacije (EXTENSION: <Pick publication type>) i organizacione pripadnosti (EXTENSION: <Pick

institutions/organizations>). Korisnik prosleđuje upit pretrage sistemu (INCLUSION: <Submit query>).



Slika 5.1 Pretraga publikacija

<Search publications>. Korisnik odabire sekciju “Pretraga publikacija”. Korisnik unosi tekst za pretragu (INCLUSION: <Input search term>). Uneseni tekst može se tretirati kao približna pretraga (EXTENSION: <Formulate Fuzzy query>). Korisnik može definisati više termova za pretragu (EXTENSION: <Formulate Boolean query>) povezujući ih logičkim operatorima. Pretraga se može limitirati odabirom vremenskog perioda (EXTENSION: <Pick period>), tipa publikacije (EXTENSION: <Pick publication type>) i organizacione pripadnosti (EXTENSION: <Pick institutions/organizations>). Korisnik prosleđuje upit pretrage sistemu (INCLUSION: <Submit query>).

<Input search term>. Unos teksta po kome će se vršiti pretraga publikacija. Korisnik može odabrati indeks za koji će se vršiti pretraga (EXTENSION: <Set search index>). Prilikom unosa teksta korisniku se pruža mogućnost da definiše uneseni tekst kao pretragu fraze (EXTENSION: <Formulate Phrase query>) ili pretragu nedovršenih reči/rečenica (EXTENSION: <Formulate Wildcard query>).

<Set search index>. Odabir tipa indeksa za polje pretrage. Ponuđeni indeksi su: sva polja; naslov, apstrakt i ključne reči; autor; prvi autor; naslov publikacije; apstrakt; ključne reči; naziv konferencije; naziv časopisa. Sistem korisniku prikazuje listu mogućih vrednosti ukoliko se odabere neki od sledećih indeksa: autor (EXTENSION: <Pick author>), naziv konferencije (EXTENSION: <Pick conference>) ili naziv časopisa (EXTENSION: <Pick journal>).

<Pick author>. Odabirom indeksa autor za polje pretrage, pri daljem unosu teksta terma, aktivira se padajuća lista predloženih vrednosti koja sadrži spisak imena autora. Autori pripadaju listi ukoliko njihovi nazivi ispunjavaju kriterijum sličnosti sa unešenim termom. Kriterijum sličnosti je definisan sledećim pravilima: slova ćirilicnog i latinićnog pisma se ravnopravno tretiraju prilikom zadavanja terma, reć u termu se ne razlikuje za više od 2 slova od reći u nazivu autora, ukoliko u termu postoji više od 5 reći onda prethodno pravilo važi za 80% reći.

<Pick conference>. Odabirom indeksa naziv konferencije za polje pretrage, aktivira se lista predloženih vrednosti koja sadrži spisak naziva konferencija. Konferencije pripadaju listi ukoliko njihovi nazivi ispunjavaju kriterijum sličnosti sa unešenim termom. Kriterijum sličnosti je identičan kao kod <Pick author> samo što se primenjuje nad nazivima konferencija.

<Pick journal>. Odabirom indeksa naziv časopisa za polje pretrage, aktivira se lista predloženih vrednosti koja sadrži spisak naziva časopisa. Časopisi pripadaju listi ukoliko njihovi nazivi ispunjavaju kriterijum sličnosti sa unešenim termom. Kriterijum sličnosti je identičan kao kod <Pick author> samo što se primenjuje nad nazivima časopisa

<Formulate Phrase query>. Formulisanje upita koji će uneseni tekst pretražiti kao frazu. Korisnik definiše operator fraze nad unesenim tekstem. Operator fraze predstavljen je parom otvorenih i zatvorenih navodnika

<Formulate Wildcard query>. Formulisanje upita koji će uneseni tekst tretirati kao pretragu nedovršениh reći/rećenica. Korisnik definiše neki od mogućih wildcard operatora nad unetim tekstem. Operator "?" zamenjuje taćno jedan neodređen karakter sa kojim se mođe započeti, završiti ili upotpuniti u celosti reć u okviru terma. Sa operatorom "*" korisnik definiše niz karaktera proizvoljne dužine i sadržaja kojim se mođe završiti ili upotpuniti sadržaj terma.

<Pick period>. Limitiranje pretrage publikacija u odnosu na vremenski period (donja i gornja granica perioda). Korisnik može odabrati donju i/ili gornju granicu. Granica se definiše kao brojeva vrednost ili podrazumevana vrednost.

<Pick institution/organization>. Limitiranje pretrage publikacija u odnosu na autorovu pripadnost organizaciji/instituciji. Korisnik iz hijerarhijskog prikaza organizacione strukture naučno-istraživačkih organizacija/institucija odabire željene organizacije/institucije

<Pick publication type>. Limitiranje pretrage publikacija u odnosu na vrstu publikovanog materijala. Korisnik iz liste odabire željeni tip publikacije (radovi u časopisu, radovi u zbornicima sa konferencija, monografije, teze i disertacije).

<Formulate fuzzy query>. Formulisanje upita koji će unesene termine tretirati kao približnu pretragu. Korisnik odabire opciju približne pretrage. Približna pretraga predstavlja pronalaženje svih vrednosti koje se u velikoj meri (minimalno 80%) poklapaju sa pretraženim tekstom.

<Formulate Boolean query>. Formulisanje logičkog upita kojim se definiše međusobni odnos između polja u upitu. Odnos između susednih polja može se definisati kao: konjunkcija (AND), disjunkcija (OR) ili konjunkcija sa negacijom drugog polja (NOT).

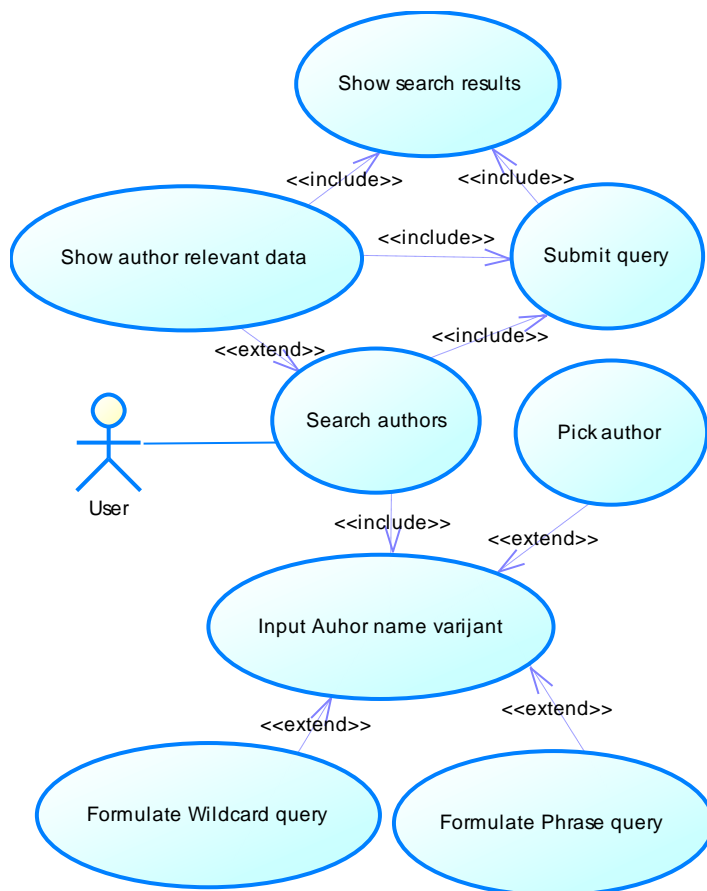
<Submit query>. Korisnik inicira pretragu podataka. Na osnovu unesenih vrednosti u interfejsu formira se CQL upit pretrage koji se prosleđuje sistemu. Sistem po obradi upita vraća rezultate pretrage koji se prikazuju u korisničkom interfejsu (INCLUSION: <Show search results>).

<Show search results>. Pretpostavka je da je upit obrađen i uspešno izvršen. Korisniku je prikazan ukupan broj pronađenih zapisa i lista njihovih tekstualnih reprezentacija. U slučaju da je obrada upita pretrage rezultovala greškom, korisniku će biti prikazana odgovarajuća poruka koja deskriptivno objašnjava uzrok greške (npr. pretraga po indeksu koji nije podržan u okviru CRIS kontekst seta; uneseni upit ne odgovara sintaksi CQL jezika...).

5.1.2. Pretraga autora

Dijagram slučajeva korišćenja koji se odnosi na pretragu autora prikazan je na slici 5.2. Prikazani slučajevi korišćenja <Formulate Phrase query>,

<Formulate Wildcard query>, <Pick author> i <Submit query> opisani su u prethodnom odeljku te se neće posebno objašnjavati, dok će u narednom tekstu biti opisani preostali slučajevi korišćenja.



Slika 5.2 Pretraga autora (istraživača)

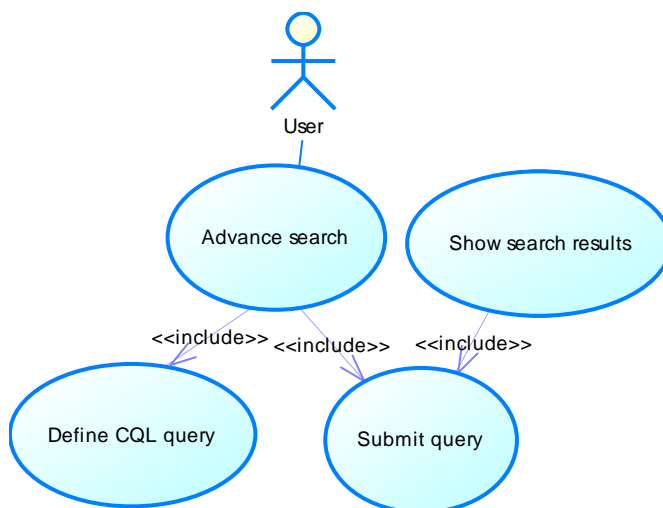
<Search authors>. Korisnik odabire sekciju “Pretraga autora/istraživača”. Pretraga autora/istraživača sa mogućnošću pregleda dodatnih informacija (publikacija, radova u časopisima, radova na konferencijama, monografija, teza i disertacija) o pretraženim autora/istraživača. Korisnik unosi ime i/ili prezime autora (INCLUSION: <Input author name variant>). Zatim se upit pretrage prosleđuje sistemu (INCLUSION: <Submit query>). Prikazuju se rezultati pretrage <Show search results>. Ukoliko lista rezultata (autora/istraživača) nije prazna, korisniku se pruža mogućnost pretrage relevantnih informacija za autore (EXTENSION: <Show author relevant data>).

<Input author name variant>. Unos teksta po kome će se vršiti pretraga nekih od varijanti imena i prezimena autora. Korisnik može definisati da se uneseni tekst tretira kao pretraga fraza (EXTENSION: <Formulate Phrase query>) ili nedovršenih reči/rečenica (EXTENSION: <Formulate Wildcard query>). Prilikom unosa teksta sistem korisniku prikazuje listu mogućih vrednosti koja sadrži imena autora (EXTENSION: <Pick author>).

<Show author relevant data>. Prikaz relevantnih skupova podataka (publikacija, radova u časopisima, radova na konferencijama, monografija, teza i disertacija) o autoru/istraživaču. Korisnik odabirom linka ka relevantnom skupu podataka inicira pretragu sistema (INCLUSION: <Submit query>). Prikazuju se rezultati pretrage <Show search results>.

5.1.3. Pretraga tekstualnim zadavanjem upita

Dijagram slučajeva korišćenja koji se odnosi na tekstualno zadavanje upita prikazan je na slici 5.3. Prikazani slučaj korišćenja <Submit query> opisan je u prethodnom odeljku te se neće posebno objašnjavati, dok će u narednom tekstu biti opisani preostali slučajevi korišćenja.



Slika 5.3 Tekstualno zadavanje upita

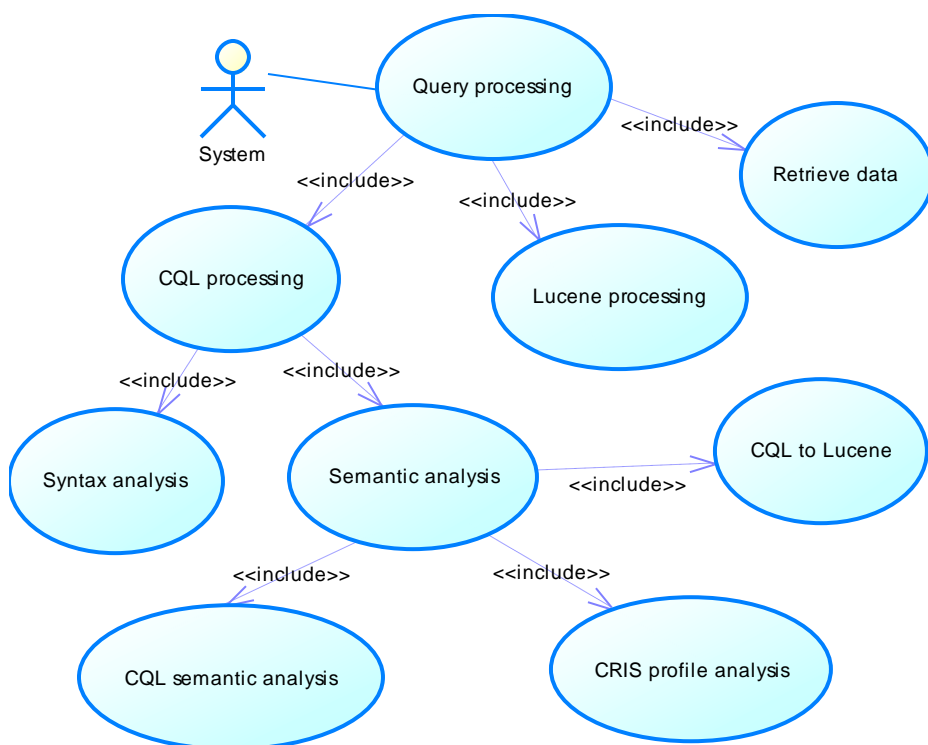
<Text query search>. Korisnik odabire sekciju “Tekstualna pretraga”. U polju za unos teksta korisnik unosi CQL upit (INCLUSION: <Define CQL query>). Upit pretrage prosleđuje se sistemu (INCLUSION: <Submit

query>). Sistem po obradi upita vraća rezultate pretrage koji se prikazuju u korisničkom interfejsu (INCLUSION: <Show search results>).

< Define CQL query >. Korisnik u formi slobodnog teksta zadaje upit koji odgovara CQL specifikaciji i saglasan je sa CRIS profilom pretrage.

5.1.4. Procesiranje CQL upita i vraćanje rezultata

Dijagram slučajeva korišćenja koji opisuje obradu CQL upita i definiše odgovor na prosleđeni upit prikazan je na slici 5.4.



Slika 5.4 Procesiranje CQL upita i vraćanje rezultata

<Query processing>. Sistem prihvata CQL upit i na osnovu njega formira odgovarajući odgovor (rezultate pretrage ili poruka o grešci). Pristigao CQL upit se obrađuje (INCLUSION: <CQL processing>) i transformiše u upitni jezik tekst servera (Lucene). Zatim sledi procesiranje Lucene upita (INCLUSION: <Lucene processing>) iz kojeg nastaju rezultati pretrage (INCLUSION: <Retrieve data>). U zavisnosti od parametara poziva, sistem

prosleđuje rezultate pretrage korisničkom interfejsu ili generiše poruke za razmenu sa odgovarajućim SRU/W klijentima.

<CQL processing>. Sistem prihvata CQL upit i vrši njegovu sintaksnu (INCLUSION: <Syntax analysis>) i semantičku analizu (INCLUSION: <Semantic analysis>). Na osnovu obrađenog CQL upita nastaje Lucene reprezentacija istoimenog upita. Procesiranje upita nije rezultiralo greškom, izvršava se mapiranje CQL na Lucene upit.

<Syntax analysis>. Sistem parsira pristigli CQL upit. Upit se dekomponuje na osnovne strukture CQL jezika i vrši se njihova sintaksna validacija. Ukoliko je validacija upita neuspešna, sistem će prekinuti dalje procesiranje upita i vratiće poruku o grešci.

<Semantic analysis >. Sistem proverava da li je CQL upit u skladu sa semantikom CQL jezika i CRIS profila. Izvršava se osnovna semantička provera CQL upita (INCLUSION: <CQL semantic analysis>). Zatim se izvršava semantička provera CQL upita (INCLUSION: <CRIS profile analysis>) na osnovu definisanog CRIS profila. Semantičke provere nisu rezultirale greškom, vrši se pretvaranje CQL upita u upitni jezik tekst servera (INCLUSION: <CQL to Lucene>).

<CQL semantic analysis>. Sistem prolazi kroz pojedinačne strukture CQL upita. Vršiti se provera osnovnih semantičkih koncepata propisanih CQL jezikom. Pri analizi semantike nije došlo do greške, tada sistem transformiše delove CQL upita u odgovarajuće objektne reprezentacije CQL upita.

<CRIS profile analysis>. Sistem prolazi kroz hijerarhijsku strukturu CQL objekata. Sistem proverava dozvoljene vrednosti relacija, modifikatora i indeksa sa vrednostima definisanim u kontekst setovima CRIS profila. Ukoliko je došlo do greške sistem će prekinuti dalju obradu CQL upita.

<CQL to Lucene>. Sistem prolazi kroz hijerarhijsku strukturu CQL objekata i izvršava transformaciju (mapiranje) CQL u Lucene reprezentaciju upita. Mapiranje se primenjuje kako na nivou osnovnih koncepata (indeksa, relacija, relacionih modifikatora, termova) tako i na nivou složenih konstrukcija (npr. logički upiti). Ceo CQL upit preveden je u format koja je u skladu sa Lucene jezikom.

<Lucene processing>. Sistem prihvata Lucene upit, obrađuje ga i vraća skup odgovora. Poziva se analizator Lucene upita koji vrši parsiranje i

normalizaciju vrednosti polja (prebacivanje velikih u mala slova, prebacivanje ćirilčnih u latinična slova, filtriranje rezervisanih karaktera,...) polja u upitu. Sistem izvršava Lucene upit nad indeksiranim podacima.

<Retrieve data>. Sistem obrađuje rezultate pretrage. U zavisnosti od parametara poziva rezultati pretrage se konvertuju u odgovarajuću objektu (Java Object) ili XML (CERIF XML, Dublin Core XML, MARC XML...) reprezentaciju.

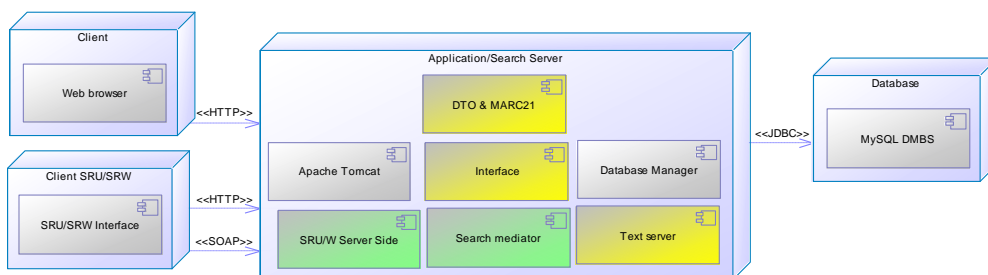
5.2. Implementacija sistema za pretragu podataka naučno-istraživačke delatnosti

U ovom poglavlju opisana je implementacija aplikacije za pretragu podataka naučno-istraživačke delatnosti. Aplikacija poseduje višeslojnu klijent-server arhitekturu zasnovanu na skupu rešenja otvorenog koda, baziranih pre svega na programskom jeziku Java. Aplikacija je namenjena svim korisnicima koji žele da dođu do podataka iz domena naučno-istraživačke delatnosti. Za pristup aplikaciji koristi se proizvoljan veb čitač (eng. *web browser*) ili proizvoljna aplikacija koja implementira klijentsku stranu SRU/W protokola. Koristeći veb čitač, korisniku je omogućena pretraga sistema bez prethodnog znanja navedenog jezika. Sistem za pretragu se oslanja na kontekst setoveopisanih XML notacijom. Izdvajanjem opisa kontekst setova od programskog koda pojednostavljeno je preuzimanje opisa kontekst setova drugim aplikacijama, kao i njihova jednostavna izmena bez potrebe modifikacije programskog koda same aplikacije. U ovom poglavlju opisana je arhitektura sistema, statički i dinamički model sistema. Na kraju poglavlja opisano je korišćenje same aplikacije.

5.2.1. Statički model sistema

Sistem za pretragu naučno-istraživačkih rezultata je integrisan sa postojećim sistemom za upravljanje naučno-istraživačkim podacima. Integracija ovih sistema ostvarena je izmenom postojećih i dodavanjem novih komponenti. S obzirom da je sistem za upravljanje naučno-istraživačkih podataka razvijen kao veb aplikacija, integrisani sistem mora podržati postojeći protokol za komunikaciju (HTTP). Na slici **5.5** prikazana je arhitektura sistema upotrebom dijagrama razmeštaja. Žutom bojom su prikazane izmenjene

komponente, dok je zelenom bojom prikazane dodate komponenta u postojećem CRIS UNS sistemu.



Slika 5.5 Implementacija arhitekture sistema

Client - Web Browser Prvi tip klijentske strane aplikacije čini veb čitač. Moguće je koristiti sve savremene čitače koji podržavaju minimalno HTML 4, skript jezik JavaScript i CSS (Mozilla Firefox, Google Chrome, Internet Explorer i drugi). Komunikacija između klijentske aplikacije i serverske strane aplikacije obavlja se posredstvom HTTP protokola.

Client SRU/SRW - SRU/W Interface: Drugi tip klijentske strane aplikacije predstavlja proizvoljna eksterna aplikacija (deo nekog drugog sistema) koja implementira klijentsku stranu SRU/W protokola verzije 2.0. Kontekst setovi CQL verzije 1.2 i DC verzije 1.1 čine neophodni minimum koji mora ispuniti klijentska strana aplikacije. Pretraga svih raspoloživih zapisa sistema moguća je ukoliko klijentska strana aplikacije podržava CRIS profil (opisan u narednom poglavlju). Komunikacija između klijentske aplikacije i serverske strane aplikacije obavlja se posredstvom HTTP ili SOAP protokola.

Application/Search Server

Apache Tomcat: Serverska strana aplikacije izvršava se na aplikativnom serveru Apache Tomcat (Apache Tomcat, 2014) napisanom u Javi.

SRU/W Server Side: Komponenta izvršava serversku stranu SRU/W protokola verzije 2.0 implementacijom *Search/Retrive*, *Scan* i *Explain* servisa. Serverska strana SRU/W prihvata posredstvom HTTP i SOAP protokola *Search/Retrive Request* zahteve i obrađuje CQL upit definisan unutar zahteva. *SRU/W Server Side* komponenta poziva komponentu *Search Mediator* i prosleđuje joj CQL upit na dalju obradu. Rezultati pretrage dobijeni od *Search Mediator* komponente, *SRU/W Server Side* prosleđuje odgovarajućem SRU/W klijentu.

