**LOD repozitorijum srpskih propisa**

Filip Jakovljević

Računarstvo i automatika

Fakultet tehničkih nauka, Univerzitet u Novom Sadu

Novi Sad, Srbija

[filipjakovljevic@uns.ac.rs](mailto:filipjakovljevic@uns.ac.rs)

Marko Vuković

Računarstvo i automatika

Fakultet tehničkih nauka, Univerzitet u Novom Sadu

Novi Sad, Srbija

[marko.vukovic9@uns.ac.rs](mailto:marko.vukovic9@uns.ac.rs)

***Apstrakt*—Efikasan sistem za pretraživanje i skladištenje propisa predstavlja osnovu za modernizaciju pravnog sistema države. Servisi koji se trenutno upotrebljavaju ne nude efikasan način pretraživanja velikog broja propisa koji treba da stoje na raspolaganju krajnjim korisnicima. Efikasan sistem za pretraživanje i skladištenje pravnih dokumenta omogućio bi korisnicima velike uštede u vremenu koje bi inače trošili tragajući za informacijama koje su im potrebne. Osnova problema je način predstavljanja metapodataka dokumenata. Jedno od mogućih rešenja je korišćenje ontologija i koncepata semantičkog veba. Ovaj rad predstaviće implementaciju sistema za pretraživanje i skladištenje propisa čiju osnovu predstavlja upravo ontologija koja opisuje metapodatke dokumenata. Ta ontologija formirana je kombinacijom već postojećih ontologija, ali i kreiranjem posebnih elemenata kako bi se modelovale specifičnosti pravnog sistema Republike Srbije. Implementirani sistem daje mnogo veću fleksibilnost prilikom pretraživanja velikog skupa propisa, a takođe omogućuje i efikasniji prikaz veza između dokumenata. Povezivanje sa pojmovima servisa umreženih podataka poput DBpedia predstavlja korak bliže do cilja – mašinske čitljivosti dokumenata i na samom kraju - veba podataka.**

***Ključne reči—propisi; skladištenje; pretraživanje; ontologija; veb aplikacija***

**1. Uvod:**

Propisi su tekstualni dokumenti koji prate određenu strukturu i pravila, ali često njihova struktura ostaje samo ljudski čitljiva i samo na papiru. Prebacivanje tekstualnih dokumenata u elektronsku formu i omogućavanje da oni budu mašinski čitljivi i semantički povezani znatno bi olakšala rad sa ovim dokumentima. Ovaj projekat ima za ideju objavljivanje na internet mašinski čitljivih propisa Republike Srbije, kao i njihovo struktuirano skladištenje u pozadinskoj bazi podataka. Mašinska čitljivost dokumenata ima za cilj da omogući olakšanu i naprednu pretragu kao i da semantički poveže dokumente i pojmove u njima.

Kao pogodan format za objavljivanje propisa izabran je XML format *Akoma Ntoso* (engl. *Architecture for Knowledge - Oriented Management of African Normative Texts using Open Standards and Ontologies*) [1] koji je prilagođen za propise Republike Srbije. Pored njega razvijena je i ontologija koja omogućuje semantičko posmatranje propisa i događaja koji na njih utiču. Ontologija je pravljena po preporukama *Akoma Ntoso* formata i modelovana prema pravnom sistemu Republike Srbije i popunjena je individuama identifikovanim u njemu.

Aplikacija predstavlja model servera i baze, pri čemu je planirano da postoje više servera i baza za različite pravne ustanove koje objavljuju i skladište propise u Republici Srbiji. Ovi serveri bi bili objedinjeni front-end aplikacijom koja bi bila dostupna na internetu. Za pravljenje servera korišćen je *RESTful* arhitektonski stil.

U upotrebi je nekoliko srodnih servisa, pri čemu se izdvajaju *Pravno-informacioni sistem RS* [2] i *Paragraf Lex* [3]. Oba ova servisa nude mogućnost skladištenja i prikazivanja propisa, dok njihove glavne nedostatke predstavljaju skromne mogućnosti pretraživanja i slaba podrška za semantičke veze između dokumenata. Cilj sistema koji se predlaže u ovom radu je da obezbedi sve funkcionalnosti koje nude prethodno navedeni servisi, ali i da otkloni njihove glavne nedostatke.