Interface: Komponenta omogućuje korisniku kreiranje CQL upita pretrage upotrebom vizuelnih komponenti. Za implementaciju GUI korišćena je JSF (“Mojarra JavaServer Faces,” 2014) tehnologija. JSF se sve više koristi u kombinaciji sa AJAX tehnologijom. Upotrebom AJAX tehnologije deo upravljanja i kontrole SUI prebacuje se sa serverske na klijentsku stranu, što de fakto utiče na poboljšanje performansi same aplikacije. U implementaciji aplikacije koja je opisana u ovom radu korišćena je *RichFaces* (RichFaces - JBoss, 2014) biblioteka JSF komponenti baziranih na AJAX, što doprinosi da interfejs postaje interaktivniji i vizuelno bogatiji. *Interface* komponenta omogućuje korisniku da preko SUI poziva pretragu naučno-istraživačkih podataka pri čemu je ona direktno odgovorna za konstrukciju CQL upita. Po konstrukciji CQL upita komponenta *Interface* poziva komponentu *Search Mediator* i prosleđuje joj CQL upit na dalju obradu.

Search Mediator: Komponenta koja omogućuje pretragu sistema korišćenjem CQL upitnog jezika. XML opisi kontekst setova i specifikacija CQL jezika koriste se za verifikaciju CQL upita (ispravnost sintakse i semantike upita). Sistem koristi pomenute XML opise i JAXB (JAXB Reference Implementation, 2005) biblioteke za formiranje objektno predstave kontekst setova. CQL upit se transformiše u upitni jezik komponente Text Server, a zatim se transformisani upit izvršava u Text Server. Search mediator komponenta prihvata i obrađuje korisničke CQL upite SRU/W Server Side i Interface komponenti, nakon čega se rezultati pretrage konvertuju u zavisnosti koja je komponenta inicirala poziv *Search Mediator* komponente. *Search Mediator* komponenta koristi konvertore definisane u DTO & MARC21 komponenti.

Text server: Komponenta *Text Server* za potrebe pretraživanja i indeksiranja naučno-istraživačkih podataka koristi *Apache Lucene* biblioteku. Zbog činjenice da sam sistem ravnopravno podržava unos podataka upotrebom ćirilicnog ili latinićnog pisma komponenta *Text Server* mora da omogući pretragu za upite konstruisane upotrebom navedenih pisama. Tako je neophodno da se svi ćirilicni sadržaji pre indeksiranja prevedu na latinićno pismo, a prilikom pretrage podataka svi ćirilicni upiti se prevode na latinićno pismo. Stoga, *Apache Lucene* radi samo sa sadržajima zapisanim latinićnim pismom, dok se u komponenti Database sadržaji ćuvaju onako kako ih je korisnik uneo. Prevođenje ćirilicnih sadržaja na latinićno pismo je jednoznaćno.

DTO & MARC 21: *DTO & MARC 21* komponenta omogućuje konverziju između DTO objekata i objektne reprezentacije MARC 21 zapisa. DTO su objekti koji se koriste za transport podataka između komponenti aplikacije. Prilikom pretrage sistema komponenta *Mediator* koristi komponentu *DTO & MARC 21* za konverziju rezultata iz MARC 21 objektne reprezentacije u korespondentne DTO objekte ili XML predstave (MARC 21 XML Schema, 2014) ili (Dublin Core Extended XML, 2008). Ukoliko je pretraga inicirana od komponente *Interface*, komponenta *DTO & MARC 21* neće pokrenuti učitavanje korespondentnih DTO objekte iz baze podataka, već će se samo preuzeti njihova postojeća tekstualna reprezentacija iz *Apache Lucene*. Zarad poboljšanja performansi sistema izbegava se potpuno učitavanje zapisa pohranjenih u bazi podataka.

Database Manager: Komponenta zadužena za komunikaciju sa bazom podataka.

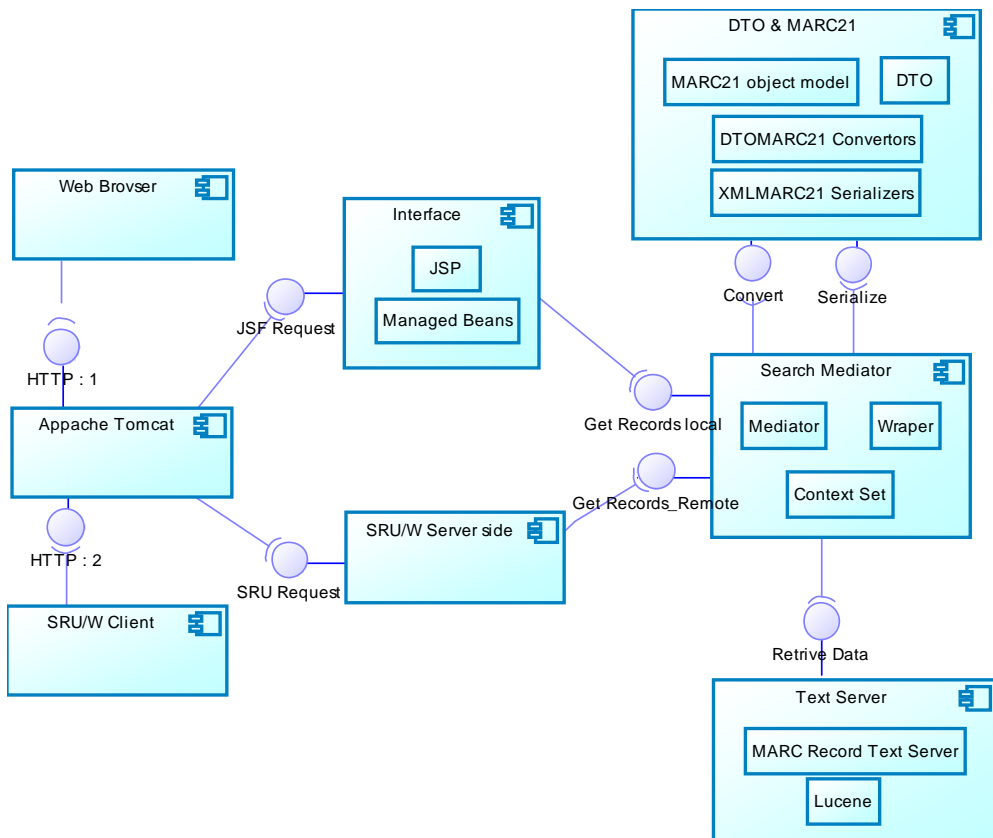
Database

MySQL DBMS: Kao sistem za upravljanje bazom podataka koristi se MySQL (MySQL, 2014).

5.2.2. Dijagram komponenti

Na slici 5.6 je prikazan dijagram komponenti aplikacije. Ovakva arhitektura omogućuje sledeće:

- Razvijanje eksterne aplikacije za pretragu, koja bi se bazirala na CQL bibliotečkom standardu.
- Promenu upitnog jezika Tekst servera, koja ne bi uticala na komponentu *Interface* preko koje korisnik kreira upite pretrage.
- Mogućnost uporedne pretrage više Tekst servera čija realizacija i fizička lokacija može biti proizvoljna. U tom slučaju, potrebno je samo definisati odgovarajuća mapiranja CQL jezika na jezike korišćenih Tekst servera.

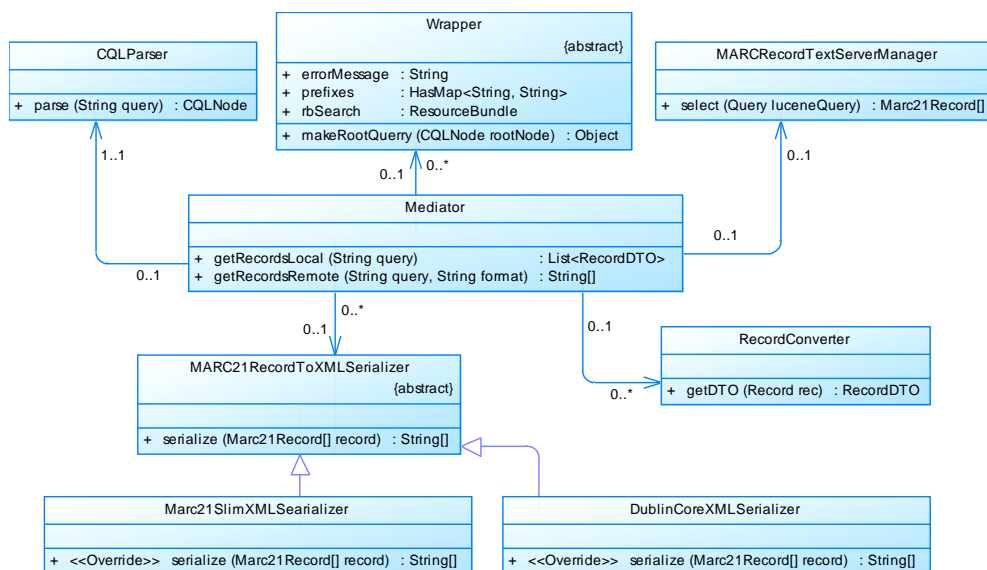


Slika 5.6 Dijagram komponenti sistema za pretragu

5.2.3. Dijagram klasa

Na slici 5.7 i slici 5.8 prikazani su dijagrami klasa koje čine jezgro sistema pretrage. Klasa *Mediator* (slika 5.7) deo je istoimenog paketa koja pripada komponenti *Search Mediator*. Klasa upravlja procesima sintaksne i osnovne semantičke analize CQL upita, kao i njegovom transformacijom u upitni jezik Tekst servera. Metoda *getRecordsLocal* koristi se kod lokalne obrade CQL upita (inicira *Interface* komponenta), pri čemu je rezultat poziva metode objektna reprezentacija zapisa podataka Tekst servera. Za razliku od prethodne funkcije, *getRecordsRemote* funkciju poziva komponenta *SRU/W Server side*. Funkcija *getRecordsRemote* na osnovu CQL upita vraća XML reprezentacije zapisa u formatu definisanim parametrom *format*. Klasa *CQLParser* transformiše korisnički CQL upit u njegovu korespondentnu objektnu reprezentaciju (*CQL-Java* - <http://zing.z3950.org/cql/java/>) i ujedno izvršava njegovu sintaksnu i osnovnu

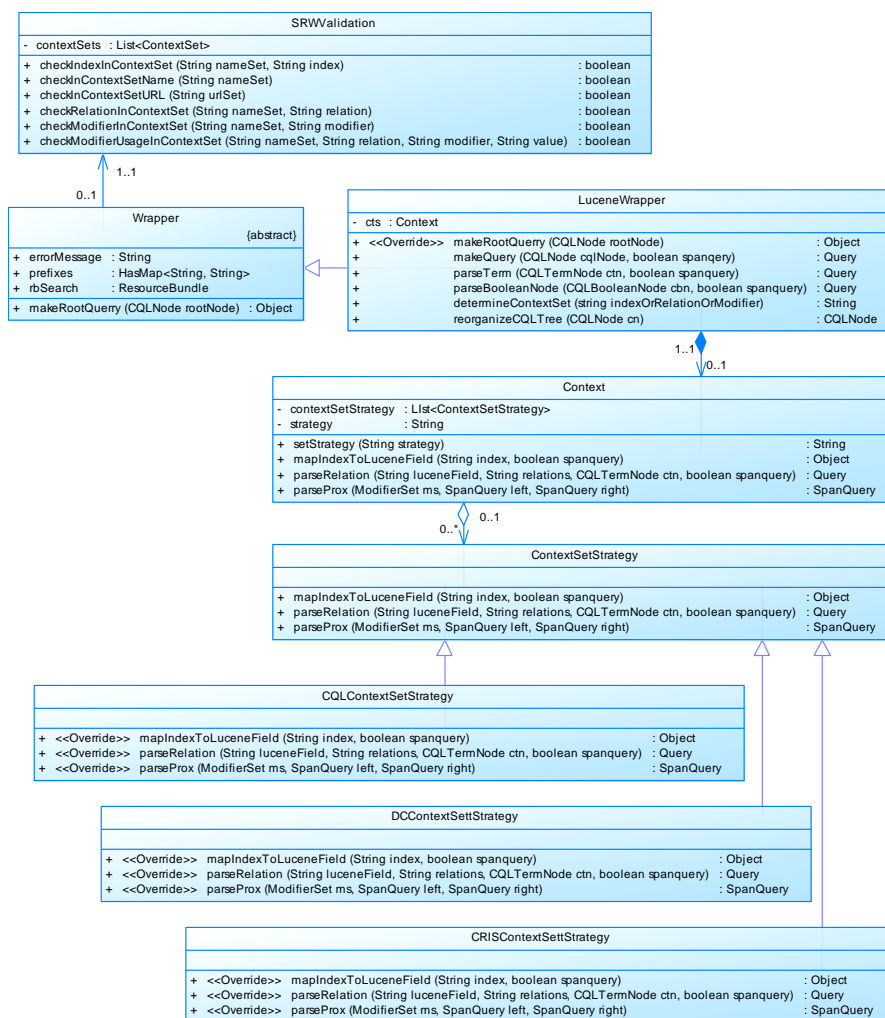
semantičku analizu. Metode apstraktne klase `Wrapper` omogućuju transformaciju objektno reprezentacije CQL upita u odgovarajući oblik upita određen jezikom Tekst Servera. Poziv apstraktne metode `makeRootQuery` klase `Wrapper` kao rezultat vraća objekat Java klase tipa `Object`. Naslednice klase `Wrapper` će pri realizaciji apstraktne metode `makeRootQuery` definisati reprezentaciju upita konkretnog Tekst Servera. Klasa `MARCRecordTextServerManager` (deo komponente *Text Server*) na osnovu upita koji je namenjen Tekst serveru, pribavlja bibliografske zapise o naučno-istraživačkim podacima. Ona enkapsulira rad sa bibliotekom *Lucene*. Klase `RecordConverter` i `Marc21RecordToXMLSerializer` koje pripadaju komponenti *DTO & MARC21*, učestvuju u konverziji pribavljenih zapisa Tekst servera. Upotreba pomenutih klasa zavisi od poziva metoda `getRecordsLocal` i `getRecordsRemote`.



Slika 5.7 Dijagram komponenti sistema za pretragu

Klasa `Wrapper` (slika 5.8) predstavlja polaznu tačku transformacije objektno reprezentacije CQL upita u konkretan upitni jezik Tekst Servera. Klasa `SRWValidation` koja pripada paketu *Context Set* omogućava proširenu proveru semantike (slučaj korišćenja *CRIS profile analysis*) pristiglog CQL upita u skladu sa CRIS profilom. Pomenuta klasa pribavlja informacije o CRIS profilu (nazivi dostupnih indeksa, relacije, modifikatori, kao i njihove moguće vrednosti) kroz obradu XML opisa dostupnih kontekst setova. `LuceneWrapper` predstavlja jednu od mogućih realizacija apstraktne klase

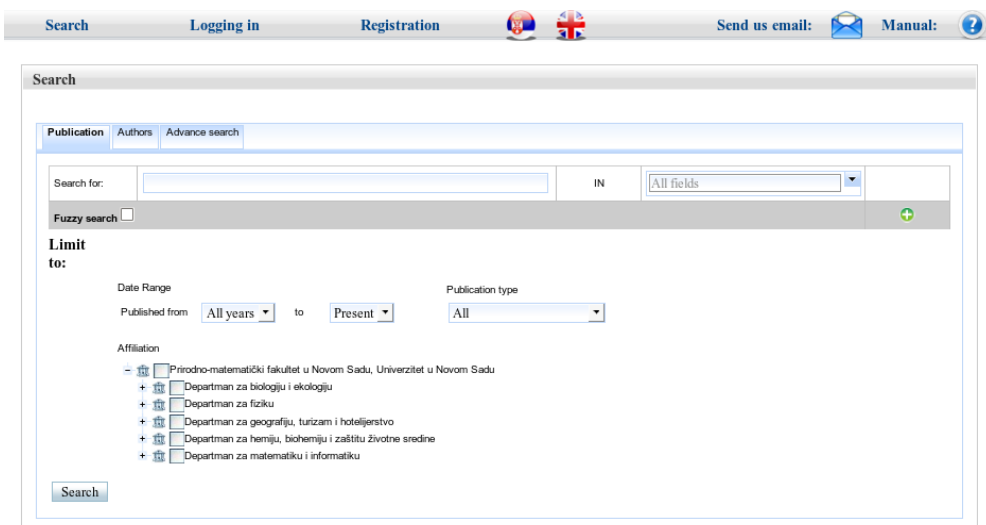
Wrapper u slučaju da komponenta Tekst Server koristi *Lucene* upitni jezik. Metode klase *LuceneWrapper* omogućuju transformaciju binarnog stabla objektivne reprezentacije CQL upita u ekvivalentne strukture *Lucene* sintakse. Klasa *Context* predstavlja upravljačku i kontejner klasu koja sadrži transformacije različitih kontekst setova. *ContextSetStrategy* predstavlja apstraktnu klasu koja definiše okvire za transformaciju CQL upita. Klase naslednice *CQLContextSetStrategy*, *DCContextSetStrategy* i *CRISContextSetStrategy* redefinišu metode klase *ContextSetStrategy* kao strategije za mapiranje CQL, DC i CRIS kontekst setova u strukture *Lucene* sintakse.



Slika 5.8 Dijagram klasa transformacije CQL upita u Lucene upit

5.3. Opis korišćenja aplikacije

U ovom odeljku biće opisan SUI aplikacije CRIS UNS. Specifikacija ovog segmenta opisana je u slučajevim korišćenja u odeljku **5.1 Specifikacija informacionih zahteva**. Trenutno je pretraga u sistemu CRIS UNS omogućena svim korisnicima, nezavisno da li su oni registrovani u okviru sistema ili ne. Na slici **5.9** prikazan je izgled osnovnog prozora. Sistem pretrage podržava višejezičnu predstavu korisničkog interfejsa. Korisniku je u svakom trenutku korišćenja aplikacije omogućena promena jezika interfejsa, bez prethodne potrebe odjave ili resetovanja konekcije ka aplikaciji. Po potrebi, korisnik može odabrati opciju interfejsa Uputstvo koja sadrži osnovne smernice za rad sa samom aplikacijom. SUI CRIS UNS je implementiran u skladu sa preporukama dizajna interfejsa diskutovanim u odeljku **1.2 Aktuelno stanje u oblasti**. Interfejs pretrage je jednostavan i lak za korišćenje, ali i dovoljno moćan da zadovolji potrebe najzahtevnijih korisnika. Imajući u vidu da sistem treba da podrži rad korisnika sa različitim nivoom iskustva i činjenicu da postoje različiti tipovi naučno-istraživačkih podataka celokupan SUI podeljen je u tri odvojene sekcije. Prve dve sekcije su namenjene za standardnu pretragu i podržavaju pretragu publikacija i autora/istraživača. Treća sekcija je namenjena iskusnim korisnicima koji su ovladali kako jezikom pretrage tako i strukturama podataka. Svaka sekcija izdvojena je kao posebna kartica u okviru osnovnog prozora



Slika 5.9 Osnovni prozor aplikacije za pretragu

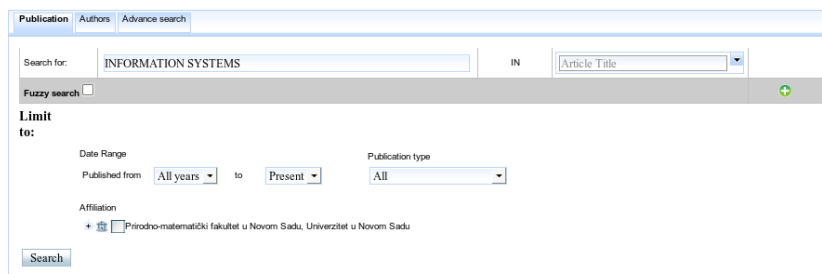
Prva sekcija interfejsa (slika 5.9) namenjena je korisnicima koji žele da pretraže podatke o publikacijama. Funkcionalnost ove sekcije je opisana u slučajevima korišćenja odeljak 5.1.1 *Pretraga publikacija*. Inicijalno u interfejsu postoji svega nekoliko komponenti (jedno polje za zadavanje upite pretrage i nekoliko opcionih polja za limitiranje). Ovakav inicijalni izgled interfejsa napravljen je da zadovolji osnovne potrebe korisnika, gde je akcenat stavljen na preglednost, brzo izvršavanje pretrage i korišćenja sistema pretrage bez potrebe za bilo kakvom obukom. U okviru polja pretrage omogućen je odabir dostupnih CQL indeksa (Penca et al., 2014b) i unos vrednosti terma (slučaj korišćenja Input search term). Dostupni indeksi za svako polje pretrage odabiru se iz padajuće liste (slučaj korišćenja Input search term). Za određene indekse (autori, prvi autor, naziv konferencije i naziv časopisa), prilikom unosa vrednosti terma u polje pretrage, pojavljuje se lista mogućih vrednosti (slučaj korišćenja *Pick author*) čije se vrednosti mogu preuzeti. Sadržaj liste menja se u skladu sa unetim karakteristikama sa tastature, što je rezultat primene AJAX tehnologije. Evidentno je da upotreba liste mogućih vrednosti smanjuje potencijalne greške prilikom unosa terma i dodatno povećava dostupnost samih podataka. Sa obzirom da je SUI baziran na AJAX tehnologiji, korisniku je omogućeno da promeni inicijalni izgled interfejsa uključivanjem dodatnih polja pretrage, bez potrebe da se tekuća stranica ponovno očita sa servera. Zadavanje proizvoljnog broja polja pretrage kao i njihova kompozicija, omogućuje korisniku stvaranje kompleksnih i preciznijih upita. Nedvosmisleno je da ovakav dinamički izgled interfejsa podstiče korisnike da istražuju opcije sistema (vremenom će korisnici pokušati da kreiraju preciznije upite pretrage), što doprinosi njihovom usavršavanju i povećava njihovu želju za korišćenjem celokupnog sistema. Kompozicija između polja interfejsa je predstavljena AND, OR i NOT operatorima CQL logičkog upita, kojima se definiše međusobni odnos polja u upitu. Uvidevši da određeni indeksi pretrage poseduju predefinisani skup vrednosti, neophodno je bilo njihovo izdvajanje u zasebne opcije interfejsa (polja za limitiranje pretrage sa rečnikom dozvoljenih vrednosti). U prikazanom sistemu predefinisani skupovi vrednosti su: vremenski period, vrsta publikovanog materijala i organizaciona struktura naučno-istraživačkih institucija. Pretraga se može limitirati upotrebom stavki interfejsa period, tip publikacije i pripadnost. Pored kreiranja kompleksnih upita, interfejs podržava korišćenje naprednih funkcionalnosti pretrage kao što su pretraga

fraza, nedovršenih reči/rečenica (upotreba džoker karaktera) i približna (*fuzzy*) pretraga.

U nastavku će se sa nekoliko ilustrativnih primera opisati deo mogućnosti i specifičnosti CRIS UNS sistema pretrage.

Primer 1. Korišćenjem sistema pretrage pronaći sve publikacije koje u svom naslovu sadrže reč INFORMATION i reč SYSTEMS. Pomenutu pretragu moguće je realizovati (formulisati) na više različitih načina (varijanti).

- a) Na slici **5.10** navedena je jedna od mogućih formulacija upita pretrage koja će prikazati sve rezultate koji sadrže tražene reči ne uzimajući u razmatranje njihov položaj (udaljenost između traženih reči) i redosled navođenja (slika **5.10**) u rezultatima. Konkretno, sistem pretrage će vratiti isti broj pogodaka (10 pogodaka) bez obzira ako se za vrednost terma navede INFORMATION SYSTEMS ili SYSTEMS INFORMATION.
- b) Slika 11 opisuje proširenje upita navedenog pod „1.a“ sa opcijom fazi pretrage. U ovom slučaju dobija se znatno veći broj rezultata (63 pogodaka) u odnosu na primer pod „1.a“. Veći broj pogodaka je očekivan obzirom da se prilikom fazi pretrage prikazuju svi rezultati koji zadovoljavaju kriterijum sličnosti (slučaj korišćenja Formulate fuzzy query). Svaki od rezultata se mora minimalno 80% poklapati u odnosu na zadati term INFORMATION SYSTEMS. Kako u CRIS UNS sistemu postoje višejezični podaci, u rezultatima pretrage srećemo sledeće varijacije: *information systems*, *information system*, *informacioni sistemi*, *informacioni sistem*, *informacionog sistema*. Uviđa se da primenom fazi pretrage sistem pronalazi podatke bez obzira na gramatičke osobine reči u srpskom jeziku (padež, rod i broj).



The screenshot shows the 'Advance search' interface of the CRIS UNS system. At the top, there are tabs for 'Publication', 'Authors', and 'Advance search'. The main search area contains a 'Search for:' field with the text 'INFORMATION SYSTEMS', an 'IN' field with a dropdown menu set to 'Article Title', and a 'Fuzzy search' checkbox which is currently unchecked. Below this, there is a 'Limit to:' section with several filters: 'Date Range' with 'Published from' set to 'All years' and 'to' set to 'Present'; 'Publication type' set to 'All'; and 'Affiliation' with a plus sign icon and the text 'Prirodno-matematički fakultet u Novom Sadu, Univerzitet u Novom Sadu'. A 'Search' button is located at the bottom left of the form.

Slika 5.10 Izgled pretrage za primer 1a

Publication Authors Advance search

Search for: INFORMATION SYSTEMS IN Article Title

Fuzzy search

Limit to:

Date Range
Published from: All years to Present Publication type: All

Affiliation
+ Prirodno-matematički fakultet u Novom Sadu, Univerzitet u Novom Sadu

Search

Slika 5.11 Izgled pretrage za primer 1b

Ako za zadati upit postoje relevantni podaci u sistemu, rezultati pretrage imaju oblik kao na slici 5.12. Sa slike se uočava da su rezultati u skladu sa Harvardskim stilom referenciranja (*The Harvard system of referencing* - <http://libweb.anglia.ac.uk/referencing/harvard.htm>), stoga bi se oni u vrlo lako mogli koristiti kao "pomoćni alat" prilikom unosa referenci. Zbog bolje preglednosti ukoliko je broj rezultata veći od 100, u tabeli će inicijalno biti prikazano prvih 100 rezultata, dok će svi ostali biti vidljivi upotrebom opcija kretanja kroz listu koje se vizuelno nalaze ispod tabele. Ukoliko je broj rezultata manji od 100 u tabeli su već prikazani svi rezultati pretrage i stoga ne postoje opcije kretanja kroz listu.

Search results

Number of results is 63

RAKIĆ, M., BODROŠKI, Ž. and SKRBIC, S. (2008) An Application of the GWT in Faculty Information System. In: <i>Zbornik radova 52. konferencije za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku, Palić, 2008</i>
IVANOVIĆ, M., TODORIĆ-VUKAŠIN, D. and BUDIMAC, Z. (2009) Information system topics for studies in software engineering. In: <i>Conference on Computer Systems and Technologies (CompSysTech'09), 2009</i> , pp. IV.151-IV.154
IVANOVIĆ, M., TODORIĆ-VUKAŠIN, D. and BUDIMAC, Z. (2009) INFORMATION SYSTEM TOPICS FOR STUDIES IN SOFTWARE ENGINEERING. In: <i>Conference on Computer Systems and Technologies (CompSysTech'09), 2009</i> , pp. IV.151-IV.154
TOŠIĆ, T. (1998) <i>Razvoj informacionih sistema baziranih na strukturiranim dokumentima.</i> (Магистарска теза). Prirodno-matematički fakultet u Novom Sadu
KONJOVIĆ, Z. and SURLA, D. (2003) An Overview of the Development of the Library Information System BISIS. In: <i>Proceeding of the 4th International Conference on Informatics and Information Technology, Bitola, 2003</i>
SURLA, D. and KONJOVIĆ, Z. (1996) The Solution of the Library Information System of The University of Novi Sad. In: <i>Proceedings of Informatics and Social Changes, Zrenjanin, 1996</i> , pp. 46-53
SURLA, D. and KONJOVIĆ, Z. (1995) Distribuirani unos podataka u bibliotečkom informacionom sistemu. In: <i>Zbornik apstrakata SINFON'95, Zlatibor, 1995</i> . Beograd: FON, p. 2
СУРПА, Д. and КОЊОВИЋ, З. (2004) Преглед развоја библиотеког информационог система БИСИС. In: СУРПА, Д. and КОЊОВИЋ, З. (eds.) <i>Дистрибуирани библиотеку информациони систем БИСИС</i> . Нови Сад: Група за информационе технологије, pp. 29-41
SURLA, D. and TRIKOVIĆ, I. (1997) Object-Oriented Methods for Information Systems Design. <i>Informatics and Information Technologies – Theory and Practice</i> , pp. 242-251
IVANOVIĆ, M. et al. (2010) Advances in Databases and Information Systems at the University of Novi Sad. In: IVANOVIĆ, M et al. (eds.) <i>Local proceedings of the 14th Conference on Advances in Databases and Information Systems, Novi Sad, 2010</i> , pp. 190-204
MILOSAVIJEVIĆ, B., VUKAČKOVIĆ, M. and SIRIĆ, D. (2007) višeznačajnim tekstem u Bibliotečkom informacionom sistemu BISIS, ver. 3.0. In: <i>Zbornik radova Inform i čitilica: smisli i svrhi sistema i</i>

Slika 5.12 Primer dela rezultata pretrage

Primer 2. Korišćenjem sistema pretrage pronaći sve publikacije u kojima se Dragan Ivanović navodi kao prvi autor i gde je jedan od mogućih koautora Dušan Surla. Ovaj primer podrazumeva da se imena autora preuzimaju iz liste mogućih vrednosti (slika 5.13). Lista mogućih vrednosti sadrži listu autora čiji nazivi ispunjavaju kriterijum sličnosti sa unešenim nazivom

(slučaj korišćenja *Pick Author*). Po odabiru imena autora iz ponuđene liste, polje za unos teksta se popunjava sa kompletnim imenom i prezimenom npr. "Surla Dušan"(slika 5.14). Odabirom imena autora iz liste mogućih vrednosti preuzeta vrednost pretražuje se kao fraza.

The screenshot shows a search interface with tabs for 'Publication', 'Authors', and 'Advance search'. The 'Search for:' field contains '"Ivanović Dragan"'. Below it, a dropdown menu is open, listing several author names: 'Surla D', 'Surla Dušan; Сурла Душан; Surla Dusan', 'Dimić Surla Bojana; Димић Сурла Бојана; Dimić Bojana; Димић Бојана', 'Skala Dejan', and 'Skala U.D'. The interface also includes an 'AND' operator, a 'Fuzzy search' checkbox, and a 'Limit to:' section.

Slika 5.13 Primer korišćenja liste predloženih vrednosti

The screenshot shows a search interface with tabs for 'Publication', 'Authors', and 'Advance search'. The 'Search for:' field contains '"Ivanović Dragan"'. Below it, a dropdown menu is open, listing several author names: 'Surla D', 'Surla Dušan; Сурла Душан; Surla Dusan', 'Dimić Surla Bojana; Димић Сурла Бојана; Dimić Bojana; Димић Бојана', 'Skala Dejan', and 'Skala U.D'. The interface also includes an 'AND' operator, a 'Fuzzy search' checkbox, and a 'Limit to:' section with filters for 'Date Range', 'Publication type', and 'Affiliation'. A 'Search' button is visible at the bottom.

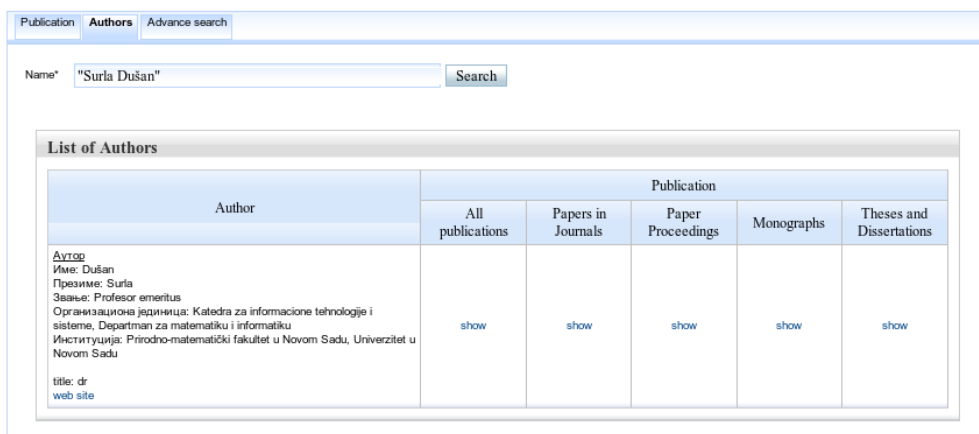
Slika 5.14 Izgled pretrage za primer 2

Primer 3. Korišćenjem sistema pretrage pronaći sve publikacije u kojima podaci o prvom autoru sadrže reč Ivanović i gde je jedan od koautora Dušan Surla (slika 4.15). U ovom slučaju samo se ime autora Dušan Surla preuzima iz liste mogućih vrednosti. U rezultatima pretrage Dušan Surla sreće kao koautor, dok se za prvog autora identifikuju Dragan Ivanović ili Lidija Ivanović ili Mira Ivanović. Pojavljivanje više različitih instanci prvog autora posledica je specifikacije samo dela podataka o autoru.

Slika 5.15 Izgled pretrage za primer 3

Druga sekcija interfejsa namenjena je korisnicima koji pretražuju podatke o istraživačima (autorima). Podaci o istraživačima pretražuju se unosom ličnom imena i/ili prezimena. Nalik pretrazi publikacija, postoji mogućnost korišćenja liste mogućih vrednosti. U okviru ove sekcije dozvoljena je pretraga nedovršenih reči/rečenica (upotreba džoker karaktera), kao i pretraga fraza. Pretraga se inicira aktiviranjem dugmeta Traži. Rezultati pretrage se prikazuju tabelarno jer više autora može zadovoljiti definisani kriterijum. Tabelu rezultata čine dva veća segmenta, prvi koji sadrži opšte podatke o autoru i drugi u kome se nalaze veze ka autorovim istraživačkim rezultatima. Prikazani opšti podaci o autoru su: ime i prezime autora, pripadnost instituciji i/ili organizaciji, titula i lični veb sajt. Veze ka istraživačkim rezultatima autora podeljene su po sledećim podsekcijama: *sve publikacije*, *radovi u časopisu*, *radovi na konferencijama*, *monografije i teze i disertacije*. Svaka podsekcija sadrži link koji pri aktivaciji rezultuje zapisima.

Primer 4. Korišćenjem sistema pretrage pronaći sve istraživače/autore koji u svom nazivu sadrže ime Surla Dušan (slika 5.16). U rezultatima pretrage nalaze se samo podaci o jednom autoru. Korišćenjem linkova iz tabela moguće je izlistati određene tipove publikacija odabranog autora.



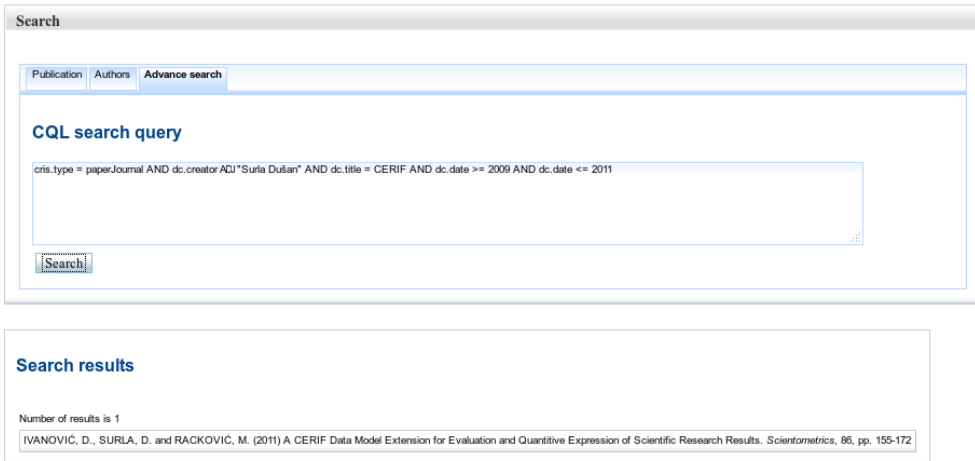
Slika 5.16 Izgled pretrage za primer 4

Трета секција интерфејса намењена је искуснијим корисницима, путем које је омогућена претрага свих типова записа у оквиру CRIS система и путем које је могуће креирати изразито сложене упите. Записи који се претражују односе се на све могуће податке о научно-истраживачкој делатности. Све напредне функционалности система претраге доступне су у овој секцији интерфејса. Претрагу по удаљености између термина могуће је користити само у оквиру ове секције интерфејса, што се оправдава тврђњом да је то ретко коришћена опција у претрази коју користе само искусни корисници. Ова секција је реализована једним текст полјем у коме корисник уноси текстуални упит базиран на спецификацији CQL језика и подржаног CRIS профила претраге. За коришћење ове функционалности неопходно је познавање синтаксе CQL језика и пожељно је познавање CRIS профила (поглавље 3).

Пример 5. Коришћењем система претраге пронаћи све публикације који у свом називу садрже назив CERIF, чији је аутор Surla Dušan, које су публиковане од 2009 до 2011 године (слика 4.17).

CQL упит који би одговарао примеру 5 је:

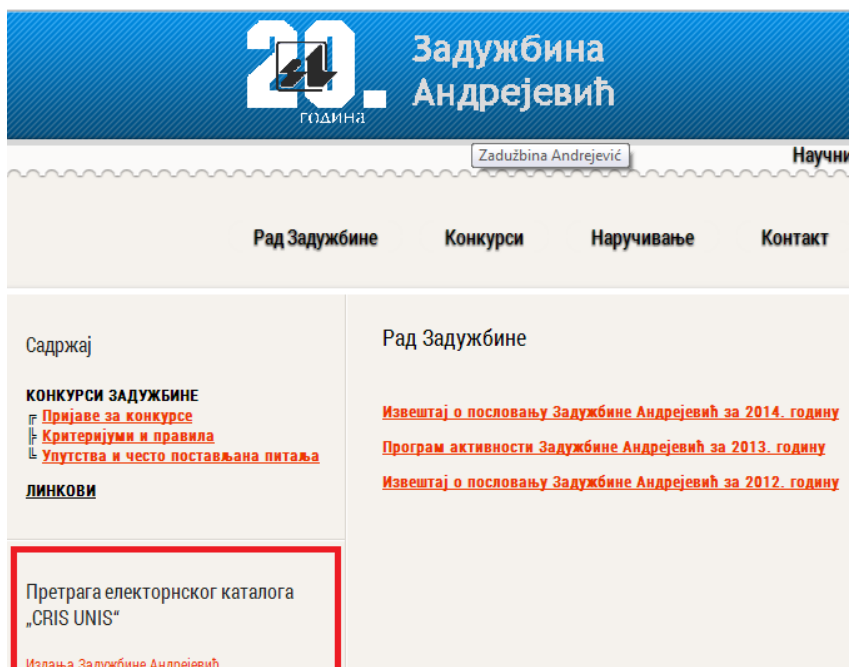
```
cris.type=paperJournal AND dc.creator ADJ "Surla Dušan" AND dc.title = CRIS AND dc.date >= 2009 AND dc.date <= 2011
```



Slika 5.17 Izgled pretrage za primer 5

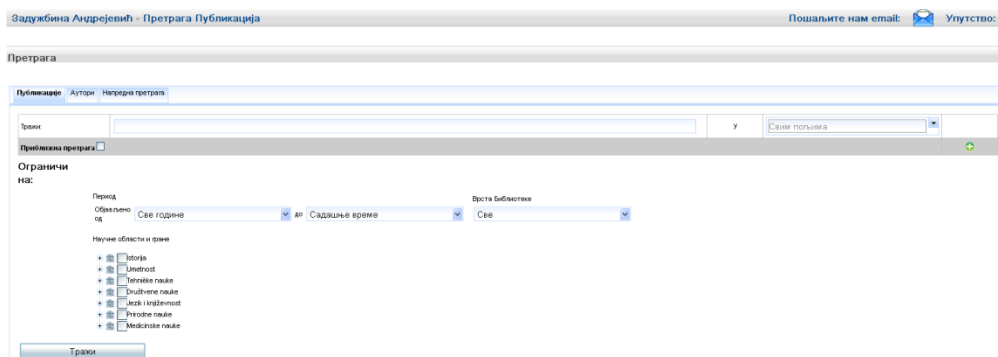
5.4. Implementacija podsistema za Zadužbinu Andrejević

Kao dobar primer jednostavne modifikacije modularnog sveobuhvatnog sistema pretragu za pretragu naučno-istraživačkih podataka, prikazan je kratak opis implementacije pretrage podataka Zadužbine Andrejević. Zadužbina Andrejević predstavlja izdavačku instituciju čiji je fokus izdavanje naučnih monografija, gde je objavljeno 1.100 naučnih monografija, u tiražu od blizu 280 hiljada primeraka. Rukovodeći se činjenicom da je najteže javnosti predstaviti prvo verifikovano naučno delo – magistarsku tezu, doktorsku disertaciju, specijalistički rad ili naučnu studiju, profesori dr Dušanka i dr Kosta Andrejević osnovali su Zadužbinu 1994. godine radi pružanja sistematske i trajne podrške i pomoći perspektivnim naučnicima. Implementirani podsistem pretrage za javno je dostupan na putem sajta zadužbine (*Zadužbina Andrejević* - <http://zandrejevic.rs/>) (slika 5.18).



Слика 5.18 Link ka podsystemu za pretragu podataka Zadužbine Andrejević

Nakon klika na link otvara se stranica za pretragu monografija Zadužbine Andrejević (slika 5.19). Interfejs je realizovan na bazi Osnovni prozor aplikacije za pretragu sa slike 5.9, obzirom da smo već pokazali intuitivnost i prednosti pomenutog interfejsa kod pretrage podataka iz naučno-istraživačke delatnosti.



Слика 5.19 Osnovni prozor aplikacije za pretragu podataka Zadužbine Andrejević

Ponovo je SUI podeljen u tri odvojene sekcije. Prve dve sekcije su namenjene za standardnu pretragu i podržavaju pretragu monografija i autora istih. Treća sekcija je namenjena iskusnim korisnicima koji su ovladali kako

jezikom pretrage tako i strukturama podataka. Svaka sekcija izdvojena je kao posebna kartica u okviru osnovnog prozora. Prva sekcija interfejsa namenjena je korisnicima koji žele da pretraže podatke o publikacijama. Za razliku od interfejsa sa slike 5.9 u polju za pretragu unose se vrednosti za sledeće indekse: *Sva polja*, *Naslov/Apstrakt/Ključne reči*, *Autorima*, *Prvom autoru*, *Naslovu*, *Apstraktu* i *ključnim rečima*. Pored perioda u kojem je monografija objavljena, jedan od postojećih filtera je *tip biblioteke* kojoj pripada monografija. U bazi Zaduzbine Andrejević monografije su grupisane po sledećim bibliotekama: *DISSERTATIO*, *ACADEMIA*, *POSEBNA IZDANJA*, *SPECIALIS*, *INITIUM*, *Edicija naučnici Srbije*, *INSPIRATIO*, *EDUCATIO*, *Zbornici*, *Zelena linija života*. U okviru svojih biblioteka *DISSERTATIO* i *ACADEMIA* Zaduzbina objavljuje priređene doktorske i magistarske radove. Biblioteka *POSEBNA IZDANJA* pruža mogućnost objavljivanja naučnih dela već afirmisanim naučnicima iz zemlje i inostranstva, dok se u biblioteci *SPECIALIS* objavljuju specijalistički radovi, a u biblioteci *INITIUM* master radovi. *Edicija naučnici Srbije* predstavlja javnosti podatke o doktorima nauka, magistrima i specijalistima u našoj zemlji od osnivanja Beogradskog univerziteta 1905. godine. Podaci se iskazuju po godinama i kumulativno. Biblioteka *INSPIRATIO* donosi posebno vredne naučne radove ili informacije o njima, a biblioteka *EDUCATIO* prezentira radove s velikom upotrebnom vrednošću u praksi. U biblioteci *Zbornici* objavljuju se radovi većeg broja autora u okviru iste ili više tematskih oblasti. *Zelena linija života* je biblioteka predviđena za objavljivanje dela naučne, stručne i praktične sadržine o racionalnom korišćenju i negovanju prirodnih potencijala primenom savremene tehnologije. Objavljena dela imaju visoku upotrebnu vrednost, naročito u praksi. Monografije su pripadaju određenoj *naučnoj oblasti* ili *grani* što je jedan od dostupnih filtera u indeksu. Treba napomenuti da Naučna oblast sadrži više naučni grana (npr. naučna oblast *Umetnost* se deli na naučne grane *muzika*, *strip*, *teatrologija*, *istorija umetnosti*, *film* i *primenjena umetnost*). Rezultati pretrage (slika 5.19) imaju sličan izgled kao na slici 5.12 sa razlikom da pored autora monografije stoji i ime biblioteke kojoj pripada monografija. Pretraga po autorima izgleda identično kao na slici 5.16 s tim što se prikazuju samo rezultati koji se odnose na monografije.