**2. Akoma Ntoso:**

Akoma Ntoso je internacionalni standard razvijen za reprezentovanje izvršnih, zakonodavnih i sudskih dokumenata. Razvili su ga profesori sa Univerziteta u Bolonji u okviru UN DESA (*United Nations Department of Economic and Social Affairs*) projekta za potrebe pravno informacionog sistema Afrike. Kasniji njegov razvoj nastavlja OASIS (*Organization for the Advancement of Structured Information Standards*) koji su preuzeli standard kao osnovu za svoj jezik za obeležavanje *LegalDocumentML* (*LegalDocML*) legislativnih dokumenata. Standard se danas proširio i izvan okvira Afrike i sve više se primenjuje u svojoj oblasti u celom svetu.

Osim sadržaja propisa Republike Srbije, važni su i metapodaci koji ih dodatno opisuju. Ovi metapodaci sadrže informacije važne za pravnike i druge korisnike sistema predloženog u ovom radu, i omogućuju naprednu pretragu nad dokumentima.

Najvažniji metapodaci koji opisuju dokument, svakako su broj i naziv dokumenta. Broj dokumenta omogućava preciznu i nedvosmislenu identifikaciju propisa u okviru pravnog sistema Republike Srbije, dok naziv dokumenta sažeto predstavlja njegov sadržaj i korisnicima olakšava pronalazak željene informacije. U najopštije metapodatke spada i informacija o državi na koju se propis odnosi. Takođe, postoji kategorizacija dokumenata po sadržaju na tri nivoa apstrakcije: podregistar, grupa i oblast kojoj propis može pripadati. Značajni su i tipovi i podtipovi kojima dokumenti mogu pripadati, koji se ne odnose na sadržaj poput prethodne klasifikacije, već na njihov oblik, način kreiranja i donošenja, kao i opseg važenja. Svaki dokument prate i dodatne napomene koje mogu sadržati različite značajne informacije koje ne moraju biti nužno strukturirane.

Za propise od izuzetnog značaja su i različiti metapodaci koji predstavljaju određene događaje u fazi životnog ciklusa dokumenta. Značajni događaji uključuju, ali se ne ograničavaju na: donošenje pravnog propisa, objavljivanje pravnog propisa, stupanja na snagu pravnog propisa i stupanje pravnog propisa van snage. Za ove događaje, osim datuma, beleže se i odgovorne strane (organizacije ili lica). Za donošenje propisa i drugih pravnih dokumenata, odgovorne strane poput predsednika, vlade, ministarstva ili skupština različitih nivoa, dok se za objavljivanje pre svega misli na zvanične listove za objavljivanje pravnih propisa.

Na manje opštem nivou, značajni su podaci o verziji i jeziku propisa, kao i o formatu u kojem se propis čuva.

**3. Ontologija:**

Ontologija koja je predložena u ovom radu nastala je od osnove koja je formirana kompozicijom nekoliko već postojećih ontologija, a zatim je ta osnova proširena kreiranjem dodatnih klasa koje prilagođavaju Akoma Ntoso standard modelujući specifičnosti pravnih propisa Republike Srbije. Ontologije koje su iskorišćene za formiranje osnove ontologije su: FRBR [4] i SKOS [5]. U nastavku će detaljno biti prikazana rezultujuća ontologija, a [slika 1](#Slika1) sadrži šematski prikaz njenih klasa i objektnih svojstava.

*Akoma Ntoso* standard propisuje opšte klase ontologije za predstavljanje metapodataka propisa. Ove klase dele se u dve kategorije u odnosu na elemente metapodataka koje modeluju:

* klase koje predstavljaju dokumente i
* klase koje predstavljaju sadržaj dokumenata.

Klase koje predstavljaju dokumente u okviru Akoma Ntoso specifikacije direktno su preuzete iz standarda FRBR (*Functional Requirements for Bibliographic Records*) koji održava IFLA (*International Federation of Library Associations*). U nastavku, biće opisane klase i njihova svojstva, preuzete ontologije koja implementira FRBR standard, kao i klase koje su naknado dodate.

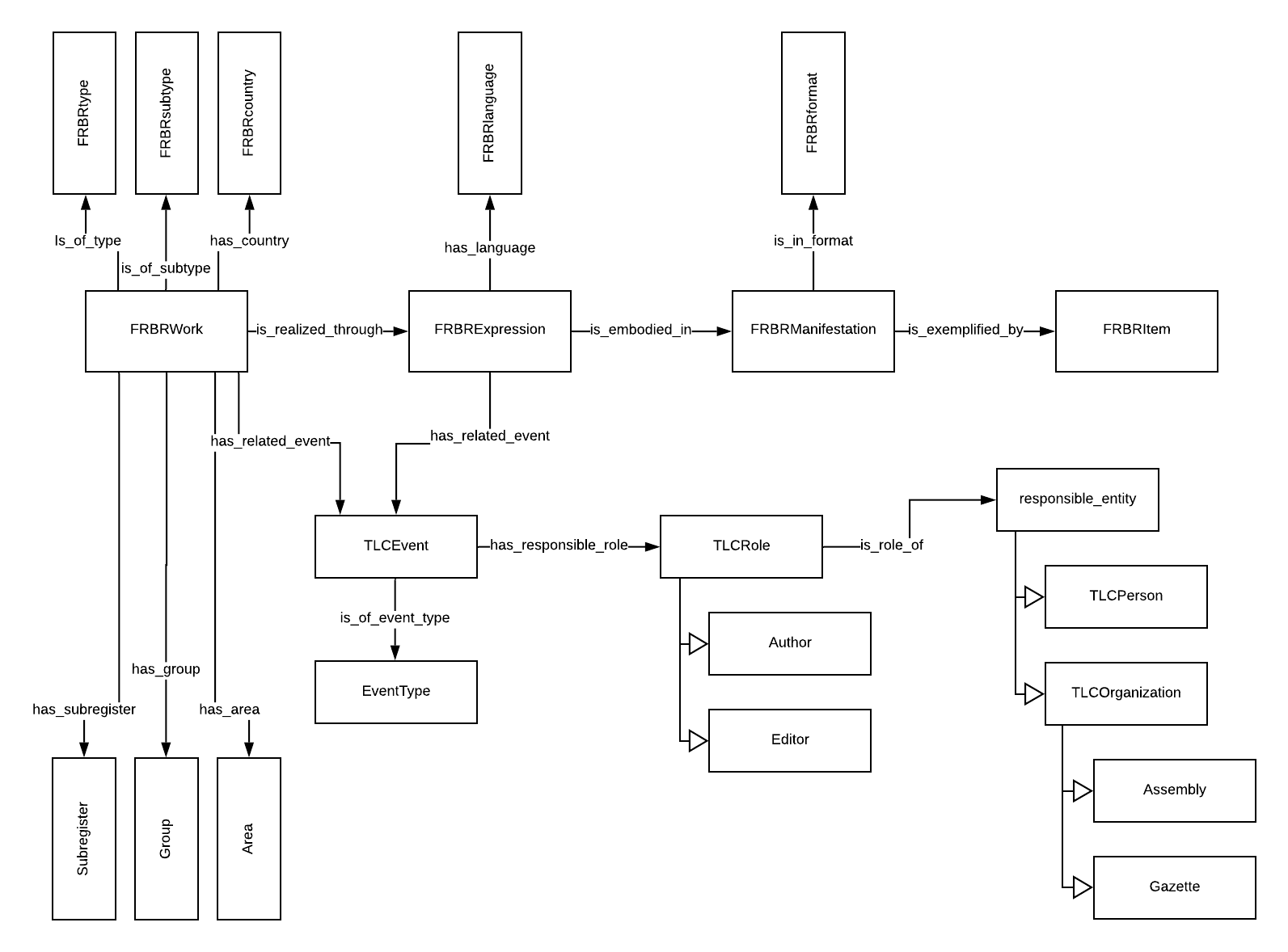
Postoji dve grupe klasa ove ontologije, u odnosu na elemente dokumenta koje predstavljaju. To su klase koje:

* predstavljaju sam dokument i
* predstavljaju svojstva dokumenta.