Rezultati pretrage

Broj rezultata pretrage je 12

RADOVIĆ, B. (2009) <i>Preživljavanje filma. INITIUM</i> . Beograd: Zadužbina Andrejević
DRAGUTINOVIĆ KOMATINA, S. (2004) <i>Konzervacija zlatnih slika u grobnicama i pećinama</i> . ACADEMIA. Beograd: Zadužbina Andrejević
ČALOVIĆ, D. (2006) <i>Josip Broz Tito: studija imaloža</i> . ACADEMIA. Beograd: Zadužbina Andrejević
NIKOLIĆ, M. (2006) <i>Radio u Srbiji (1924-1941)</i> . ACADEMIA. Beograd: Zadužbina Andrejević
ĐURKOVIĆ, A. (2011) <i>U traganju za izgubljenom Istinom. POSEBNA IZDANJA</i> . Beograd: Zadužbina Andrejević
BOJIĆ, Z. (2007) <i>Stanislav Rapotec, a Barbarogenius In Australian art. POSEBNA IZDANJA</i> . Beograd: Zadužbina Andrejević
BOJIĆ, Z. (2007) <i>Imaginary homelands, the art of Danila Vassiliev. POSEBNA IZDANJA</i> . Beograd: Zadužbina Andrejević
MARIĆ, D. (2003) <i>Pozorišna produkcija. POSEBNA IZDANJA</i> . Beograd: Zadužbina Andrejević
PETROVIĆ, M. (2000) <i>Oslobođenje drame</i> . ACADEMIA. Beograd: Zadužbina Andrejević
GRUŠANOVIĆ, Z. (2010) <i>Tri drame Ljubomira Simovića</i> . ACADEMIA. Beograd: Zadužbina Andrejević
TUCAKOV, A. (2000) <i>Strip u Srbiji (1975-1995)</i> . ACADEMIA. Beograd: Zadužbina Andrejević
MACURA, N. (2006) <i>Kulturna politika u oblasti muzike</i> . ACADEMIA. Beograd: Zadužbina Andrejević

Slika 5.19 Rezultati pretrage podataka iz Zadužbine Andrejević

6. Implementacija SRU/W standarda u CRIS UNS sistemu

Predmet ovog odeljka je implementacija servisa pretrage (Penca et al., 2014a) podataka naučno-istraživačke delatnosti baziranog na SRU/W standardu. Sama implementacija nastavlja praksu postojećeg sistema za unos podatka (CRIS UNS) koristeći samo tehnologije otvorenog koda. SRU/W servis pretrage je modularan i dozvoljava da se pojedine komponente u sistemu pretrage koriste i realizuju na više različitih načina, usled čega postoje mogućnosti za:

- razvijanje eksterne aplikacije za pretragu, koja bi se bazirala na SRU/W bibliotečkom standardu.
- jednostavnu izmenu Tekst servera, koja ne bi uticala na komponentu *SRU/W server side*.
- realizaciju uporedne pretrage više Tekst servera čija implementacija i fizička lokacija može biti različita. U tom slučaju, potrebno je samo definisati odgovarajuća mapiranja CQL jezika na jezike korišćenih Tekst servera.
- potencijalnu interoperabilnost sa različitim bibliotečkim sistemima jer je SRU/W de fakto standard u ovim sistemima.

Iz dijagrama komponenti sa slike 5.6 (implementacija arhitekture sistema) se jasno vidi da komunikacija klijenta/servera po SRU/W standardu počinje od komponente *Client SRU/SRW - SRU/W Interface* a nastavlja se u *SRU/W Server Side*. O njima je bilo detaljno reči u poglavlju statički deo sistema. Standard SRU/W ne sugeriše tehnologije za implementaciju entiteta koji komuniciraju, već naprotiv podrazumeva da oni budu nezavisni „servisi“ kojima je samo bitno da poznaju principe funkcionisanja SRU/W standarda. Kako je postojeća aplikacija CRIS UNS bazirana na veb tehnologijama odlučeno je da u ovom sistemu klijentska i serverska strana pretrage implementiraju kao veb servisi. Kako je čitav sistem baziran na java programskom jeziku tako su i tehnologije korišćene za razvoj Veb servisa bazirane na programskom jeziku Java.

U narednoj tabeli dat je kratak pregled tehnologija, specifikacija i frejmvorka kojima se mogu realizovati veb servisi java programskim jezikom.

Ime	Opis	Prednosti	Mane
RMI	<i>Java Remote Method Invocation</i> (Java RMI) je Java API koji omogućuje objektno orijentisani pristup pozivanju udaljenih procedura (RPC – <i>Remote Procedure Call</i>) sa podrškom za direktan transport serijalizovanih objekata i njihovo uklanjanje sa udaljenih lokacija.	Komunikacija između entiteta koje koriste različite Java virtualne mašine. Tehnologija je dostupna još od JDK1.02.	Vrlo komplikovana implementacija. Zastareo koncept. Limitiran na entitete koji poseduju java virtualnu mašinu.
CORBA	<i>Common Object Request Broker Architecture</i> (CORBA) je tehnologija razvijena za potrebe distribuiranih sistema. CORBA servisi su opisani interfejsima koji su napisani u specifičnom IDL jeziku (Interface Definition Language)	Nije ograničena na entitete koji su napisani u Java programskom jeziku, što rezultira visokim stepenom interoperabilnosti. Interfejs definisan IDL može se iskoristiti za različite implementacije.	Implementiranje servisa zahteva učenje novog jezika IDL. Ne podržava slanje kompletnog programskog koda.
JAX-RPC	<i>Java API for XML-based RPC</i> (JAX-RPC) je XML specifikacija koja omogućuje Java aplikacijama da pozovu Java veb servise koji su opisani WSDL gramatika.	Veb servis definisan ovom specifikacijom može biti implementiran na serverskoj strani kao Servlet ili čak EJB kontejner. Omogućuje pristup servisima koji podržavaju RPC iako nisu napisani u Javi.	JAX-RPC 1 biva napušten sa Java EE 6, jer se sve ređe SOAP veb servisi realizuju kao RPC.
JAX-WS	The Java API for XML based Web Services (JAX-WS) je naslednik JAX-RPC specifikacije. JAX-WS je orijentisana na razmeni poruka i podržava asinhronu komunikaciju. Koristi dodatno JAXB i SAX za predstavljanje objekata i parsiranje	Skup svih neophodnih tehnologija za komforno rukovanje veb servisima u javi. Sve tehnologije je propisalo telo W3C (HTTP, SOAP i WSDL). Bazira se na razmeni XML poruka. Ubedljivo se najviše sreće kod sistema koji imaju podršku za Java	Previše tehnologija koje treba poznavati da bi se adekvatno implementirao veb servis. XML poruke unose usporenje ako imamo veliki broj entiteta u komunikaciji ili ako su velike poruke u pitanju. Ne može da šalje čitave fajlove u

Ime	Opis podataka.	Prednosti veb servise.	Mane porukama.
Apache Axis	Apache Axis (Apache eXtensible Interaction System) je XML bazirani veb servis frejmvork.	Omogućuje entitetima u komunikaciji razmenu SOAP poruka. U potpunosti podržava JAX-RPC specifikaciju.	Napravljeni servisi se teško mogu modifikovati. Podržava samo RPC i WSDL do verzije 1.1
Apache Axis2	Redizajn Apache Axis frejmvorke.	Apache Axis2 podržava SOAP 1.1 i SOAP 1.2. Posедуje integrisanu podršku za REST veb servise. Direktna podrška za različite veb frejmvorke. npr. Spring Framework.	Sve manje se razvija, poslednja verzija je distribuirana početkom 2012.
CXF	CXF je takođe projekat Apache softver fondacije, a nastao je kao spoj dva projekta "Celtix" and "XFire".	Potpuna separacija izvornog koda od „front end“ (kao JAX-WS). Visoke performanse i mogućnost kombinacije sa različitim tehnologijama; CORBA, JAVASCRIPT Mogućnost automatskog generisanja servisa iz specifikacija ili java koda.	Nije u potpunosti interoperabilan sa servisima koji nisu implementirani Java tehnologijama. Dodatne varijante vezane za bezbednost nisu u potpunosti podržane.
Metro	Metro je projekat Oracle Corporation i deo je GlassFish Aplikacionog Servera.	U sebi sadrži različite standarde i specifikacije: JAXB RI, JAX-WS RI, SAAJ RI i WSIT.	Može se koristiti i kao deo drugih servera (da nije GlassFish) ali nije preporučljivo.
Web Services Interoperability Technology (WSIT)	Web Services Interoperability Technology (WSIT) je projekat otvorenog koda Sun Microsystems. Obezbeđuje interoperabilnost između Java Web Servisa i Microsoft's Windows	Java programming language APIs koji omogućuju upotrebu Java veb servis specifikacija u .NET aplikacijama. WSIT pruža niz ekstenzija za osnovnu varijantu SOAP protokola, kao i mogućnost upotrebe JAX-WS i JAXB.	Komplikovana implementacija.

Ime	Opis	Prednosti	Mane
	Communication Foundation (WCF).		
XML Interface for Network Services (XINS)	XML Interface for Network Services (XINS) je tehnologija otvorenog koda za definisanje i implementaciju internet aplikacija, koja se zasniva na pristupu gde je neophodno da se prethodno formira adekvatna specifikacija servisa.	Iz specifikacije se formira: HTML dokumentacija, test forme, WSDL, unit test kod (u Javi) stabovi (in Java) klijenti (in Java)	Sama tehnologija zahteva poznavanje različitih tehnologija POX [], log4j []. Ograničena samo na RPC.
Web Services Invocation Framework	The Web Services Invocation Framework (WSIF) sadrži jednostavan Java API za povezivanje Web servisa, bez obzira kako su ti servisi nastali.	Frejmworok omogućuje maksimalnu fleksibilnost kada se pozivaju servisi opisani WSDL specifikacijom.	Ne razvija se od 2003 godine.
JBossWS	JBossWS je JBoss implementacija J2EE kompatibilnih web servisa.	Ovaj framework je dizajniran kako bi bio u potpunosti u skladu sa JBoss tehnologijama. Redovno se revidira i poslednja aktuelna verzija je iz marta 2014.	Mora se koristiti JBoss aplikativni server.
Hessian	Hessian je jednostavan binarni protokol koji se koristi za komunikaciju između veb servisa.	Ovaj protokol ne zahteva korišćenje glomaznih frejmvorka. Brzina prenosa podataka je na visokom nivou.	Neophodno je naučiti same detalje protokola. Brzina rapidno opada kada se poveća broj servisa u komunikaciji.

Tabela 6.1 Java tehnologije za implementaciju veb servisa

Iz pregleda dostupnih realizacionih okvira veb servisa i prirode korisnika koji bi koristili servise sistema pretrage naučno istraživačkih podataka došlo se do sledećih zaključaka:

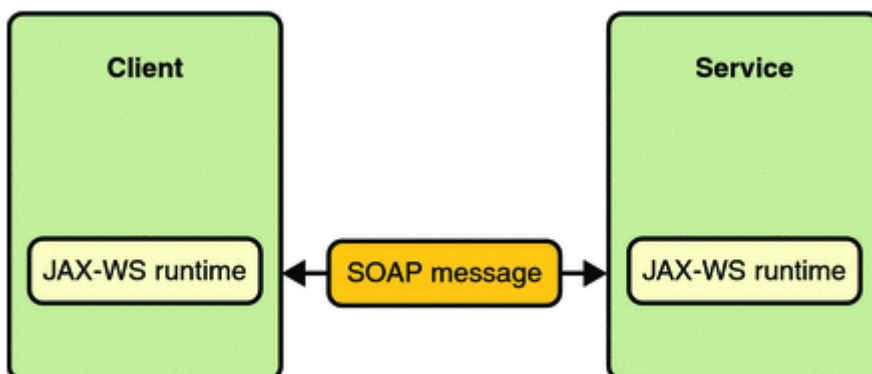
- veb servis treba da se zasniva na opšte prihvaćenim standardima kao što su: XML, SOAP, WSDL

- XML bazirane tehnologije su neophodne kako bi se napravio preduslov za jednostavnije povezivanje sa drugim delovima CRIS UNS sistema
- kako bi se omogućila lakša implementacija i kasnije održavanje, bilo bi dobro koristiti neki već dobro razrađeni frejmovork za implementaciju veb servisa.

Shodno navedenim zaključcima za implementaciju veb servisa korišćen je *JAX-WS* i *Apache CXF* frejmovork. U nastavku su opisane korišćene tehnologije i njihova primena u implementaciji veb servisa sistema za pretragu naučno-istraživačkih podataka.

6.1. JAX-WS

Java API for XML Web Services (JAX-WS), poznata i kao JSR-224, predstavlja set Java tehnologija koje se koriste za razvoj veb servisa. Po rečima *Sun Microsystem* JAX-WS pripada grupi bazičnih tehnologija za implementaciju Web servisa. Kao i većina tehnologija iz ove grupe, JAX-WS se najčešće koristi u kombinaciji sa drugim tehnologijama iz pomenute grupe, npr JAXB. Kako JAX-WS obuhvata niz tehnologija vrlo često se u literaturi koristi kao „okvir“ za razvoj veb servisa. JAX-WS omogućava pozive udaljenih procedura ili razmenu poruka upotrebom nekog od XML baziranih protokola kao što je SOAP, s tim što "sakriva" kompleksnost SOAP protokola iza dobro definisanog Java API. Programeri koriste API kako bi definisali metode i neophodne klase i stvorili neophodne uslove da se omogući komunikacija između servisa upotrebom JAX-WS API. JAX-WS konvertuje klijentske pozive kroz API u SOAP poruke (slika 6.1). JAX-WS sadrži programski model zasnovan na anotacijama koji pojednostavljuje razvoj veb servisa. JAX-WS se može koristiti u JAVA SE počevši od verzije 6. JAX-WS je zamenila JAX-RPC API u JAVA EE verziji 5 zbog trenda da se sve više koriste document style (Kalin, 2009) veb Servisi. JAX-WS se učestalo koristi u raznim frejmorcima za razvoj Veb servisa od kojih su najpoznatiji: Apache CXF i Project Metro.



Slika 6.1 Komunikacija u JAX-WS okviru

JAX-WS tehnologija je odabrana za razvoj veb servisa u sistemu CRIS UNS iz sledećih razloga:

- JAX-WS je vodeća tehnologija za izradu veb servisa baziranih na programskom jeziku Java.
- Pratimo dobru praksu u izradi CRIS UNS aplikacije gde se implemetacija zasniva na otvorenom kodu Java programskog jezika.

Postoje dva pristupa u razvoju veb servisa upotrebom JAX-WS tehnologije:

- Razvoj JAX-WS veb servisa iz Java *Bean* klasa (pristup odozdo ka gore - bottom-up development). Prilikom razvoja veb servisa iz Java Bean, može se iskoristiti postojeća klasa kojoj samo treba dodati neophodne stavke za razvoj JAX-WS veb servisa. Korišćenje anotacija kvalifikuje Java Bean kao deo implementacije veb servisa. Kod ovog pristupa, nije neophodno napisati WSDL fajl, jer se anotacijama u java klasama definišu sve neophodne informacije za servis (servisnu pristupnu tačku) ili klijenta.
- Drugi pristup podrazumeva kreiranje JAX-WS veb servisa iz postojećeg WSDL fajla korišćenjem JavaBeans (pristup odozgo na dole - top-down development). WSDL dokument se može samostalno konstruisati ili preuzeti iz drugih izvora (veb, UDDI registar ...).

U ovoj disertaciji izabran je drugi pristup (top-down development) gde se za kreiranje veb servisa koristi WSDL fajl.

WSDL predstavlja XML dokument koji se koristi za opis veb servisa, odnosno za opis operacija koje se mogu pozivati i struktura podataka koja se

razmenjuju u komunikaciji. WSDL (*Web Services Description Language*) skraćena ukazuje da je u pitanju specifičan jezik za opis veb servisa koji se bazira na XML, što je i njegova velika predanost obzirom da nije vezan ni za kakvu programsku implementaciju. Aktivno se koriste verzije 1.1 i 2.0. WSDL dokument se sastoji od osam elemenata: *definitions*, *types*, *message*, *operation*, *portType*, *binding*, *port*. U nastavku su prikazani elementi WSDL specifikacije koja je korišćena za razvoj SRU/W servisa za pretragu naučno-istraživačkih podataka.

Element *definitions* predstavlja korenski element WSDL dokumenta i često se koristi za globalne *namespace* deklaracije (Listing 6.1).

```
<definitions xmlns:srw-bindings="http://www.loc.gov/zing/srw/srw-bindings/"
              xmlns="http://schemas.xmlsoap.org/wsdl/"
              xmlns:xsd="http://www.w3.org/2001/XMLSchema"
              xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
              xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
              xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
              targetNamespace="http://www.loc.gov/zing/srw/srw-sample-service/"
              name="SRW">
```

Listing 6.1 element definition

Element *types* koji je definisan WSDL specifikacijom koristi se za opis tipova ili struktura podataka koji se koriste prilikom razmene poruka. Tipovi podataka su uglavnom preuzeti iz XSD (XML Schema - Datatypes, 2004) ali mogu biti definisane i alternativne šeme poput (<http://www.w3.org/TR/wsdl20-altscemalangs>). Pošto je izabran *document/literal wrapped* patern za razmenu poruka svaka poruka predstavljena je odgovarajućim XML elementom koji je specifikiran XML šemom **Prilog 2**. XML šemom definisano je šest elemenata za svaki servis (operaciju) koji je predviđen SRU/W standardom po dva elementa, poruka koja predstavlja zahtev i poruka koja predstavlja odgovor. Prema tome definisani su sledeći elementi: *searchRetrieveRequest*, *searchRetrieveResponse*, *scanRequest*, *scanResponse*, *explainRequest*, *explainResponse*.

Element *message* definiše poruke koje se razmenjuju u komunikaciji sa veb servisom i na listingu 6.2 dat je deo WSDL dokumenta koji opisuje poruke koje se razmenjuju kada se pozivaju servisi *SearchRetrieve*, *Scan*, *Explain* definisani SRU/W standardom. Za svaki servis definisane su dve poruke (zahtev i odgovor) koje se razmenjuju. Atribut *name* jedinstveno identifikuje

poruke i to ime je u skladu sa imenom SRU/W servisa koji koristi samu poruku. Za svaku poruku koja je data na listingu postoji po jedan element *part* (logički deo), pri čemu svaki element *part* ukazuje na određeni tip koji je definisan elementom *types*. Logičkih delova u jednoj poruci može biti više a samo značenje *part* elementa se definiše u WSDL elementu *binding*. Element *part* vrlo često može biti i parametar metode veb servisa.

```

<message name="SearchRetrieveRequestMessage">
  <part name="body" element="srw:searchRetrieveRequest"/>
</message>
<message name="SearchRetrieveResponseMessage">
  <part name="body" element="srw:searchRetrieveResponse"/>
</message>
<message name="ScanRequestMessage">
  <part name="body" element="srw:scanRequest"/>
</message>
<message name="ScanResponseMessage">
  <part name="body" element="srw:scanResponse"/>
</message>
<message name="ExplainRequestMessage">
  <part name="body" element="srw:explainRequest"/>
</message>
<message name="ExplainResponseMessage">
  <part name="body" element="srw:explainResponse"/>
</message>

```

Listing 6.2 element definition

Element *binding* je mesto gde se apstraktna WSDL definicija pretvara u konkretnu. Naime, element *binding* služi za definisanje protokola koji će se koristiti prilikom razmene poruka, kao i za definisanje formata i enkodiranja poruke. Ovaj element nam daje informaciju koji se protokol koristi za transfer poruka operacija iz *portType*. Atributi elementa *binding* su *name* i *type*, gde je *name* jedinstveno ime mapiranja (*binding*), dok je vrednost atributa *type* ime koje se poklapa sa imenom definisanim u elementu *portType*. Tako je za operaciju *SearchRetrieveOperation*, čiji element *binding* je prikazan na listingu **6.3**, izabran je HTTP protokol za transport SOAP poruke i *document/literal* stil za ulaznu i izlaznu poruku (definisano elementom *soap:body*). Vrednost atributa *name* iz elementa *operation*, mora se poklopiti sa vrednošću atributa *name* elementa *operation* koji je podelement elementa *portType*. Kompletan listing elementa *binding* za SRU/W protokol dat u **Prilogu 3**.

```

<binding name="SRW-SoapBinding" type="SRWPort">
  <soap:binding style="document"
  transport="http://schemas.xmlsoap/soap/http"/>
  <operation name="SearchRetrieveOperation">
  <soap:operation soapAction="" style="document"/> <input>

```



```

<soap:body use="literal"/> </input> <output> <soap:body
use="literal"/> </output>
</operation>
...
</binding>

```

Listing 6.3 Element *binding*

Element *portType* sastoji se od skupa operacija, predstavljenih elementom *operation* pri čemu se za svaku operaciju definišu ulazni i izlazni parametri/formati poruka pomoću elementa *message*. Na listingu **6.4** dat je deo XML dokumenta koji opisuje operacije koje su podržane SRU/W standardom dok je kompletan dokument dat u **Prilogu 3**. U našem slučaju imamo tri operacije koje su sa servisima *SearchRetrieve*, *Scan*, *Explain* definisani po SRU/W standardu. Ime porta je definisano atributom *name* i ono mora biti jedinstveno.

```

<portType name="SRWPort">
<operation name="SearchRetrieveOperation"> <input
message="SearchRetrieveRequestMessage"/><output
message="SearchRetrieveResponseMessage"/> </operation> <operation
name="ScanOperation"> <input message="ScanRequestMessage"/> <output
message="ScanResponseMessage"/> </operation>
<operation name="ExplainOperation"> <input
message="ExplainRequestMessage"/> <output
message="ExplainResponseMessage"/> </operation>
<binding name="SRW-SoapBinding" type="SRWPort">

```

Listing 6.4 *portType* operation

Element *service* definiše fizičku lokaciju veb servisa i na listingu **6.5** dat je primer definicije veb servisa koji je instaliran na lokalnom računaru. Za svaki od podržanih protokola postoji jedan port element, gde servis predstavlja kolekciju portova. Za svaki port je upotrebljeno adekvatno mapiranje (*binding*) i adresa (<http://localhost:8080/SRU>) na kojoj je port dostupan. Korišćeni port je prethodno specificiran elementom *portType*.

```

<service name="SRWCIRSService">
<port name="SRW" binding="SRW-SoapBinding">
<soap:addresslocation="http://localhost:8080/SRU"/> </port> <port
name="ExplainSOAP" binding="Explain-SoapBinding">
<soap:addresslocation="http://localhost:8080/"> </port> </service>
</definitions>

```

Listing 6.5 *Service* location

6.2. Apache CXF

Za implemetaciju i realizaciju veb servisa po JAX-WS specifikaciji iz WSDL odabrana je Apache CXF biblioteka (Apache CXF: An Open-Source

Services Framework, 2014). Apache CXF je biblioteka otvorenog koda namenjena razvoju veb servisa, koja je u potpunosti u skladu sa JAX-WS specifikacijom i pruža sledeće mogućnosti:

- Potpuno odvaja "front-end" deo od osnovnog koda (zbog JAX-WS)
- Različite mogućnosti kreiranja klijenata, servisa (sa anotacijama ili iz WSDL)
- Visok nivo performansi i relativno laka implementacija servisa
- Podržava najrazličitije protokole: SOAP, XML, JSON, RESTful HTTP
- Uključena podrška za frejmovrke koji se koriste u razvoju veb aplikacija poput *Spring* i *Geronimo*.