Prvoj grupi pripadaju klase: *FRBRWork*, *FRBRExpression*, *FRBRManifestation* i *FRBRItem*. *FRBRWork* predstavlja pogled na dokument sa najvišeg nivoa apstrakcije. Odnosi se na konceptualni aspekt propisa i u svojstvima (engl. *data property*) *has\_number* i *has\_name* sadrži informacije o broju i nazivu dokumenta.

U objektnim svojstvima (engl. *object propery*) *is\_of\_type* i *is\_of\_subtype* smešteni metapodaci koji govore o tome kom tipu (*FRBRtype*) i podtipu (*FRBRsubtype*) dokumenta propis pripada. Takođe, objektna svojstva *has\_subregister*, *has\_group* i *has\_area* sadrže metapodatke koji govore kojem podregistru (koji modeluje posebno kreirana klasa *Subregister*), grupi (posebno kreirana klasa *Group*) i oblasti (posebno kreirana klasa *Area*) u skladu sa podelom u okviru pravnog sistema Republike Srbije, pravni dokument pripada. Ove klase predstavljaju potklase klase *Concept*, koja je preuzeta iz SKOS ontologije. Individue ovih klasa međusobno su povezane pomoću objektnih svojstava *has broader* i *has narrower* (takođe preuzete iz SKOS ontologije), koje opisuju njihov hijerarhijski odnos. Podregistar predstavlja najširu kategoriju, Grupa srednju, dok Oblast predstavlja najužu. Takođe, ove klase sadrže i svojstvo *has\_name* koje sadrži podatak o njihovom punom nazivu. Kao što je obrazloženo u odeljku [2](#AkomaNtoso2), veoma značajni su i događaji koji govore o životnom ciklusu dokumenta. Oni se vezuju kako za klasu *FRBRWork*, tako i za klasu *FRBRExpression*. Događaji su implementirani preko drugog tipa klasa koje propisuje standard Akoma Ntoso i o njima će biti reči kasnije. Veoma je značajna informacija i na koju državu se propis odnosi. U ontologiji, ovaj metapodatak smešten je u objektnom svojstvu *has\_country*. Individue koje predstavljaju države preuzete su iz projekta *DBpedia* [6]. *DBpedia* predstavlja projekat koji teži da omogući pristup strukturiranom sadržaju na osnovu informacija objavljenih na internet enciklopediji *Wikipedia*. Povezivanje sa servisom kakav je *DBpedia*, predstavlja korak dalje u težnji da propisi budu mašinski čitljivi, zbog toga što omogućava pristup veoma velikoj bazi definisanih pojmova, što omogućava da ovaj servis koriste i drugi servisi, a ne samo krajnji korisnici, a daje i semantički bogatiji kontekst dokumentima u okviru servisa.

Klasa *FRBRExpression* posmatra propis na nižem nivou apstrakcije u odnosu na prethodno opisanu klasu. Ova klasa modeluje metapodatke koji se odnose na formu dokumenta, odnosno na različite oblike propisa koji mogu postojati. Pre svega se misli na jezike na kojima propis može biti napisan, odnosno na različite verzije dokumenta (dopune i slično). Podatak o jeziku nalazi se u objektnom svojstu *has\_language*, čiji je predikat klasa *FRBRlangugage*. URI individua ove klase predstavlja kod jezika po standardu *ISO 639-2 alpha-3*, dok u svojstvu *has\_name* čuva pun naziv jezika.



Slika 1- Šematski prikaz klasa i objektnih svojstava ontologije

Pogled na dokument klase *FRBRManifestation* niži je za stepen apstrakcije u odnosu na prethodno opisanu klasu. On se bavi metapodacima koji se odnose na format dokumenta. Kao takav, sadrži objektno svojstvo *is\_in\_format*, čiji predikat predstavlja individua klase *FRBRformat*. URI ove klase predstavlja naziv formata u kojem je dokument sačuvan. Osim klasa, svojstava i objektnih svojstava koja su preuzeta iz prethodno navedenih ontologije, dodato je i svojstvo *has\_content*, koje sadrži sadržaj dokumenta u *xml* formatu, zbog odluke da se dokumenti u okviru sistema čuvaju isključivo u tom obliku, a da se ostali formati generišu na osnovu ovog. Na ovaj način obezbeđeno je pojednostavljenje implementacije ontologije, ali je zadržana fleksibilnost sistema koji nudi preuzimanje dokumenata u različitim formatima.

Najniži nivo apstrakcije u pogledu na modelovanje metapodataka dokumenta sadržan je u klasi *FRBRItem*. Individue ove klase zaslužene su za modelovanje lokacije samog pravnog dokumenta. S obzirom na to da sistem opisan u ovom delu ne vodi računa o propisima koji se nalaze van njega, ova klasa nije korišćena prilikom modelovanja, već je zadržana isključivo radi očuvanja potpune usaglašenosti sa *Akoma Ntoso* specifikacijom.

Kako bi se povezale prethodno opisane klase, kreirana su sledeća objektna svojstva: *is\_realized\_through*, *is\_embodied\_in* i *is\_exemplified\_by*. U ontologiji FRBR već su postojala objektna svojstva koja predstavljaju ove relacije, ali uz izvesna odstupanja koja se tiču imenovanja u odnosu na ostatak ontologije, pa je svojstvo *is\_realized\_through*, pored postavljanja osobina domen (engl. *Domain*) na *FRBRWork* i osobina opseg (engl. *Range*) na *FRBRExpression*, takođe i izjednačena sa postojećim svojstvom *has\_realization* pomoću objektnog svojstva koje pripada OWL (*Web Ontology Language*) specifikaciji – *owl:sameAs*. Slično je i sa ostalim objektnim svojstvima, *is\_embodied\_in* ekvivalentno je svojstu *FRBR* ontologije *has\_embodiment* koje povezuje klase *FRBRExpression* i *FRBRManifestation*, a svojstvo *is\_exemplified\_by*, koje povezuje klase *FRBRManifestation* i *FRBRItem*, ekvivalentno je svojstvu *has\_exemplar*.

Drugi tip klasa koji propisuje *Akoma Ntoso* standard, u okviru specifikacije nazvan je TLC klasama (engl. *Top Level Classes*). Ove klase razlikuju se u odnosu na *FRBR* klase po tome što za razliku od njih opisuju sadržaj dokumenata, a ne same dokumente. Kreirane su sve klase koje *Akoma Ntoso* standard navodi, ali su korišćene samo sledeće: *TLCOrganization*, *TLCPerson*, *TLCRole*, *TLCEvent* i *TLCTerm*.

Klase *TLCOrganizacion* i *TLCPerson* predstavljaju potklase klase *responsible entity* i predstavljaju odgovorna lica u donošenju i objavljivanju pravnih propisa. *TLCOrganization* predstavlja organizacije, odnosno udružene grupe ljudi sa istom funkcijom i sadrži dve dodatno kreirane klase *Assembly* – koja modeluje skupštine i *Gazette* – koja modeluje zvanične listove za objavljivanje propisa. *TLCPerson* predstavlja funkciju odgovornog lica koje predstavlja jedan čovek, poput Predsednika Republike Srbije, Predsednika Vlade Republike Srbije i Predsednika Narodne Skupštine Republike Srbije. Individue prethodno nabrojanih klasa u svojstu *has\_name* sadrže naziv organizacije odnosno funkcije koju vrše.