Razvoj veb servisa je dodatno olakšan obzirom da u Apache CXF postoje set ugrađenih alata koji mogu vidno olakšati programsku implementaciju samih servisa. U nastavku su dati neki od najčešće korišćenih CXF alata:

- *Java to web service*. Ovaj alat omogućava da se formiraju POJO java klase sa specifičnim anotacijama koje će samom alatu omogućiti da od „obične“ java klase formira veb servis.
- *Java to WSDL*. Alat koji omogućava da se iz Java klase koja ima odlike servisa formira adekvatna WSDL specifikacija.
- *WSDL to Java*. Uz pomoć ovog alata se prosleđeni WSDL konvertuje u odgovarajuće java klase.
- *WSDL to JavaScript*. Alat koji od fajla po WSDL specifikaciji generiše Javascript kod budućeg veb servisa.
- *WSDL Validator*. Proverava validnost WSDL fajla u odnosu na postojeću XML šemu.
- *XSD to WSDL*. Formira WSDL fajl na osnovu prosleđenog XSD fajla.

Ova biblioteka podrazumeva da se nakon kreiranja WSDL dokumenta, proces razvoja JAX-WS servisa odvija u tri nezavisna koraka:

1. Generisanje inicijalnog programskog koda.

2. Implementacija operacija servisa.
3. Publikovanje implementiranog servisa.

Generisanje inicijalnog programskog koda.

U CXF biblioteci se konkretno `wSDL2java` alatom vrši automatsko generisanje koda iz WSDL fajla. Produkti ovog alata predstavljaju polaznu tačku za implementaciju servisa. Sam alat sadrži mnoštvo opcija za generisanje različitih klasa na osnovu sadržaja iz WSDL specifikacije. Ovaj alat podrazumeva pozivanje klase `org.apache.cxf.tools.wSDLto.WSDLToJava` i prosleđivanje fajla kao parametara klase. Samo generisanje klasa je za potrebe ove teze realizovano kao ant akcija:

```
<java failonerror="true"
classname="org.apache.cxf.tools.wSDLto.WSDLToJava" fork="yes">
<classpath>
<path refid="cxf.classpath" />
</classpath>
<arg value="srw.wsdl"/>
<arg value="-verbose"/>
<arg value="-validate"/>
<arg value="-client"/>
<arg value="-server"/>
<arg value="-impl"/>
</java>
```

Listing 6.6 Deo ant skripta za generisanje klasa

Iz *ant* skripta se vidi da su prilikom poziva klasi prosleđeni sledeći parametri:

- Prvi parametar je *srw.wsdl* što je naziv WSDL specifikacije na osnovu koje će se generisati java klase.
- *-verbose* parametar omogućuje prikaz deskriptivnih komentara tokom procesa generisanja.
- *-validate* parametar uključuje proces validacije WSDL fajla pre generisanja klasa
- *-client* parametar omogućuje generisanje okvira klijentskih klasa
- *-server* parametar omogućuje generisanje okvira serverskih klasa
- *-impl* omogućuje generisanje okvira za servisne klase

Pored ovih parametara postoji još niz koji se mogu koristiti prilikom generisanja klasa i definisani su na (<http://cxf.apache.org/docs/wsdl-to-java.html>).

JAX-WS specificira detaljno mapiranje servisa definisanog WSDL specifikacijom, na Java klase koje implementiraju taj servis. WSDL segment *definitions* sadrži listu korišćenih *namespace*, što za domen generisanja programskog koda znači da se klase koje nastaju iz WSDL smeštaju u folder (pakete) u skladu sa vrednošću *namespace* deklaracije. Npr. za *namespace xmlns:srw-interfaces=http://www.loc.gov/zing/srw/interfaces/* formiraće se hijerarhija foldera u skladu sa strukturom URI *namespace /gov/loc/zing/srw/interfaces/*.

Logički interfejs definisan elementom *wSDL:portType* je mapiran na tzv. "Service Endpoint Interface (SEI)" datoteku. SEI datoteka predstavlja java interface klasu koju servis treba da implementira. Ova datoteka samo po sebi treba ostati neizmenjena. Ime datoteke se poklapa sa imenom porta, što je u našem slučaju klasa pod nazivom *SRWPort*. Metode ovog interfejsa nastale su kao posledica mapiranja podelementa operation elemenata *portType*, gde se mapira ime operacije i parametri. Sadržaj ove klase naveden je na listingu **6.7**.

```
@WebService(targetNamespace =
"http://www.loc.gov/zing/srw/interfaces/", name = "SRWPort")
@XmlSeeAlso({gov.loc.zing.srw.ObjectFactory.class,gov.loc.zing.cql.x
cql.ObjectFactory.class,gov.loc.zing.srw.diagnostic.ObjectFactory.cl
ass})
@SOAPBinding(parameterStyle = SOAPBinding.ParameterStyle.BARE)
public interface SRWPort {
@WebResult(name = "searchRetrieveResponse", targetNamespace =
"http://www.loc.gov/zing/srw/", partName = "body")
@WebMethod(operationName = "SearchRetrieveOperation")
public gov.loc.zing.srw.SearchRetrieveResponseType
searchRetrieveOperation(
@WebParam(partName = "body", name = "searchRetrieveRequest",
targetNamespace = "http://www.loc.gov/zing/srw/")
gov.loc.zing.srw.SearchRetrieveRequestType body
);
}
```

Listing 6.7 Izgenerisana klasa od elementa

Za adekvatno funkcionisanje servisa pored java koda nalazimo i niz java anotacija:

- `@WebService` je anotacija koja klasu proglašava za veb servis po JAX-WS specifikaciji. Uz anotaciju su navedeni i atributi *name* i *targetNamespace*. Vrednost *name* atributa se poklapa sa vrednošću iz `<wsdl:portType>`. *targetNamespace* direktno ukazuje na prostor imena u kome je servis definisan.
- `@SOAPBinding` je definisana `javax.jws.soap.SOAPBinding` interfejsom. Anotacija se koristi kada želimo da specificiramo povezivanje (*binding*) SOAP porukama u veb servisu. Atribut *style* (`SOAPBinding.ParameterStyle.BARE`) podrazumeva način korišćenja SOAP poruka.
- `@WebMethod` anotacija je definisana u paketu `javax.jws.@WebMethod` i ukazuje na ime operacije definisane u WSDL (operation element).
- `@WebParam` koja bliže opisuje metodu veb servisa definišući njegove parametere. Atribut *partName* je ime definisano u podelementu `wsdl:part`. Ime parametra definisano je *name* atributom. Atribut *targetNamespace* definiše prostor imena u kome je parametar definisan.

Alat *WSDL2Java* generiše i okvir za implementaciju veb servisa tzv. skeleton implementacionu klasu, koja implementira prethodno opisani java interfejs Listing 6.8 Implementaciona klasa servisa. Ova klasa treba da sadrži konkretnu implementaciju SRU/W servisa pretrage naučno-istraživačkih podataka. Kao i kod interfejs klase i u implementacionoj klasi (Listing 6.9) stoji anotacija `@WebService` u čijim se atributima navodi: ime servisa (`SRWCRISService`) dobijeno iz `wsdl` elementa `service`, ime porta (`portName = "SRW"`), *targetNamespace* što je ime prostora imena u kome je definisan servis i *endpointInterface* koji ukazuje na klasu (putanja do adekvatnog paketa u projektu) koja je okvir (java interfejs) veb servisa.

```

@WebService(
    serviceName = "SRWCRISService", portName = "SRW",
    targetNamespace = "http://www.loc.gov/zing/srw/srw-sample-
service/",
    endpointInterface = "gov.loc.zing.srw.interfaces.SRWPort")
public class SRWPortImpl implements SRWPort {
...

```

Listing 6.8 Implementaciona klasa servisa

Iz elementa `wSDL:service` se generiše java klasa koju upotrebljavaju korisnici kako bi pristupili implementaciji javno objavljenog interfejsa servisa. U našem slučaju to je klasa `SRUWCRISClient` (Listing 6.9). Ovakav tip klase se prepoznaje po anotaciji `@WebServiceClient` na početku klase i u sebi sadrži samo lookup metode (za pronalaženje) za portove. Atributi pomenute anotacije ukazuju na sledeće:

- *Name* je vrednost dobijena iz atributa *name* WSDL elementa *service* i ukazuje na ime servisa
- *targetNamespace* je vrednost prostora imena u kome je definisan veb servis
- *wSDLLocation* je fizička putanja do WSDL specifikacije

```

@WebServiceClient(name = "SRWSampleService",
wSDLLocation =
"file:/home/valentin/Workspaces/workspaceSRU/VezebeWSDLWS/WEB-
INF/wSDL/srw-sample-service.wSDL",
targetNamespace = "http://www.loc.gov/zing/srw/srw-sample-service/")
public final static URL WSDL_LOCATION=
http://localhost:9999/bibliography/srw?wSDL";
    @WebEndpoint(name = "SRW")
    public SRWPort getSRW() {
        return super.getPort(SRW, SRWPort.class);
    }
}

```

Listing 6.9 CRIS UNS veb servis klijent

Tipovi podataka koje koristi WSDL su definisani u posebnoj datoteci *srw-types.xsd*. Svaki tip podataka definisan u WSDL se mapira na Java klase u skladu sa pravilima definisanim JAXB (*Java Architecture for XML Binding*) tehnologijom:

- XML prostorima imena odgovaraju Java paketi

- Složenim XML tipovima podataka odgovaraju Java klase
- XML atributima, referencama na elemente i deklaracijama lokalnih elemenata odgovaraju Java svojstva
- Ugrađenim *XML Schema* prostim tipovima podataka odgovaraju ugrađeni Java tipovi podataka

Tako je npr. povratna vrednost funkcije `searchRetrieveOperation` klasa `searchRetrieveResponseType` koja je kompleksni tip u *srw-types.xsd* (Listing 6.10).

```

<xsd:complexType name="searchRetrieveResponseType">
  <xsd:complexContent>
    <xsd:extension base="responseType">
      <xsd:sequence>
        <xsd:element ref="numberOfRecords"/>
        <xsd:element ref="resultSetId" minOccurs="0"/>
        <xsd:element ref="resultSetIdleTime" minOccurs="0"/>
        <xsd:element ref="records" minOccurs="0"/>
        <xsd:element ref="nextRecordPosition" minOccurs="0"/>
        <xsd:element ref="echoedSearchRetrieveRequest"
minOccurs="0"/>
        <xsd:element ref="diagnostics" minOccurs="0"/>
        <xsd:element ref="extraResponseData" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

Listing 6.10 Deo *srw-types.xsd*

Na osnovu prethodno navedenog listinga 6.10 primenom JAXB mapiranja nastaje java klasa sledećeg formata (Listing 6.11)

```

public class SearchRetrieveResponseType
  extends ResponseType
{
    @XmlElement(required = true)
    @XmlSchemaType(name = "nonNegativeInteger")
    protected BigInteger numberOfRecords;
}

```

```

protected String resultSetId;
@XmlSchemaType(name = "positiveInteger")
protected BigInteger resultSetIdleTime;
protected RecordsType records;
@XmlSchemaType(name = "positiveInteger")
protected BigInteger nextRecordPosition;
protected EchoedSearchRetrieveRequestType
echoedSearchRetrieveRequest;
protected DiagnosticsType diagnostics;
protected ExtraDataType extraResponseData;
...
public BigInteger getNumberOfRecords() {
    return numberOfRecords;
}
public void setNumberOfRecords(BigInteger value) {
    this.numberOfRecords = value;
}
...

```

Listing 6.11 JAXB klasa za element

U slučaju da je bio izabran metod razvijanja veb servisa odozdo ka gore, tada bi pisali java klase koje sadrže prethodno opisane anotacije iz kojih bi mogao da se generiše adekvatna WSDL specifikacija upotrebom *Apache CXF* alata *java2wsdl*.

Na osnovu alata WSDL2JAVA izgenerisana je i klijentska klasa za pristup veb servisu sledećeg forme:

```

public final class SRWPort_SRW_Client {

    private static final QName SERVICE_NAME = new
QName("http://www.loc.gov/zing/srw/srw-sample-service/", "
SRWCRISService ");
    SRWCRISService ss = new SRWCRISService (wsdlURL,
SERVICE_NAME);
    SRWPort port = ss.getSRW();
    _searchRetrieveOperation_body = new
SearchRetrieveRequestType ();
    _searchRetrieveOperation_body.setStartRecord(1);
    _searchRetrieveOperation_body.setMaximumRecords(5);
    gov.loc.zing.srw.SearchRetrieveResponseType
_searchRetrieveOperation_return =
port.searchRetrieveOperation(_searchRetrieveOperation_body);
...

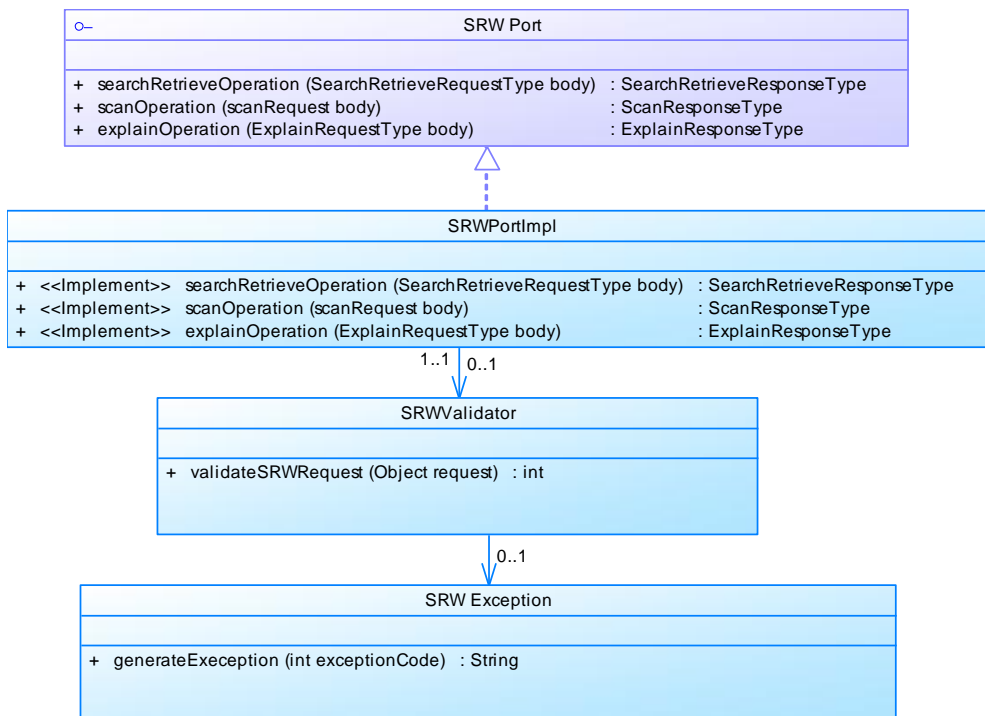
```

Listing 6.12 JAXB klasa za element

Dijagram klasa serverske strane protokola SRU/W dat je na slici **6.2**. Klasa SRW Port predstavlja običan Java *Interface* u kome se nalaze prototipovi funkcija veb servisa. Implementacija poslovne logike servisa nalazi se u klasi SRW Port Implementation, gde je za svaku operaciju SRU/W protokola (*searchRetrive*, *Scan*, *Explain*) implementirana odgovarajuća funkcija. Da li

su zahtevi klijenata u skladu sa SRU/W standardom proverava klasa SRWValidator. U zavisnosti od tipa zahteva i neslaganja sa SRU/W standardom, klasa SRU exception generiše odgovarajuću poruku.

Komunikacija između klijenta i veb servisa u CRIS UNS sistemu se naravno obavlja razmenom SOAP poruka. Jedan od najčešćih scenarija je da klijent šalje validan CQL upit ubačen kao parametar elementa *searchRetrieveRequest* kao što je to prikazano na listingu 6.12. CQL upitom se zahtevaju zapisi iz sistema koji u naslovu imaju reč “service”. U SOAP poruci zahteva se definiše i verzija korišćenog SRU/W protokola (<version>2.0</version>). Parametar <maximumRecords> predstavlja maksimalni broj zapisa koje klijent želi da dobije kao odgovor. Element *startRecord* definiše početnu tačku u skupu rezultata zapisa (npr. ako je vrednost ovog elementa 3, korisnik želi da dobije zapise počevši od trećeg iz skupa svih rezultata). Vrednost elementa *startRecord* mora biti manja ili jednaka sa vrednošću *maximumRecords*. Iz prethodnog proizilazi jasno ograničenje da je *startRecord* ≤ *maximumRecords*.



Slika 6.2 Dijagram klasa serverske strane SRU/W

Na listingu **6.13** je prikazana SOAP poruka koja odgovara na prethodno definisanom zahtevu. Kako je i očekivano čitav odgovor se nalazi u posebnom elementu `<searchRetrieveResponse>`. U okviru odgovora postavljena je verzija protokola (2.0) na serveru i ukupan broj zapisa (`<numberOfRecords>78</numberOfRecords>`) za zadati upit. Centralni deo odgovara predstavlja element koji se odnosi na konkretne zapise (`records`) u sistemu. Ujedno u okviru zapisa postoji poseban element (`record`) koji reprezentuje pojedinačni zapis. Svi zapisi su skladu sa CRIS profilom (Penca et al., 2014b) u elementu `<recordData>` se nalaze konkretni podaci o zapisu (*title, date, language*). Tako na primer, naslov rada se nalazi u elementu `<dc:title>` iz *Dublin Core Context Set* koji je deo CRIS profila. Za svaki zapis postoji i XML šema kao vrednost elementa `<recordSchema>info:srw/schema/1/dc-v1.1</recordSchema>`. Odgovor i zahtev se nalaze u SOAP *Envelope* elementu SOAP poruke (Listing **6.13** i **6.14**).

```

<?xml version="1.0" ?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
<searchRetrieveRequest xmlns="http://www.loc.gov/zing/srw/"
xmlns:ns2="http://www.loc.gov/zing/cql/xqql/"
xmlns:ns3="http://www.loc.gov/zing/srw/semantic/"
<query>dc.title=&quot;service&quot;</query>
<version>2.0</version>
<startRecord>1</startRecord>
<maximumRecords>5</maximumRecords>
<recordPacking>xml</recordPacking>
</searchRetrieveRequest>
</soapenv:Body>
</soapenv:Envelope>
</operation>

```

Listing 6.13 SRU/W SOAP request

```

<?xml version="1.0" ?>
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
<searchRetrieveResponse>
<version>2.0</version>
<numberOfRecords>78</numberOfRecords>
<records>
  <record>
    <recordSchema>info:srw/schema/1/dc-v1.1</recordSchema>
    <recordData>
<dc:title>Student Service Software System, version 2.0</dc:title>
<dc:creator>Rackovi&#263; Miloon;</dc:creator> Rackovic Milos
<dc:creator>&Scaron;krbi&#263; Sr&#273;an</dc:creator><br/> Škrbić
Srđan
<dc:creator>Pupovac Biljana</dc:creator><br/>Pupovac Biljana

```

```

<dc:creator>Bodroon;ki
&#381;arko</dc:creator><br/>&#160;&#160;&#160<dc:publisher>,      Novi
Sad, Serbia</dc:publisher>
<dc:date>2007</dc:date>                <dc:type>Text</dc:type>
<dc:identifier>http://www.cris.uns.ac.rs/record.jsf?recordId=6535</d
c:identifier>
<dc:language>English</dc:language>
  <recordData>
    </record>
  </record>
<recordSchema>http://srw.cris.uns.ac.rs/contextSets/CRIS/1.0</record
Schema>
  <recordPacking>xml</recordPacking>
  <recordData>
    <srw_dc:dc>
      <dc:title>CRIS service for journals and journal articles
evaluation</dc:title>
      <dc:date>2011</dc:date>
    </srw_dc>
    <srw_cris:cris>
<cris:type>JournalArticle</cris:type>
<cris:firstAuthor>Sinisa Nikolic </cris:firstAuthor>
<cris:abstract>This paper...</cris:abstract>
    ...
  </srw_cris:cris>
  </recordData>
</record>
</records>
</searchRetrieveResponse>
</soapenv:Body>
</soapenv:Envelope>
</operation>

```

Listing 6.14 SRU/W SOAP response

6.3. Korišćenje udaljenog pretraživanja

U ovom odeljku dati su mogući scenariji korišćenja udaljenog pretraživanja na bazi SRU/W standarda. Prepoznati su sledeći mogući scenariji korišćenja:

- Jasno je naznačeno da sistem za pretragu naučno-istraživačkih rezultata predstavlja samo deo (podsistem) CRIS UNS sistema. Prva verzija CRIS UNS razvijena još 2010 i omogućuje unos naučno-istraživačkih podataka u skladu sa CERIF standardom. Kako je sistem za unos podataka već neko vreme u fazi eksploatacije postoji više entiteta (fakulteta) sa Novosadskom univerziteta koji koriste sistem. Ideja je da se budućim korisnicima omogući **sloboda prilikom implementacije** svog **lokalnog sistema** za pretragu podataka iz CRIS UNS. Naime sa slike 4.5 se vidi da su, klijenti nezavisni od samog jezgra sistema pretrage. Tako da onaj ko želi da

koristi sistem pretrage treba da samo bude **usklađen** sa **SRU/W** standardom, tj. da može da pošalje **SOAP poruku** u kojoj će smestiti **CQL upit**. Pri tom, sistemu pretrage uopšte nije važno na koji način klijent realizuje svoj deo prikaza mogućnosti (opcija) za pretragu. Tako da klijent čak uopšte **nije ograničen** da prilikom **implementacije** interfejsa koristi **Java tehnologije** koje su osnove jezgra sistema. Sloboda izbora klijentske aplikacije za pretragu može da poveća i motivaciju za korišćenje pretrage fakultetima, jer nisu ograničeni na korišćenje postojeće aplikacije (poglavlje 4), već mogu dizajnirati sopstvenu u skladu sa svojim potrebama.

- Pored sistema za unos podataka u CRIS UNS se nalazi i digitalna biblioteka teza i disertacija (e-teze) (Ivanovic et al., 2012). E-teze su opisane skupom metapodataka koji obuhvata sve metapodatke propisane Dublin Core i EDT-MS formatom metapodataka. Digitalna biblioteka e-teza Univerziteta u Novom Sadu poseduje i mogućnost preuzimanja i izvoza metapodataka putem OAI-PMH protokola. Sa stanovišta sistema pretrage naučno-istraživačkih podataka iz digitalne biblioteke e-teze bi se mogle dobiti dodatne informacije kojih nema u CERIF formatu, ali ima u specifičnom formatu EDT-MS za opis teza i disertacije (npr mentor, članovi komisije, predsednik komisije). Ovim bi sistem za pretragu naučno-istraživačkih rezultata imao potpunije i sveobuhvatnije podatke o tezama i disertacijama, što bi automatski i pretragu učinilo kvalitetnijom. Benefiti digitalne biblioteke e-teza od povezivanja sa sistemom pretrage naučno-istraživačkih podataka su sledeći:
 - pristup i povezivanje sa drugim tipovima podataka iz CRIS UNS sistema, (oni koji nisu teze i disertacije) u cilju pravljenja izveštaja ili statistika. (npr. da li je neka disertacija proizašla kao rezultat nekog projekta, produkta ili patenta)
 - upotreba SRU/W protokola povećala bi dostupnost ka drugim sistemima. Time digitalna biblioteka e-teza dobija još jedan potencijalni "kanal" za razmenu podataka sa drugim sistemima. Sa aspekta implementacije digitalna biblioteka treba samo da implementira deo CRIS profila koji se odnosi na podatke o tezama i disertacijama.

- Još jedan značajan scenario korišćenja SRU/W standarda bi mogao da bude i formiranje **veb portala za pretragu mreže CRIS sistema**. CRIS sistemi ne propisuju nikakva pravila, tehnologije ili sl. za implementaciju istih, već koriste samo CERIF standard za opis podataka. Već je rečeno i da CRIS sistemi ne pružaju nikakvu standardizaciju sa stanovišta pretrage. Stoga bi za kreiranje **veb portala za pretragu mreže CRIS sistema** bilo neophodno implementirati CRIS profil pretrage u svim CRIS sistemima nad kojima želimo da se izvrši pretraga. CRIS profil ni na koji način ne uslovljava CRIS sisteme da u potpunosti implementiraju sve stavke iz navedenog profila. Svako treba implementirati samo deo profila (indeksa) da bi ostao u skladu sa podacima iz sopstvenog sistema, bez potrebe za izmenom arhitekture kao i strukture podataka samih sistema. Tako bi sa jednog centralnog mesta bili dostupni podaci iz niza najrazličitijih CRIS sistema.
- Korišćenjem standarda SRU/W mogao bi se konstruisati pretraživi **centralni repozitorijum normativnih podataka** (časopisa, istraživača, konferencija...). Poznato je da se navedeni podaci ne nalaze **samo** u **CRIS** sistemima, već i u različitim **bibliotečkim** sistema, u kojima je naravno SRU/W de fakto standard. SRU/W standard bi predstavljao vezu između CRIS sistema i bibliotečkih sistema, jer oni imaju potpuno drugačije standarde (MARC 21, UNIMARC ...) za predstavu podataka od CRIS sistema (CERIF). Upotrebom SRU/W standarda bi izbegli prikupljanje podataka na jednom mestu i njihovu konverziju iz jednog formata u drugi. **Centralni repozitorijum normativnih podataka** bi mogao da bude interesantan državnim institucijama da na osnovu normativnih podataka generišu najrazličitije vrste izveštaja i statistike.

7. Zaključak

Predmet istraživanja je sistem za pretragu naučno-istraživačkih rezultata. Sistem za pretragu naučno-istraživačkih rezultata je nastao kao proširenje postojećeg sistema za unos podataka o tim rezultatima (CRIS UNS). Analizom postojećih sistema za naučno-istraživačkih sistema došlo se do zaključka da je korišćeno vrlo malo opšte prihvaćenih standarda za predstavljanje i pretragu podataka iz naučno-istraživačkog domena. Standardizacija u ovakvim sistemima je od ključnog značaja jer treba da obezbedi nezavisnost pretraživanja od samih modela za reprezentaciju podataka naučno-istraživačke delatnosti. Stoga je i rezultat istraživanja sistem pretrage naučno-istraživačkih podataka izgrađen upotrebom postojećih standarda (Z39.50, SRU/W ...) i tehnologija (Web 2.0, AJAX, Lucene ...). Testiranje i verifikacija sistema urađena je na naučno-istraživačkim podacima CRIS UNS sistema.

U **prvom**, odnosno uvodnom poglavlju dat je pregled postojeće literature u vezi sa pretragom naučno-istraživačkih podataka. Poseban akcenat je na podržanim funkcionalnostima i korišćenim standardima u domenu pretrage naučnih rezultata. Takođe, u ovom poglavlju bile su identifikovane i tehnologije koje se koriste u sistemima za pretragu (Web 2.0, AJAX, Lucene ...). Na kraju poglavlja su dati ciljevi disertacije, kao i postavljene hipoteze za izradu disertacije.

Pregled postojećih sistema koji omogućuju nekakav vid pretrage naučno-istraživačkih podataka dat je u **drugom** poglavlju. Pored opisa osnovnih karakteristika samih sistema sprovedena je i analiza istih u odnosu na: dostupne indekse, modove pretrage, tipove upita. Analizirane su četiri grupe sistema: Institucionalni repozitorijumi (IR), baze naučnih sadržaja, CRIS sistemi i portali pretrage.

Treće poglavlje sadrži detaljan opis CRIS profila pretrage. Pomenuti profil je u skladu sa SRU/W standardom. CRIS profil je nastao analizom postojećih SRU/W profila i na osnovu zaključaka iz analize sistema pretraga naučno-istraživačkih podataka. Ovo poglavlje sadrži i detaljnu specifikaciju SRU/W standard i upitni jezik CQL.

Četvrto poglavlje sadrži opis podataka iz postojećeg CRIS UNS sistema. Opisani su standardi CERIF i MARC21 na bazi kojih su izvršeno

skladištenje naučno-istraživačkih podataka. Takođe, opisana je i komponenta za preliminarnu obradu, indeksiranje i pretragu tekstualnih sadržaja. Ova komponenta se bazira na softverskoj biblioteci *Apache Lucene*. Iz *Apache Lucene* su opisani delovi API koji se odnose na analizu podataka i formiranje pretraživih indeksa. Na kraju poglavlja je prikazano je mapiranje uskladištenih podataka CRIS UNS sistema na adekvatne indekse pretrage.

U **petom** poglavlju je opisana specifikacija, arhitektura i implementacija sistema za pretraživanje naučno-istraživačkih podataka. Prvo je data specifikacija informacionih zahteva sistema upotrebom dijagrama slučajeva korišćenja. Nakon toga je opisana arhitektura sistema upotrebom dijagrama razmeštaja. U ovom poglavlju detaljno je specificirana implementacija sistema za pretraživanje naučno-istraživačkih podataka unutar postojećeg informacionog sistema naučno-istraživačke delatnosti CRIS UNS. Na kraju ovog poglavlja opisani mogući scenariji korišćenja sistema za pretraživanje naučno-istraživačkih podataka. Na osnovu pregleda literature proizašli su neophodni zaključci za stvaranje sistema pretrage koji je pre svega intuitivan za korišćenje i usklađen sa postojećim standardima koji se intenzivno koriste u sistemima pretrage podataka iz naučno-istraživačke delatnosti. Sistem predstavlja veb aplikaciju koja je u potpunosti implementirana u skladu sa Web 2.0 specifikacijom, gde je poseban akcenat stavljen na JSF i AJAX tehnologiju. Primena AJAX tehnologija i realizacija naprednih komponenti interfejsa omogućena je upotrebom *RichFaces* frejmvorka. Standardizacija pretrage ostvarena je upotrebom SRU/W standarda. Sama implementacija nastavlja praksu postojećeg sistema za unos podatka koristeći samo tehnologije otvorenog koda. Sistem pretrage je modularan i dozvoljava da se pojedine komponente koriste i realizuju na više različitih načina, usled čega postoje mogućnosti za:

- razvijanje eksterne aplikacije za pretragu, koja bi se bazirala na SRU/W bibliotečkom standardu.
- jednostavnu izmenu Tekst servera, koja ne bi uticala na komponentu *Interface*.
- realizaciju uporedne pretrage više Tekst servera čija implementacija i fizička lokacija može biti proizvoljna. U tom slučaju, potrebno je samo definisati odgovarajuća mapiranja CQL jezika na jezike korišćenih Tekst servera

Šesto poglavlje sadrži detaljan opis specifikacija i implementacija serverske strane SRU/W protokola u sistemu za pretragu naučno-istraživačkih rezultata. Na kraju je izneta i diskusija o mogućim scenarijima korišćenja udaljenog pretraživanja.

Pomenuti CRIS profil koji je detaljno opisan u trećem poglavlju rešava jedan od najčešćih problema u CRIS sistemima i glavni je doprinos ove doktorske disertacije. Naime, prioritet CRIS sistema je da obezbede samo standarde za predstavu i skladištenje samih podataka, dok su standardi pretrage i pristupa podataka zanemareni. Obzirom da ne postoje jasni standardi i preporuke u domenu pretrage podataka naučno-istraživačke delatnosti dolazi se do situacije gde nastaje mnoštvo različitih implementacija. Sam profil predstavlja predlog kako da se izvrši standardizacija pretrage podataka iz naučno-istraživačke oblasti. Posebna prednost CRIS profila je nezavisnost istog od implementacije samog sistema pretrage. Na primer, profil ostavlja otvoren prostor za odabir programskog jezika implementacije pretrage i dozvoljava integraciju već realizovanih komponenti (tekst servera). Pri implementaciji profila treba uzeti obzir da je neophodno koristiti nove informacione tehnologije kao što su WSDL, SOAP, HTTP i XML. Korišćenje navedenih tehnologija je postala svakodnevica i predstavlja nezaobilaznu stavku u aplikacijama koje žele da obezbede dostupnost velikom broju korisnika. Uvođenje predloženog profila ni u kom slučaju ne treba da navede programere na rekonstrukciju postojećih ili konstrukciju potpuno novih sistema. Naime, primena predloženog profila je vrlo jednostavna jer se njegovi elementi trebaju samo asocirati sa podacima u postojećim sistemima. Jednostavnost primene profila praktično je pokazana u CRIS sistemu Univerziteta u Novom Sadu. Sistem za pretragu poseduje servis pretrage podataka naučno-istraživačke delatnosti baziranog na SRU/W standardu tako da u startu pruža i potencijalnu interoperabilnost sa različitim bibliotečkim sistemima, jer SRU/W de fakto standard u ovim sistemima

Rezultati istraživanja u ovoj disertaciji su potvrdili navedene polazne hipoteze. Razvijen je unifikovani profil pretrage za CRIS sisteme i implementiran je standardizovani sistem pretrage naučno-istraživačkih podataka koji se koristi na Univerzitetu u Novom Sadu. Specificiran i implementiran je sistem za pretragu naučno-istraživački rezultata baziran na Web 2.0 tehnologijama i upotrebi CQL upitnog jezika. Specificiran i

implementiran je servis koji omogućuje pretragu sa udaljenih računara upotrebom SRU/W protokola.

Trenutno sistem podržava nezavisne pretrage publikacija i autora, kao i poseban mod za zadavanje tekstualnog CQL upita. U budućnosti se pored poboljšanja interfejsa, planira i dodavanje novih modova pretrage kojima bi se u potpunosti podržao CERIF model, kao i neprekidan rad na poboljšanju prikaza rezultata pretrage. Stoga bi se prikaz rezultata pored Harvard stila referenciranja mogao biti i u *APA*, *Vancouver* ili *IEEE* obliku. Dalji razvoj sistema biće usklađen sa trendovima i inovacijama u SUI i standarda pretrage. U potpunosti je jasno da predloženi profil oslikava trenutno stanje u domenu pretrage naučno-istraživačkih podataka, i da bi se u budućnosti elementi samog profila mogli revidirati u skladu sa stanjem u pomenutom domenu. U budućnosti je planirano da veb servis za pretragu naučno-istraživačkih podataka omogući i pretragu upotrebom *RESTful* standarda.

Prilog 1- Analizirani sistemi pretraga

1. **DSpace@Cambridge** – – институционални репозиторијум Cambridge Универзитета заснован на DSpace систему. Доступан на: <<http://www.dspace.cam.ac.uk/>>
2. **MARS (Mason Archival Repository Service)** – дигитална библиотека George Mason Универзитета заснован на DSpace систему. Доступан на:< <http://digilib.gmu.edu:8080/>>
3. **DSpace@MIT**– институционални репозиторијум Massachusetts Института за Технологију заснован на DSpace систему. Доступан на: <<http://dspace.mit.edu/>>
4. **ERA (Edinburgh Research Archive)** – - институционални репозиторијум Edinburgh Универзитета заснован на DSpace систему. Доступан на: <<http://www.era.lib.ed.ac.uk/>>
5. **ROSE (Repository of Scholarly eprints)** институционални репозиторијум Bristol Универзитета заснован на DSpace систему. Доступан на: <<http://rose.bris.ac.uk/>>
6. **WSU (Washington State University Research Repository)** - - институционални репозиторијум Washington State Универзитета заснован на DSpace систему. Доступан на: <<https://research.wsulibs.wsu.edu:8443/jspui/>>
7. **BELSU (Belgorod State University Research Repository)** - институционални репозиторијум Belgorod State Универзитета заснован на DSpace систему. Доступан на: <<http://dspace.bsu.edu.ru/>>
8. **UnisaIR (University of South Africa Research Repository)** - институционални репозиторијум South Africa Универзитета заснован на DSpace систему. Доступан на: <<http://uir.unisa.ac.za/>>
9. **UT Repository (University of Tokyo Research Repository)** - институционални репозиторијум Токуо Универзитета заснован на DSpace систему. Доступан на: <<http://repository.dl.itc.u-tokyo.ac.jp/dspace/>>

10. **The Kingston University Research Repository** - - институционални репозиторијум Kingston Универзитета заснован на EPrints систему. Доступан на: <<http://eprints.kingston.ac.uk/>>
11. **WRRO (White Rose Research Online)** - обједињени институционални репозиторијум Leeds, Sheffield и York Универзитета Универзитета заснован на EPrints систему. Доступан на: <<http://eprints.whiterose.ac.uk/>>
12. **University of Southampton Research Repository** - институционални репозиторијум Southampton Универзитета заснован на EPrints систему. Доступан на: <<http://eprints.soton.ac.uk/>>
13. **PaDUA@Research** - институционални репозиторијум Padova Универзитета заснован на EPrints систему. Доступан на: <<http://paduaresearch.cab.unipd.it/>>
14. **UTAS Eprints** институционални репозиторијум Tasmania Универзитета заснован на EPrints систему. Доступан на: <<http://eprints.utas.edu.au/>>
15. **Nottingham ePrints** - институционални репозиторијум Nottingham Универзитета заснован на EPrints систему. Доступан на: <<http://eprints.nottingham.ac.uk/>>
16. **IEEE Computer Society Digital library** – база научног садржаја IEEE Computer Society удружења. Доступана на: <<http://www.computer.org/portal/web/search/advanced/>>
17. **SpringerLink** - – база научног садржаја Springer удружења. Доступана на: <<http://www.springerlink.com/>>
18. **Emerald** - база научног садржаја Emerald Group Publishing Limited удружења. Доступана на: <<http://www.emeraldinsight.com/>>
19. **SciVerse Scopus** - база научног садржаја Elsevier удружења које садржи апстрактну и цитатну база података рецензиране научне литературе и други веб садржај. Доступан на: <<http://www.scopus.com/home.url>>
20. **NARCIS (National Academic Research and Collaborations Information System)** - национални портал Холандије за претрагу

- академско истраживачких података. Доступан на:
<<http://www.narcis.nl/>>
21. **USDA CRIS** - CRIS систем развијен за потребе Департмана за Пољопривреду Сједињених Америчких Држава. Доступан на:
<<http://cris.nifa.usda.gov/>>
 22. **SICRIS (Slovenian Current Research Information System)** - CRIS систем државе Словеније. Доступан на:
<<http://sicris.izum.si/>>
 23. **E-CRIS SR (Current Research Information System in Serbia)** - CRIS систем државе Србије. Доступан на: <<http://e-cris.sr.cobiss.net/>>
 24. **Cristin (Current Research Information System In Norway)** - CRIS систем државе Норвешке. Доступан на:
<<https://www.cristin.no/>>
 25. **HunCRIS (Hungarian Current Research Information System)** - CRIS систем државе Мађарске. Доступан на:
<http://nkr.info.omikk.bme.hu/HunCRIS_eng.htm>
 26. **FRIS (Flanders Research Information Space)** - Flamanski portal istraživački podataka. Dostupan na:
<<http://www.researchportal.be/en>>
 27. **OpenAIRE (Open Access Infrastructure Research for Europe)** - Otvoireni Evropski portal. Available at: <<https://www.openaire.eu/>>
 28. **GTR (Gateway to research)** - portal Istraživačkog saveta Velike Britanije. Dostupan na: <<http://gtr.rcuk.ac.uk/>>
 29. **EuroRISNet+ Observatory** - Evropska mreža nacionalnih Kontaktnih tačaka. Dostupan na: <<http://observatory.euroris-net.eu>>
 30. **Royal Holloway** - CRIS sistem Univerziteta u Londonu baziran na komercijalnoj PURE platformi. Dostupan na:
<<https://pure.rhul.ac.uk/portal/>>
 31. **Aalborg University** - CRIS sistem Univerziteta u Aalborgu baziran na komercijalnoj PURE platformi. Dostupan na
<<http://vbn.aau.dk/en/>>

32. **HoGent** - CRIS sistem Univerziteta/Koleža iz Genta baziran na komercijalnoj PURE platformi. Dostupan na: <<http://pure.hogent.be/portal/en/service/search.html/>>
33. **Mid Sweden** - CRIS sistem Mid Sweden Univerziteta baziran na komercijalnoj Converis platformi. Dostupan na: <<http://miun.avedas.com/converis/>>
34. **WWU Münster** - CRIS sistem Münster Univerziteta baziran na komercijalnoj Converis platformi. Dostupan na: <<https://www.uni-muenster.de/forschungaz/search>>
35. **University of HULL** - CRIS sistem Hull Univerziteta baziran na komercijalnoj Converis platformi. Dostupan na: <<http://hull.avedas.com/converis/search>>
36. **DRIVER (Digital Repository Infrastructure Vision for European Research)** - Pan-evropski za pretragu publikacija. Dostupan na: <<http://www.driver-repository.eu>>
37. **GoogleScholar** – Globalni portal za pretragu naučnih publikacija u vlasništvu kompanije Google. Dostupan na: <<http://scholar.google.com/>>
38. **Microsoft Academic Search** - Globalni portal za pretragu naučno-istraživačkih podataka u vlasništvu kompanije Microsoft. Dostupan na: <<http://academic.research.microsoft.com/>>

Prilog 2 - SRU/W tipovi podata

U prilogu 2 je data kompletna XML šema sa korišćenim tipovima podataka.

```
<xsd:schema targetNamespace="http://www.loc.gov/zing/srw/"
xmlns:diag="http://www.loc.gov/zing/srw/diagnostic/"
xmlns:xcql="http://www.loc.gov/zing/cql/xcql/"
xmlns="http://www.loc.gov/zing/srw/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xsd:import namespace="http://www.loc.gov/zing/cql/xcql/"
schemaLocation="xcql.xsd"/>
  <xsd:import namespace="http://www.loc.gov/zing/srw/diagnostic/"
schemaLocation="diagnostics.xsd"/>
  <xsd:complexType name="requestType">
    <xsd:sequence>
      <xsd:element ref="version"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="responseType">
    <xsd:sequence>
      <xsd:element ref="version"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="searchRetrieveRequest"
type="searchRetrieveRequestType" nillable="false"/>
  <xsd:complexType name="searchRetrieveRequestType">
    <xsd:complexContent>
      <xsd:extension base="requestType">
        <xsd:sequence>
          <xsd:element ref="query"/>
          <xsd:element ref="startRecord" minOccurs="0"/>
          <xsd:element ref="maximumRecords" minOccurs="0"/>
          <xsd:element ref="recordPacking" minOccurs="0"/>
          <xsd:element ref="recordSchema" minOccurs="0"/>
          <xsd:element ref="recordXPath" minOccurs="0"/>
          <xsd:element ref="resultSetTTL" minOccurs="0"/>
          <xsd:element ref="sortKeys" minOccurs="0"/>
          <xsd:element ref="stylesheet" minOccurs="0"/>
          <xsd:element ref="extraRequestData" minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:element name="searchRetrieveResponse"
type="searchRetrieveResponseType" nillable="false"/>
  <xsd:complexType name="searchRetrieveResponseType">
    <xsd:complexContent>
      <xsd:extension base="responseType">
        <xsd:sequence>
          <xsd:element ref="numberOfRecords"/>
          <xsd:element ref="resultSetId" minOccurs="0"/>
          <xsd:element ref="resultSetIdleTime" minOccurs="0"/>
          <xsd:element ref="records" minOccurs="0"/>
          <xsd:element ref="nextRecordPosition" minOccurs="0"/>
          <xsd:element ref="echoedSearchRetrieveRequest"
minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

```

        <xsd:element ref="diagnostics" minOccurs="0"/>
        <xsd:element ref="extraResponseData" minOccurs="0"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:element name="scanRequest" type="scanRequestType"
nillable="false"/>
<xsd:complexType name="scanRequestType">
    <xsd:complexContent>
        <xsd:extension base="requestType">
            <xsd:sequence>
                <xsd:element ref="scanClause"/>
                <xsd:element ref="responsePosition" minOccurs="0"/>
                <xsd:element ref="maximumTerms" minOccurs="0"/>
                <xsd:element ref="stylesheet" minOccurs="0"/>
                <xsd:element ref="extraRequestData" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:element name="scanResponse" type="scanResponseType"
nillable="false"/>
<xsd:complexType name="scanResponseType">
    <xsd:complexContent>
        <xsd:extension base="responseType">
            <xsd:sequence>
                <xsd:element ref="terms" minOccurs="0"/>
                <xsd:element ref="echoedScanRequest" minOccurs="0"/>
                <xsd:element ref="diagnostics" minOccurs="0"/>
                <xsd:element ref="extraResponseData" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:element name="explainRequest" type="explainRequestType"
nillable="false"/>
<xsd:complexType name="explainRequestType">
    <xsd:complexContent>
        <xsd:extension base="requestType">
            <xsd:sequence>
                <xsd:element ref="recordPacking" minOccurs="0"/>
                <xsd:element ref="stylesheet" minOccurs="0"/>
                <xsd:element ref="extraRequestData" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
<xsd:element name="explainResponse" type="explainResponseType"
nillable="false"/>
<xsd:complexType name="explainResponseType">
    <xsd:complexContent>
        <xsd:extension base="responseType">
            <xsd:sequence>
                <xsd:element ref="record"/>
                <xsd:element ref="echoedExplainRequest" minOccurs="0"/>
                <xsd:element ref="diagnostics" minOccurs="0"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>

```

```

        <xsd:element ref="extraResponseData" minOccurs="0"/>
    </xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:element name="version" type="xsd:string" nillable="false"/>
<xsd:element name="query" type="xsd:string" nillable="false"/>
<xsd:element name="startRecord" type="xsd:positiveInteger"
default="1" nillable="false"/>
<xsd:element name="maximumRecords" type="xsd:nonNegativeInteger"
nillable="false"/>
<xsd:element name="recordPacking" type="xsd:string"
nillable="false"/>
<xsd:element name="recordSchema" type="xsd:string"
nillable="false"/>
<xsd:element name="recordXPath" type="xsd:string"
nillable="false"/>
<xsd:element name="resultSetTTL" type="xsd:nonNegativeInteger"
nillable="false"/>
<xsd:element name="stylesheet" type="xsd:anyURI"
nillable="false"/>
<xsd:element name="sortKeys" type="xsd:string" nillable="false"/>
<xsd:element name="extraRequestData" type="extraDataType"
nillable="false"/>
<xsd:element name="numberOfRecords" type="xsd:nonNegativeInteger"
nillable="false"/>
<xsd:element name="resultSetId" type="xsd:string"
nillable="false"/>
<xsd:element name="resultSetIdleTime" type="xsd:positiveInteger"
nillable="false"/>
<xsd:element name="records" type="recordsType" nillable="false"/>
<xsd:complexType name="recordsType">
    <xsd:sequence>
        <xsd:element ref="record" maxOccurs="unbounded"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="record" type="recordType" nillable="false"/>
<xsd:complexType name="recordType">
    <xsd:sequence>
        <xsd:element ref="recordSchema"/>
        <xsd:element ref="recordPacking"/>
        <xsd:element ref="recordData"/>
        <xsd:element ref="recordPosition" minOccurs="0"/>
        <xsd:element ref="extraRecordData" minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
<xsd:element name="recordData" type="stringOrXmlFragment"
nillable="false"/>
<xsd:element name="recordPosition" type="xsd:positiveInteger"
nillable="false"/>
<xsd:element name="extraRecordData" type="extraDataType"
nillable="false"/>
<xsd:element name="nextRecordPosition" type="xsd:positiveInteger"
nillable="false"/>
<xsd:element name="diagnostics" type="diagnosticsType"
nillable="false"/>
<xsd:complexType name="diagnosticsType">