Individue klase *TLCRole*, sa svojim potklasama *Author* i *Editor*, predstavljaju odgovorna lica u ulogama koje u odnosu na pravni propis obavljaju – pre svega se misli na uloge donosioca i objavljivača. Individue ovih klasa povezane su sa individuama klase *responsible entity* i njenih potklasa objektnim svojstvom *is\_role\_of* u kojoj su u ulozi subjekta. Individue klase *TLCEvent* predstavljaju događaje u životnom ciklusu propisa – donošenje, objavljivanje i slično. Sadrže informaciju o ulozi koje u njima ima odgovorno lice – objektno svojstvo *has\_responsible\_role* u kojoj objekat predstavljaju individue klase *TLCRole* i njenih potklasa *Author* i *Editor*, kao i o tipu događaja koje predstavljaju – objektno svojstvo *is\_of\_event\_type*, u kojoj objekat predstavljaju individue klase *EventType*. Instance klase *EventType* sadrže svojstvo *has\_name* koje govori koji tip događaja one predstavljaju. Takođe, individue klase *TLCEvent* sadrže i datumski podatak u svojstvu *has\_date*, koji je predstavljen tipom podataka *xsd:date*. Klase *FRBRWork* i *FRBRExpression* sa individuama ove klase povezane su objektnim svojstvom *TLCEvent*, pri čemu individue tih klasa predstavljaju subjekte u tim relacijama.

Ovako implementirana ontologija pokriva sve prethodno opisane metapodatke pravnih propisa.

**4. Dizajn i implementacija sistema**

U nastavku poglavlja dat je pregled korišćenih tehnologija: SPARQL i Apache Jena Fuseki, a nakon toga je predstavljena implementacija sistema.

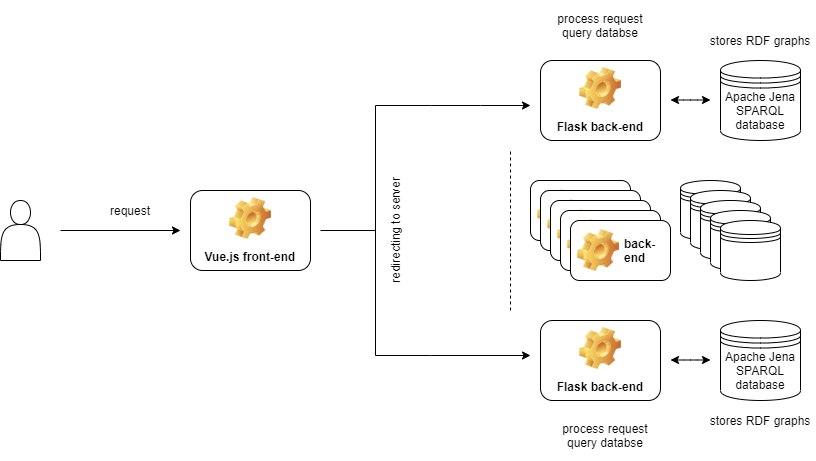
*SPARQL* je rekurzivni akronim za SPARQL Protocol and RDF Query Language [7] i predstavlja upitni jezik za RDF grafove. On nije jedini jezik za upite nad podacima u RDF formatu, ali predstavlja tekuću preporuku W3C (*World Wide Web Consortium*) kao upitni jezik za RDF podatke [8]. Pored toga *W3C* je definisao *SPARQL Protocol* za *RDF* specifikaciju (protokol za zadavanje *SPARQL* upita i prihvatanje rezultata) [9][10].

*SPARQL* omogućava ekstrakciju podataka iz strukturiranih i polustruktuiranih izvora, transformaciju podataka u RDF formatu iz jednog rečnika u drugi i izvršavanje složenih povezivanja (*join* operacija) u jednom jednostavnom upitu [11].