```



```

    <xsd:sequence>
      <xsd:element ref="diag:diagnostic" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="extraResponseData" type="extraDataType"
nillable="false"/>
  <xsd:element name="echoedSearchRetrieveRequest"
type="echoedSearchRetrieveRequestType" nillable="false"/>
  <xsd:complexType name="echoedSearchRetrieveRequestType">
    <xsd:complexContent>
      <xsd:extension base="requestType">
        <xsd:sequence>
          <xsd:element ref="query"/>
          <xsd:element name="xQuery" type="xcql:operandType"
nillable="false"/>
          <xsd:element ref="startRecord" minOccurs="0"/>
          <xsd:element ref="maximumRecords" minOccurs="0"/>
          <xsd:element ref="recordPacking" minOccurs="0"/>
          <xsd:element ref="recordSchema" minOccurs="0"/>
          <xsd:element ref="recordXPath" minOccurs="0"/>
          <xsd:element ref="resultSetTTL" minOccurs="0"/>
          <xsd:element ref="sortKeys" minOccurs="0"/>
          <xsd:element name="xSortKeys" type="xSortKeysType"
nillable="false" minOccurs="0"/>
          <xsd:element ref="stylesheet" minOccurs="0"/>
          <xsd:element ref="extraRequestData" minOccurs="0"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <xsd:complexType name="xSortKeysType">
    <xsd:sequence>
      <xsd:element ref="sortKey" maxOccurs="unbounded"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="sortKey" type="sortKeyType" nillable="false"/>
  <xsd:complexType name="sortKeyType">
    <xsd:sequence>
      <xsd:element ref="path"/>
      <xsd:element ref="schema" minOccurs="0"/>
      <xsd:element ref="ascending" minOccurs="0"/>
      <xsd:element ref="caseSensitive" minOccurs="0"/>
      <xsd:element ref="missingValue" minOccurs="0"/>
    </xsd:sequence>
  </xsd:complexType>
  <xsd:element name="path" type="xsd:string" nillable="false"/>
  <xsd:element name="schema" type="xsd:string" nillable="false"/>
  <xsd:element name="ascending" type="xsd:boolean"
nillable="false"/>
  <xsd:element name="caseSensitive" type="xsd:boolean"
nillable="false"/>
  <xsd:element name="missingValue" type="xsd:string"
nillable="false"/>
  <xsd:element name="scanClause" type="xsd:string"
nillable="false"/>
  <xsd:element name="responsePosition" type="xsd:nonNegativeInteger"
default="1" nillable="false"/>

```

```

<xsd:element name="maximumTerms" type="xsd:positiveInteger"
nillable="false"/>
<xsd:element name="terms" type="termsType" nillable="false"/>
<xsd:complexType name="termsType">
  <xsd:sequence>
    <xsd:element ref="term" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="term" type="termType" nillable="false"/>
<xsd:complexType name="termType">
  <xsd:sequence>
    <xsd:element ref="value"/>
    <xsd:element ref="numberOfRecords" minOccurs="0"/>
    <xsd:element ref="displayTerm" minOccurs="0"/>
    <xsd:element name="whereInList" minOccurs="0">
      <xsd:simpleType>
        <xsd:restriction base="xsd:string">
          <xsd:enumeration value="first"/>
          <xsd:enumeration value="last"/>
          <xsd:enumeration value="only"/>
          <xsd:enumeration value="inner"/>
        </xsd:restriction>
      </xsd:simpleType>
    </xsd:element>
    <xsd:element ref="extraTermData" minOccurs="0"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="value" type="xsd:string" nillable="false"/>
<xsd:element name="displayTerm" type="xsd:string"
nillable="false"/>
<xsd:element name="extraTermData" type="extraDataType"
nillable="false"/>
<xsd:element name="echoedScanRequest" type="echoedScanRequestType"
nillable="false"/>
<xsd:complexType name="echoedScanRequestType">
  <xsd:complexContent>
    <xsd:extension base="requestType">
      <xsd:sequence>
        <xsd:element ref="scanClause"/>
        <xsd:element name="xScanClause"
type="xcql:searchClauseType"/>
        <xsd:element ref="responsePosition" minOccurs="0"/>
        <xsd:element ref="maximumTerms" minOccurs="0"/>
        <xsd:element ref="stylesheet" minOccurs="0"/>
        <xsd:element ref="extraRequestData" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<xsd:element name="echoedExplainRequest"
type="explainRequestType"/>
<xsd:complexType name="extraDataType">
  <xsd:sequence>
    <xsd:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>

```

```
<xsd:complexType name="xmlFragment" mixed="false">
  <xsd:sequence>
    <xsd:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="stringOrXmlFragment" mixed="true">
  <xsd:sequence>
    <xsd:any namespace="##any" processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
</xsd:complexType>
</xsd:schema>
```

Prilog 3 - WSDL specifikacija SRU/W servisa

U prilogu 3 data je kompletna WSDL specifikacija SRU/W servisa za pretragu naučno-istraživačkih podataka.

```
<definitions xmlns:srw="http://www.loc.gov/zing/srw/" xmlns:srw-
interfaces="http://www.loc.gov/zing/srw/interfaces/"
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:xcql="http://www.loc.gov/zing/cql/xcql/"
xmlns:diag="http://www.loc.gov/zing/srw/ diagnostic/"
xmlns:xsort="http://www.loc.gov/zing/srw/xsortkeys/"
targetNamespace="http://www.loc.gov/zing/srw/interfaces/"
name="SRW">

  <types>
    <xsd:schema>
      <xsd:import namespace="http://www.loc.gov/zing/srw/"
schemaLocation="srw-types.xsd"/>
    </xsd:schema>
  </types>
  <message name="SearchRetrieveRequestMessage">
    <part name="body" element="srw:searchRetrieveRequest"/>
  </message>
  <message name="SearchRetrieveResponseMessage">
    <part name="body" element="srw:searchRetrieveResponse"/>
  </message>
  <message name="ScanRequestMessage">
    <part name="body" element="srw:scanRequest"/>
  </message>
  <message name="ScanResponseMessage">
    <part name="body" element="srw:scanResponse"/>
  </message>
  <message name="ExplainRequestMessage">
    <part name="body" element="srw:explainRequest"/>
  </message>
  <message name="ExplainResponseMessage">
    <part name="body" element="srw:explainResponse"/>
  </message>
  <portType name="SRWPort">
    <operation name="SearchRetrieveOperation">
      <input message="srw-
interfaces:SearchRetrieveRequestMessage"/>
      <output message="srw-
interfaces:SearchRetrieveResponseMessage"/>
    </operation>
    <operation name="ScanOperation">
      <input message="srw-interfaces:ScanRequestMessage"/>
      <output message="srw-interfaces:ScanResponseMessage"/>
    </operation>
  </portType>
```

```

    <portType name="ExplainPort">
      <operation name="ExplainOperation">
        <input message="srw-interfaces:ExplainRequestMessage"/>
        <output message="srw-
interfaces:ExplainResponseMessage"/>
      </operation>
    </portType>
    <soap:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="SearchRetrieveOperation">
      <soap:operation soapAction="" style="document"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="ScanOperation">
      <soap:operation soapAction="" style="document"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="SRWSampleService">
    <port name="SRW" binding="srw-bindings:SRW-SoapBinding">
      <soap:address
location="http://insertserver/inserturl"/>
    </port>
  </service>

```

Literatura

Altova XMLSpy XML Editor [WWW Document], 2014. URL <http://www.altova.com/xmlspy.html> (pristupljeno 12.09.2014).

Anderson, K., 2001. Publishing Online-Only Peer-Reviewed Biomedical Literature: Three Years of Citation, Author Perception, and Usage Experience. *J. Electron. Publ.* 6. doi:10.3998/3336451.0006.303

Antelman, K., 2004. Do Open-Access Articles Have a Greater Research Impact? *Coll. Res. Libr.* 65, 372–382.

Anuradha, K.T., Sivakaminathan, R., Kumar, P.A., 2011. Open-source tools for enhancing full-text searching of OPACs: Use of Koha, Greenstone and Fedora. *Program Electron. Libr. Inf. Syst.* 45, 231–239.

Apache CXF: An Open-Source Services Framework [WWW Document], 2014. URL <http://xf.apache.org/> (pristupljeno 12.09.2014).

Apache Lucene [WWW Document], 2005. URL <http://lucene.apache.org/> (pristupljeno 12.09.2014).

Apache Tomcat [WWW Document], 2014. URL <http://tomcat.apache.org/> (pristupljeno 12.09.2014).

Asserson, A., Jeffery, K., Lopatenko, A., 2002. CERIF: past, present and future: an overview, in: *Proceedings of the 6th International Conference on Current Research Information Systems*, University of Kassel. Kassel, pp. 33–40.

Asynchronous JavaScript and XML [WWW Document], 2007. URL <http://www.dmoz.org/Computers/Programming/Languages/JavaScript/AJAX> (pristupljeno 12.09.2014).

Augmented Backus-Naur Form [WWW Document], 2008. URL <http://tools.ietf.org/pdf/rfc5234.pdf> (pristupljeno 12.09.2014).

Baeza-Yates, R., 1999. *Modern information retrieval*. ACM Press ; Addison-Wesley, c1999, New York ; Harlow, England.

Bailo, D., Jeffery, K.G., 2014. EPOS: A Novel Use of CERIF for Data-intensive Science. *Procedia Comput. Sci.* 33, 3–10. doi:10.1016/j.procs.2014.06.002

- Baranov, V., Plemnek, A., Riabev, V., Sokolova, N., Sova, D., Usmanov, R., 2000. Review of Z39.50 servers and Z39.50 environment in Russia. *Libr. Hi Tech* 18, 304–315. doi:10.1108/07378830010360419
- Bates, M.J., 1979. Information search tactics. *J. Am. Soc. Inf. Sci.* 30, 205–214.
- Bender, M., Sebastian, M., Gerhard, W., Christian, Z., 2005. Challenges of distributed search across digital libraries. *Future Digit. Libr. Manag. Syst. Syst. Archit. Inf. Access.*
- Biswas, G., Paul, D., 2010. An evaluative study on the open source digital library softwares for institutional repository: Special reference to Dspace and greenstone digital library. *Int. J. Libr. Inf. Sci.* 2, 001–010.
- Boberić Krstićev, D., 2013. Information Retrieval Using a Middleware Approach. *Inf. Technol. Libr.* 32. doi:10.6017/ital.v32i1.1941
- Brophy, J., Bawden, D., 2005. Is Google enough? Comparison of an internet search engine with academic library resources. *Aslib Proc.* 57, 498–512.
- Callan, J.P., Lu, Z., Croft, W.B., 1995. Searching distributed collections with inference networks. *ACM Press*, pp. 21–28. doi:10.1145/215206.215328
- Carnell, J., Linwood, J., Zawadzki, M., 2003. *Jakarta Web applications: problem, design, solution.* Wrox, Birmingham.
- Cavaleri, P., 2008. The use of AJAX in searching a bibliographic database: A case study of the Italian Biblioteche oggi database. *Program Electron. Libr. Inf. Syst.* 42, 275–285.
- CERIF 1.5 Semantics [WWW Document], 2010. URL http://www.eurocris.org/Uploads/Web%20pages/CERIF-1.5/CERIF1.5_Semantics.xhtml (pristupljeno 12.09.2014).
- CLDR - Unicode Common Locale Data Repository [WWW Document], 2014. URL <http://cldr.unicode.org/> (pristupljeno 12.09.2014).
- Common European Research Information Format | CERIF [WWW Document], 2000. URL

<http://www.eurocris.org/Index.php?page=CERIFintroduction&t=1>
(pristupljeno 12.09.2014).

CQL Context Set, version 1.2 [WWW Document], 2009. URL <http://www.loc.gov/standards/sruBob/resources/cql-context-set-v1-2.html> (pristupljeno 12.09.2014).

CQL: the Contextual Query Language: Specifications [WWW Document], 2008. URL <http://www.loc.gov/standards/sruBob/specs/cql.html> (pristupljeno 12.09.2014).

Crudge, S.E., Johnson, F.C., 2004. Using the information seeker to elicit construct models for search engine evaluation. *J. Am. Soc. Inf. Sci. Technol.* 55, 794–806.

Current Research Information System [WWW Document], 2000. . euroCRIS. URL <http://www.eurocris.org> (pristupljeno 12.09.2014)

Data Transfer Object [WWW Document], 2003. URL <http://msdn.microsoft.com/en-us/en-%20us/library/ms978717.aspx> (pristupljeno 12.09.2014).

DC-Library Application Profile (DC-LAP) [WWW Document], 2007. URL <http://dublincore.org/groups/libraries/library-application-profile.shtml> (pristupljeno 12.09.2014).

Diagnostics List (SRU: Search/Retrieval via URL -- SRU) [WWW Document], 2012. URL <http://www.loc.gov/standards/sru/diagnostics/diagnosticsList.html> (pristupljeno 12.09.2014).

Dimić, B., Segedinac, M., Ivanović, D., 2012. A BIBO ontology extension for evaluation of scientific research results, in: *Proceedings of the Fifth Balkan Conference in Informatics*. Presented at the Fifth Balkan Conference in Informatics, Novi Sad, pp. 16–20. doi:10.1145/2371316.2371376

Dimić-Srula, B., Ivanovic, D., 2012. Software component for reporting in the CRIS systems, in: *Proceedings of the CRIS 2012 Conference*,. Presented at the CRIS 2012 Conference, Prague, pp. 61–66.

Dublin Core® Metadata Initiative (DCMI) Format [WWW Document], 2014. URL <http://dublincore.org/> (pristupljeno 12.09.2014).

Dublin Core Context Set Version 1.1 [WWW Document], 2007. URL <http://www.loc.gov/standards/sru/cql/contextSets/dc-context-set.html> (pristupljeno 12.09.2014).

Dublin Core Extended XML [WWW Document], 2008. URL <http://www.loc.gov/standards/sruBob/resources/dcx/dcx.xsd> (pristupljeno 12.09.2014).

Dvořák, J., Jörg, B., 2013. CERIF 1.5 XML - Data Exchange Format Specification. p. 16.

Eysenbach, G., 2006. Citation Advantage of Open Access Articles. *PLoS Biol.* 4, e157. doi:10.1371/journal.pbio.0040157

Gartner, R., Cox, M., Jeffery, K., 2013. A CERIF-based schema for recording research impact. *Electron. Libr.* 31, 465–482.

Gavrillis, D., Kakali, C., Papatheodorou, C., 2008. Enhancing library services with Web 2.0 functionalities, in: *Proceedings of the 12th European Conference on Research and Advanced Technology for Digital Libraries*. Springer-Verlag, Berlin, Heidelberg, pp. 148–159.

Google Query Language [WWW Document], 2010. URL <https://developers.google.com/> (pristupljeno 12.09.2014).

Google Scholar and academic libraries [WWW Document], 2012. URL http://en.wikipedia.org/wiki/Google_Scholar_and_academic_libraries (pristupljeno 12.09.2014).

Gospodnetić, O., 2005. *Lucene in action*. Manning, Greenwich, CT.

Guskov, A., Zhizhimov, O., Kikhtenko, V., Skachkov, D., Kosyakov, D., 2014. RuCRIS: A Pilot CERIF based System to Aggregate Heterogeneous Data of Russian Research Projects. *Procedia Comput. Sci.* 33, 163–167. doi:10.1016/j.procs.2014.06.027

Hafezi, M.A., 2008. Interoperability between library software: a solution for Iranian libraries. *Electron. Libr.* 26, 726–734.

Harnard, S., Brody, T., 2004. Comparing the impact of open access (OA) vs. non-OA articles in the same journals. *-Lib Mag.* 10.

Hoppe, M., Razum, M., 2008. A SRW/U-compliant Search Service for Fedora. Presented at the Third International Conference on Open Repositories, United Kingdom, Southampton.

ISO 3166 - Country codes - ISO [WWW Document], 2012. URL http://www.iso.org/iso/country_codes.htm (pristupljeno 12.09.2014).

Ivanovic, D., 2010. Informacioni sistem naučno-istraživačke delatnosti. Fakultet tehničkih nauka.

Ivanovic, D., 2011a. Data exchange between CRIS UNS, institutional repositories and library information systems, in: Proceedings of the 5th International Quality Conference, Kragujevac,. Kragujevac, pp. 371–378.

Ivanovic, D., 2011b. Sistemi za skladištenje naučnih sadržaja. Zadužbina Andrejević.

Ivanović, D., Milosavljevic, B., 2009. Software architecture of system of bibliographic data, in: Proceedings of the XXI Conference on Applied Mathematics PRIM 2009,. Presented at the PRIM, pp. 85–94.

Ivanovic, D., Milosavljevic, B., Konjovic, Z., Surla, D., 2009. Funkcionalnost korisničkog interfejsa za bibliotečku obradu. Presented at the SNTPI, Beograd, pp. 61–64.

Ivanovic, D., Milosavljevic, G., Milosavljevic, B., Surla, D., 2010. A CERIF-compatible research management system based on the MARC 21 format. Program Electron. Libr. Inf. Syst. 44, 229–251.

Ivanovic, D., Surla, D., Konjovic, Z., 2011. CERIF compatible data model based on MARC 21 format. Electron. Libr. 29, 52–70.

Ivanović, D., Surla, D., Racković, M., 2010. A CERIF data model extension for evaluation and quantitative expression of scientific research results. Scientometrics 86, 155–172.

Ivanovic, D., Surla, D., Rackovic, M., 2012. Journal evaluation based on bibliometric indicators and the CERIF data model. Comput. Sci. Inf. Syst. 9, 791–811.

Ivanović, L., Dimić Surla, B., Segedinac, M., Ivanović, D. 2012. CRISUNS ontology for theses and dissertations', in: Proceedings of the ICIST 2012. Presented at the ICIST 2012, Kopaonik, pp. 164–169.

- Ivanović, L., Surla, D., 2012. A software module for import of theses and dissertations to CRISs, in: Proceedings of the CRIS 2012 Conference. Presented at the CRIS 2012 Conference, Prague, pp. 313–322.
- Ivanović, L., 2013. Search of catalogues of theses and dissertations. *J. Math. Novi Sad* 43, 155–165.
- Ivanovic, L., Ivanovic, D., Surla, D., 2012. A data model of theses and dissertations compatible with CERIF, Dublin Core and EDT-MS. *Online Inf. Rev.* 36, 548–567. doi:10.1108/14684521211254068
- Ivanović, L., Ivanović, D., Surla, D., 2012. Integration of a Research Management System and an OAI-PMH Compatible ETDs Repository at the University of Novi Sad. *Libr. Resources Tech. Serv.* 56, 104–112.
- Ivanovic, L., Ivanovic, D., Surla, D., Konjovic, Z., 2013. User interface of web application for searching PhD dissertations of the University of Novi Sad. *IEEE*, pp. 117–122. doi:10.1109/SISY.2013.6662553
- JAXB Reference Implementation [WWW Document], 2005. URL <https://jaxb.java.net/> (pristupljeno 12.09.2014).
- Jeffery, K., 2010. The CERIF Model As the Core of a Research Organization. *Data Sci. J.* 9, CRIS7–CRIS13. doi:10.2481/dsj.CRIS2
- Jeffery, K., Asserson, A., 2010. CERIF-CRIS for the European e-Infrastructure. *Data Sci. J.* 9, CRIS1–CRIS6. doi:10.2481/dsj.CRIS1
- Jeffery, K.G., 2007. *CRIS Architectures For Interoperation*. Viena.
- Jeffery, K., Houssos, N., Jörg, B., Asserson, A., 2014. Research information management: the CERIF approach. *Int. J. Metadata Semant. Ontol.* 9, 5. doi:10.1504/IJMSO.2014.059142
- Jörg, B., 2008. CERIF: Common European Research Information Format - Insight into the CERIF 2008 Release. IZUM, Maribor, Slovenia, pp. 183–192.
- Jörg, B., Jeffery, K., Dvorak, J., Houssos, N., Asserson, A., van Grootel, G., Gartner, R., Cox, M., Rasmussen, H., Vestdam, T., Strijbosch, L., Brasse, V., Zendulkova, D., Höllrigl, T., Valkovic, L., Engfer, A., Jägerhorn, M., Mahey, M., Brennan, N., Sicilia, M.A., Ruiz-Rube, I.,

Baker, D., Evans, K., Price, A., Zielinski, M., 2012. CERIF 1.3 Full Data Model (FDM) Introduction and Specification.

Kalin, M., 2009. Java web services: up and running, 1st ed. ed. O'Reilly, Beijing ; Sebastopol, Calif.

Konchady, M., 2008. Building search applications: Lucene, LingPipe, and Gate, 1st ed. ed. Mustru Pub, Oakton, VA.

Kovacevic, A., Ivanovic, D., Milosavljevic, B., Konjovic, Z., Surla, D., 2011. Automatic extraction of metadata from scientific publications for CRIS systems. *Program Electron. Libr. Inf. Syst.* 45, 376–396. doi:10.1108/00330331111182094

Kurtz, M.J., Eichhorn, G., Accomazzi, A., Grant, C., Demleitner, M., Murray, S.S., 2005a. Worldwide use and impact of the NASA Astrophysics Data System digital library. *J. Am. Soc. Inf. Sci. Technol.* 56, 36–45. doi:10.1002/asi.20095

Kurtz, M.J., Eichhorn, G., Accomazzi, A., Grant, C., Demleitner, M., Murray, S.S., Martimbeau, N., Elwell, B., 2005b. The bibliometric properties of article readership information. *J. Am. Soc. Inf. Sci. Technol.* 56, 111–128. doi:10.1002/asi.20096

Kurtz, M.J., Eichhorn, G., Accomazzi, A., Grant, C.S., Demleitner, M., Murray, S.S., 2005. The Effect of Use and Access on Citation. *Inf. Process. Manag.* 41, 1395–1402.

Lagoze, C., Fielding, D., 1998. Defining Collections in Distributed Digital Libraries. -*Lib Mag.*

Lawrence, S., 2001. Free online availability substantially increases a paper's impact. *Nature* 411, 521. doi:10.1038/35079151

Levene, M., Poulouvassilis, A., 2004. Web dynamics: adapting to change in content, size, topology and use. Springer, Berlin; New York.

Library of Congress Authorities File [WWW Document], 2012. URL <http://authorities.loc.gov/> (pristupljeno 12.09.2014).

Library of Congress [WWW Document], 1999. URL <http://www.loc.gov/> (pristupljeno 12.09.2014).

- Lossau, N., 2004. Search Engine Technology and Digital Libraries: Libraries Need to Discover the Academic Internet. -Lib Mag. 10, 1.
- Luhn, H.P., 1957. A statistical approach to mechanized encoding and searching of literary information. IBM J. Res. Dev. 1, 309–317.
- Madhusudhan, M., Aggarwal, S., 2011. Web-based online public access catalogues of IIT libraries in India: an evaluative study. Program Electron. Libr. Inf. Syst. 45, 415–438.
- Manning, C.D., Raghavan, P., Schütze, H., 2008. Introduction to information retrieval. Cambridge University Press, New York.
- Mansourian, Y., 2008. Coping strategies in web searching. Program Electron. Libr. Inf. Syst. 42, 28–39.
- MARC 21 XML Schema [WWW Document], 2014. URL <http://www.loc.gov/standards/marcxml/> (pristupljeno 12.09.2014).
- McCallum, S., 2006. A Look at New Information Retrieval Protocols: SRU, OpenSearch/A9, CQL, and XQuery. Presented at the 10² IF LA-CDNL Alliance for Bibliographic Standards IC ABS, Seoul.
- McCandless, M., 2010. Lucene in action, 2nd ed. ed. Manning, Greenwich.
- Mehrad, J., Gilvari, A., 2006. Search Abilities in Electronic Journals: A Comparative Study of Search Capabilities of Persian and Non-persian E-Journals. Iran. J. Inf. Sci. Technol. 4, 57–66.
- Metadata Object Description Schema: MODS (Library of Congress) [WWW Document], 2013. URL <http://www.loc.gov/standards/mods/> (pristupljeno 12.09.2014).
- Mi, J., Weng, C., 2008. Revitalizing the Library OPAC: interface, searching, and display challenges. Inf. Technol. Libr. 27, 5–22.
- Milosavljevic, B., Boberic, D., Surla, D., 2010. Retrieval of bibliographic records using Apache Lucene. Electron. Libr. 28, 525–539.
- Milosavljević, B., Tešendić, D., 2010. Software architecture of distributed client/server library circulation system. Electron. Libr. 28, 286 – 299.

- Milosavljevic, G., Ivanovic, D., Surla, D., Milosavljevic, B., 2011. Automated construction of the user interface for a CERIF-compliant research management system. *Electron. Libr.* 29, 565–588.
- Moczar, L., 2014. Lucene and Solr.
- Mojarra JavaServer Faces [WWW Document], 2014. URL <https://javaserverfaces.java.net/> (pristupljeno 12.09.2014).
- Morgan, E.L., 2004. An Introduction to the Search/Retrieve URL Service (SRU). Ariadne.
- Morville, P., 2010. Search patterns, 1st ed. ed. O'Reilly Media, Sebastopol, CA.
- MySQL [WWW Document], 2014. URL <http://www.mysql.com/> (pristupljeno 12.09.2014).
- National Information Standards Organization (U.S.), A.N.S.I., 2003. International Standard Z39.50 Maintenance Agency. NISO Press, Bethesda, Md.
- Nicholas Eric Craswell, 2000. Methods for Distributed Information Retrieval. The Australian National University,.
- Nicollet, D., 2009. AJAX Incremental Search Revisited How a Search Engine for Structured Data enables a Better Autocomplete Interface.
- Nikolić, S., Penca, V., Ivanović, D., 2013. STORING OF BIBLIOMETRIC INDICATORS IN CERIF DATA MODEL. Kopaonik mountain resort, Republic of Serbia.
- Nikolić, S., Penca, V., Ivanović, D., 2014. System for modelling rulebooks for the evaluation of scientific-research results. Case study: Serbian Rulebook. Presented at the 4. International Conference on Information Science and Technology (ICIST), Kopaonik.
- Nikolić, S., Penca, V., Ivanović, D., Surla, D., Konjović, Z., 2012. CRIS service for journals and journal articles evaluation, in: Proceedings of the 11th International Conference on Current Research Information Systems. Prague, Czech Republic, pp. 323–332.

OASIS | Advancing open standards for the information society [WWW Document], n.d. URL <https://www.oasis-open.org/> (pristupljeno 12.09.2014).

Penca, V., Ivanović, D., Surla, D., Konjović, Z., 2012. Development of a unified search profile for crs systems, in: 6th International Conference on Methodologies, Technologies, and Tools Enabling E-Government (MeTTeG12). Belgrade, Serbia, pp. 56–65.

Penca, V., Nikolić, S., 2012. Scheme for mapping Published Research Results from Dspace to Cerif Format, in: 2. International Conference on Information Society Technology and Management,. pp. 170–175.

Penca, V., Nikolić, S., Ivanović, D., 2014a. SRU/W service for CRIS UNS system, in: 4. International Conference on Information Science and Technology (ICIST). Society for Information Systems and Computer Networks, Kopaonik, pp. 108–114

Penca, V., Nikolić, S., Ivanović, D., Konjović, Z., Surla, D., 2014b. SRU/W Based CRIS Systems Search Profile. Program Electron. Libr. Inf. Syst.

Pinto, C.S., Simões, C., Amaral, L., 2014. CERIF – Is the Standard Helping to Improve CRIS? *Procedia Comput. Sci.* 33, 80–85. doi:10.1016/j.procs.2014.06.013

Record Metadata Context Set [WWW Document], 2007. URL <http://srw.cheshire3.org/contextSets/rec/1> (pristupljeno 12.09.2014).

Registered SRU/W Profiles [WWW Document], 2009. URL <http://www.loc.gov/standards/sru/profiles/> (pristupljeno 12.09.2014).

Registred Context Sets [WWW Document], 2011. URL <http://www.loc.gov/standards/sruBob/resources/context-sets.html> (pristupljeno 12.09.2014).

RFC 3986 (STD 66) (Uniform Resource Identifier [WWW Document], 2005. URL <http://www.ietf.org/rfc/rfc3986.txt> (pristupljeno 12.09.2014).

RFC 4287 - The Atom Syndication Format [WWW Document], 2005. URL <http://tools.ietf.org/html/rfc4287> (pristupljeno 12.09.2014).

Richardson, W.C. (Ed.), 2004. Professional portal development with open source tools: Java™ Portlet API, Lucene, James, Slide (Wrox Press), Programmer to programmer. Wiley Pub, Indianapolis, Ind.

RichFaces - JBoss [WWW Document], 2014. URL <http://richfaces.jboss.org/> (pristupljeno 12.09.2014).

Sander-Beuermann, W., Nebel, M., Adamczak, W., 2008. Searching the CRISses. IZUM, Maribor, Slovenia.

Saracevic, T., 1997. The stratified model of information retrieval interaction: Extension and applications. *Proc. Am. Soc. Inf. Sci.* 34, 313–27.

Sastry, H.G., Reddy, L.C., 2010. Significance of Web 2.0 in Digital Libraries. *Int. J. Comput. Sci. Eng. IJCSE* 2, 2208–2211.

Schatz, B.R., 1997. Information Retrieval in Digital Libraries: Bringing Search to the Net. *Science* 275, 327–334. doi:10.1126/science.275.5298.327

Shiri, A., Revie, C., 2005. Usability and user perceptions of a thesaurus-enhanced search interface. *J. Doc.* 61, 640–656.

Smith, A.G., 2012. Internet search tactics. *Online Inf. Rev.* 36, 7–20.

Spink, A., 2003. Web search: emerging patterns. *Libr. Trends* 52, 299–306.

SRU Record Schemas [WWW Document], 2013. URL <http://www.loc.gov/standards/sruBob/resources/schemas.html> (pristupljeno 12.09.2014).

SRU- Search/Retrieve via URL [WWW Document], 2004. URL <http://www.loc.gov/standards/sru/> (pristupljeno 12.09.2014).

Stanford E-Journal User Survey [WWW Document], 2004. URL <http://ejust.stanford.edu/usersurvey2-linked.htm> (pristupljeno 12.09.2014)

Surla, D., Ivanovic, D., Konjovic, Z., 2013. Development of the software system CRIS UNS, in: *Proceedings of the 11th International Symposium on Intelligent Systems and Informatics (SISY)*. IEEE, Subotica, pp. 111–116.

Surla, D., Ivanović, D., Konjović, Z., Racković, M., 2012. Rules for Evaluation of Scientific Results Published in Scientific Journals. *Manag. Inf. Syst.* 7, 003–010.

The Harvard system of referencing [WWW Document], 2004. URL <http://libweb.anglia.ac.uk/referencing/harvard.htm> (pristupljeno 12.09.2014)

Tunkelang, D., 2009. Faceted Search. *Synth. Lect. Inf. Concepts Retr. Serv.* 1, 1–80. doi:10.2200/S00190ED1V01Y200904ICR005

Unified Modeling Language (UML) [WWW Document], 1997. URL <http://www.uml.org/> (pristupljeno 12.09.2014).

UNIMARC XML šema [WWW Document], 2004. URL <http://www.bncf.firenze.sbn.it/progetti/unimarc/slim/documentation/unimarcslim.html> (pristupljeno 12.09.2014).

Wellhausen, T., 2006. *User Interface Design for Searching A Pattern Language*.

What Is Web 2.0 - O'Reilly Media [WWW Document], 2005. URL <http://oreilly.com/web2/archive/what-is-web-20.html> (pristupljeno 12.09.2014).

White, M.S., 2013. *Enterprise search*. O'Reilly, Beijing □ ; Farnham.

Willett, P., 2006. The Porter stemming algorithm: then and now. *Program Electron. Libr. Inf. Syst.* 40, 219–223. doi:10.1108/00330330610681295

Wilson, M.L., 2012. *Search User Interface Design, SYNTHESIS LECTURES ON INFORMATION CONCEPTS, RETRIEVAL, AND SERVICES*. Morgan & Claypool.

Wusteman, J., O'Hiceadha, P., 2006. Using Ajax to empower dynamic searching. *Inf. Technol. Libr.* 25, 57–64.

XCQL XML šema [WWW Document], 2004. URL <http://www.loc.gov/standards/sru/xmlFiles/xcql.xsd> (pristupljeno 12.09.2014).

XML Schema - Datatypes [WWW Document], 2004. URL <http://www.w3.org/TR/xmlschema-2/> (pristupljeno 12.09.2014).

Z39.50 Explain, Explained and Re-Engineered in XML [WWW Document], n.d. URL <http://explain.z3950.org/dtd/commentary.html> (pristupljeno 12.09.2014).

Zaric, M., Krsticev, D.B., Surla, D., 2012. Multitarget/multiprotocol client application for search and retrieval of bibliographic records. *Electron. Libr.* 30, 351–366.

Biografija

Valentin Penca rođen je 23.04.1983 u Vršcu gde je završio osnovnu školu i gimnaziju. Studije na Fakultetu tehničkih nauka u Novom Sadu upisao je školske 2002/2003. godine. Tokom studija je bio nosilac priznanja univerziteta za izuzetan uspeh u školskim godinama 2004/2005, 2005/2006, 2006/2007, a tih godina je bio ujedno i stipendista Ministarstva prosvete i sporta Republike Srbije. Završio je integrisane osnovne akademske – bečelor studije i diplomske akademske – master studije na smeru Računarske nauke i informatika 2008. godine. Diplomski master rad na temu „Detekcija upada na računarsku mrežu bazirana na sistemima pravila“ odbranio je 1. jula 2008. godine ocenom 10.

Doktorske studije na Fakultetu tehničkih nauka upisao je školske 2009/2010 godine. Položio je sve ispite predviđene planom i programom studijskog programa Računarstvo i automatika sa prosečnom ocenom 10.

Na fakultetu tehničkih nauka od 2010 do 2011 radi u svojstvu saradnika u nastavi. Od 2011 radi kao asistent za užu naučnu oblast Primenjene računarske nauke i informatika na istom fakultetu.

Autor je 18 publikovanih naučnih radova. Učestvovao je u izradi više stručnih i naučnih projekata.

Živi u Novom Sadu. Od stranih jezika govori engleski jezik.

UNIVERZITET U NOVOM SADU

FAKULTET TEHNIČKIH NAUKA

Ključna dokumentacijska informacija

<i>Redni broj:</i> RBR	
<i>Identifikacioni broj:</i> IBR	
<i>Tip dokumentacije:</i> TD	Monografska dokumentacija
<i>Tip zapisa:</i> TZ	Tekstualni štampani materijal
<i>Vrsta rada:</i> VR	Doktorska disertacija
<i>Autor:</i> AU	Valentin Penca
<i>Mentor:</i> MN	dr Dragan Ivanović, docent, Fakultet Tehničkih Nauka, Novi Sad
<i>Naslov rada:</i> NR	Modeliranje i implementacija sistema za pretraga naučno-istraživačkih podataka
<i>Jezik publikacije:</i> JP	srpski (latinica)
<i>Jezik izvoda:</i> JI	srpski (latinica) / engleski
<i>Zemlja</i>	Republika Srbija

<i>publikovanja:</i> ZP	
<i>Uže geografsko područje:</i> UGP	Vojvodina
<i>Godina:</i> GO	2014.
<i>Izdavač:</i> IZ	Autorski reprint
<i>Mesto i adresa:</i> MA	Fakultet Tehničkih Nauka, Trg Dositeja Obradovića 6, Novi Sad, Republika Srbija
<i>Fizički opis rada:</i> FO	(7/199/144/13/36/0/3) (broj poglavlja/strana/lit.citata/ tabela/slika/grafika/priloga)
<i>Naučna oblast:</i> NO	Informatika
<i>Naučna disciplina:</i> ND	Informacioni sistemi
<i>Predmetna odrednica/ ključne reči:</i> PO UDK	SRU/W, CQL,CERIF, MARC 21, UML, sistem za pretragu naučno-istraživačkih rezultata
<i>Čuva se:</i> ČU	Biblioteka Fakulteta tehničkih nauka, Trg Dositeja Obradovića 6, Novi Sad
<i>Važna napomena:</i> VN	Nema

Izvod:

IZ

Cilj – Cilj je specifikacija, modeliranje i implementacija sistema pretrage naučno-istraživačkih podataka. Sistem pretrage treba da omogući bogatu i efikasnu pretragu naučno-istraživačkih podataka. Osnovna ideja je da se pretraga zasniva na bibliotečkim standardima SRU/W i CQL kako bi se mogli prihvatati zahtevi za pretragu sa udaljenih računara, a da korisnici sistema mogu da pretražuju putem veb forme implementirane upotrebom Web 2.0 tehnologije pri čemu se od korisnika ne očekuje poznavanje pomenutih standarda

Metodologija – Na osnovu pregledane literature došlo se do smernica za analizu sistema koji sadrže naučno-istraživačke podatke. Na osnovu izvršene analize predložene su neophodne funkcionalnosti i relevantni set indeksa koji pretraga CRIS sistema treba da podrži. Predložen je novi profil za pretragu podataka naučno-istraživačke delatnosti. Implementiran je standardizovani sistem pretrage za CRIS sisteme baziran na Web 2.0 tehnologijama, SRU/W profilu pretrage i CQL upitnom jeziku. Sistem omogućava i udaljeno pretraživanje eksternim entitetima zadavanjem adekvatnih zahteva u skladu sa SRU/W standardom. Korišćeni su CASE alati bazirani na objektno-orijentisanoj metodologiji za modeliranje sistema (UML 2.0). Softverska arhitektura je zasnovana na Internet tehnologijama. Za implementaciju je korišćena Java platforma. Dok je za izradu naprednih elemenata interfejsa korišćena se JSF biblioteka i AJAX tehnologije.

Rezultati – Razvijen je unifikovani profil pretrage za CRIS sisteme i implementiran je standardizovani sistem pretrage naučno-istraživačkih podataka koji se koristi na Univerzitetu u Novom Sadu. Specificiran i implementiran je sistem za pretragu naučno-istraživačkih rezultata baziran na Web 2.0 tehnologijama i upotrebi CQL upitnog jezika. Korisnički sistem je dovoljno intuitivan da ga mogu koristiti i korisnici koji i ne poznaju standarde na kojima se zasniva sam pretraga. Specificiran i

	<p>implementiran je servis koji omogućuje pretragu sa udaljenih računara upotrebom SRU/W protokola.</p> <p>Praktična primena – Sistem za pretragu naučno-istraživačkih rezultata je implementiran kao deo CRIS UNS sistema koji se koristi na Univerzitetu u Novom Sadu. Čime je obezbeđen javni uvid i jednostavna pretraga podataka o institucijama/organizacijama, istraživačima i publikovanim naučnim rezultatima unutar novosadskog univerziteta. Podržani SRU/W standard ga čini potencijalno dostupnim i za eksterne sisteme.</p> <p>Vrednost – Dat je profil pretrage koji predstavlja predlog kako da se izvrši standardizacija pretrage podataka iz naučno-istraživačke oblasti. Posebna prednost pomenutog CRIS profila je nezavisnost istog od implementacije samog sistema pretrage. Sama primena predloženog profila je vrlo jednostavna jer se njegovi elementi trebaju samo asocirati sa podacima u postojećim sistemima. Jednostavnost primene profila praktično je pokazana u CRIS sistemu Univerziteta u Novom Sadu. Sama aplikacija pretrage implementira SRU/W profil i pruža potencijalnu inteoperabilnost sa sistemima koji podržavaju SRU/W standard. Tako su nakon implementacije sistema, podaci iz CRIS UNS postali dostupniji i transparentniji.</p>
<p><i>Datum prihvatanja teme od NN veća:</i></p> <p>DP</p>	<p>31.03.2014.</p>
<p><i>Datum odbrane:</i></p> <p>DO</p>	
<p><i>Članovi komisije:</i></p> <p>KO</p>	

<i>Predsednik:</i>	dr Surla Dušan, profesor emeritus Prirodno-matematičkog fakulteta, Univerziteta u Novom Sadu
<i>član:</i>	dr Konjović Zora, redovni profesor Fakulteta tehničkih nauka, Univerziteta u Novom Sadu
<i>član:</i>	dr Miroslav Trajanović, redovni profesor Mašinskog fakulteta, Univerziteta u Nišu
<i>član:</i>	dr Milosavljević Branko, redovni profesor Fakulteta tehničkih nauka, Univerziteta u Novom Sadu
<i>član:</i>	dr Zarić Miroslav, docent Fakulteta tehničkih nauka, Univerziteta u Novom Sadu
<i>član:</i>	dr Dragan Ivanović, docent Fakulteta tehničkih nauka, Univerziteta u Novom Sadu, mentor

UNIVERSITY OF NOVI SAD

FACULTY OF SCIENCE

Key words documentation

<i>Accession number:</i> ANO	
<i>Identification number:</i> INO	
<i>Document type:</i> DT	Monograph publication
<i>Type of record:</i> TR	Printed text
<i>Content code:</i> CC	PhD Thesis
<i>Author:</i> AU	Valentin Penca
<i>Mentor/comentor:</i> MN	Dragan Ivanović, PhD, full professor, Faculty of Technical Sciences, Novi Sad
<i>Title:</i> TI	Modeling and implementation of search system for scientific research data
<i>Language of text:</i> LT	Serbian (Latin)
<i>Language of abstract:</i> LA	Serbian (Latin) / English
<i>Country of publication:</i>	Serbia

CP	
<i>Locality of publication:</i> LP	Vojvodina
<i>Publication year:</i> PY	2014
<i>Publisher:</i> PU	Author's reprint
<i>Publication place:</i> PP	Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad
<i>Physical description:</i> PD	(7/199/144/13/36/0/3) (chapters/pages/literature/tables/ pictures/graphs/appendix)
<i>Scientific field:</i> SF	Informatics
<i>Scientific discipline:</i> SD	Information Systems
<i>Subject/ Key words:</i> SKW UC	SRU/W, CQL,CERIF, MARC 21, UML, system for searching scientific research data
<i>Holding data:</i> HD	Library of Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad
<i>Note:</i> N	None
<i>Abstract:</i> AB	Aim – The aim is specification, modeling and implementation of search of scientific research data. Search system should provide a rich and

effective search of scientific research data. The basic idea is to enable search based on bibliographic standards SRU/W and CQL, because system need to be able to process search requests from remote computer, and also to provide searching through the web form implemented using Web 2.0 technology where users do not need to know these standards.

Methodology - From the related literature are came the guidelines for the analysis of systems that include scientific researchers data. Based on the analysis necessary functionality and relevant set of indexes that searches CRIS system should support are proposed. A new profile for the search data for scientific research was implemented. It was implemented a standardized system search for CRIS systems based on Web 2.0 technologies, SRU/W profile search and CQL Query Language. The system allows a remote search by entering appropriate requests in accordance with SRU/W standard. UML 2.0 CASE tools based on object-oriented methodology were used for modeling the system. The software architecture is relied on a set of Internet technologies. Implementation technologies are based on Java platform. For the development of advanced interface elements JSF library and AJAX technologies were used.

Results - A unified search profile for CRIS systems and a standardized system of search of scientific research data were developed, which are booth used at the University of Novi Sad. It was specified and implemented a system for search of scientific research data based on Web 2.0 technologies and the CQL query language. GUI is intuitive enough that it can be used by people who do not know the standards on which the search operation is based. Also it was specified and implemented a service that allows remote computer to search by using the

	<p>SRU/W protocol.</p> <p>Practical application - Search system of scientific research data is implemented as part of CRIS UNS system which used at the University of Novi Sad for a long time. CRIS UNS is now able to provide public access and easy search for data of institutions/organizations, researchers and published scientific results within the University of Novi Sad. Supported SRU/W standard makes the CRIS UNS potentially interoperable with external systems.</p> <p>Value – Proposed search profile is some kind of a suggestion how to standardize the search and retrieval of scientific research data. A major advantage of the CRIS profile is that he has no any affect on the implementation of the search system. The application of the proposed profile is very simple because its elements should only be associated (mapped) with the data in existing systems. Ease of application for the profile is practically demonstrated in the CRIS system of the University of Novi Sad (CRIS UNS). Search application implements SRU/W profile and offers potential interoperability with the systems that support the SRU/W standard. After the implementation of the search system, it is obviously that data from CRIS UNS become more accessible and transparent.</p>
<p><i>Accepted by the Scientific Board:</i></p> <p>ASB</p>	<p>31.03.2014.</p>
<p><i>Defended on:</i></p> <p>DE</p>	

<i>Thesis defend board:</i> DB	
<i>President:</i>	Surla Dušan, PhD, professor emeritus, Faculty of sciences, University of Novi Sad
<i>Member:</i>	Zora Konjović, PhD, full professor, Faculty of Technical Sciences, University of Novi Sad
<i>Member:</i>	Miroslav Trajanović, full professor, Faculty of Mechanical Engineering, University of Niš
<i>Member:</i>	Branko Milosavljević, PhD, full professor, Faculty of Technical Sciences, University of Novi Sad
<i>Member:</i>	Zarić Miroslav, PhD, assistant professor, Faculty of Technical Sciences, University of Novi Sad
<i>Member:</i>	Dragan Ivanović, PhD, assistant professor, Faculty of Technical Sciences, University of Novi Sad, advisor