*SPARQL* upiti su upiti trojki i može se reći da oni formiraju rečenice, odnosno članovi trojke imaju značenje subjekta, predikata i objekta u rečenici. Svaki član trojke (subjekat, predikat ili objekat) može biti zamenjen sa promenljivom. Zbog ovakvog načina zadavanja upita su intuitivni i čitki, a pored toga su i jako moćni, jer se kombinacijom relativno prostih rečenica mogu stvoriti grafovi koji vrše vrlo kompleksne pretrage.

*Apache Jena Fuseki* [12] je samostalan *SPARQL* server koji u sebi sadrži čvrsto integrisanu bazu za trajno skladištenje *RDF* grafova. Pored skladištenja omogućava i *SPARQL 1.1* protokole za pretraživanje i ažuriranje grafova kao i *Graph Store HTTP* protokole. Takođe podrzava i *OWL* modele sa kojima se može opisati dodatna semantika. Objavljen je pod otvorenom licencom i napisan je u programskom jeziku *Java*. Pouzdan je i jednostavan za korišćenje, pa ispunjava sve što je potrebno te je izabran da se u njemu skladište *RDF* grafovi.

*Apache Jena Fuseki* baza je obmotana serverom koji prati *RESTful* obrazac i omogućuje dodatne funkcionalnosti kao sto su provere, zaštite i transformacije podataka iz jednog oblika u drugi. Arhitektura rešenja, koja je prikazana na [slici 2](#Slika2), podrazumeva postojanje većeg broja instanci servera kao i većeg broja instanci *Apache Jena Fuseki* baze. Arhitektura sistema nudi mogućnost da svako odgovorno lice koje objavljuje propise u elektronskom obliku ima svoju instancu servera i baze, a da je pritom omogućen pristup svim objavljenim propisima, bez obzira na njihovu lokaciju u okviru jednog korisničkog interfejsa. Na primer, Službeni glasnik Republike Srbije i Službeni list Autonomne pokrajine Vojvodine imali bi zasebne instance servera i baze na kojima bi objavljivali propise, a korisnici bi mogli da pretražuju dokumente kao da se nalaze na jednom mestu, bez poznavanja njihove lokacije. Za implementaciju korisničkog interfejsa iskorišćen je *Quasar Framework*, radno okruženje za *Vue.js* koji predstavlja *JavaScript* okruženje za razvoj veb aplikacija i korisničkih interfejsa. Pošto je baza na zasebnom serveru i radi se sa *RDF* grafovima nije bilo potrebe za *ORM* (*object-relational mapper*) delom koji razni veb okviri nude i izabran je krajnje jednostavan *Flask* veb okvir [13] u *Python* programskom jeziku. *Flask* je lagani *WSGI* (engl. *Web Server Gateway Interface*) veb okvir [14], koji se ubraja u mikro okvire. Dizajniran je vrlo jednostavno i minimalistično, ali ima sposobnosti da se lako nadograđuje, gde sam programer bira biblioteke i alate koje će nadograditi na njega [14]. Trenutno je jedan od najpopularnijih *Python* veb okvira [15].



Slika 2 - Prikaz arhitekture sistema

Poznavanjem URI-ja individue koja pripada nekoj od FRBR klasa (Work, Expression, Manifestation) moguće je vršiti CRUD operacije nad njima.

Slanjem POST zahteva na URL koji se sastoji od URL-a servera i URI-ja individue vrši se ekstrakcija URI-ja iz linka i pristupa se kreiranju grafa koji odgovara prosleđenom *Akom Ntoso XML* formatu. Prvo se proverava da li individua sa tim URI-jem već postoji, zatim se vrši validacija XML-a XML šemom. Nakon uspešne validacije kreira se RDF graf koji se zatim ponovo validira sa RDF šemom. Ako je validacija uspešna RDF graf se upisuje u bazu.

Slanjem GET zahteva na URL koji se sastoji od URL-a servera i URI-ja individue vrši se ekstrakcija URI-ja iz linka i traži se da u bazi da li individua sa tim URI-jem postoji. Ako postoji vraća se dokument na koji se odnosi.

Ako se zahtev odnosi na *Work* vraća se najnoviji dokument u podrazumevanom *Akoma Ntoso* XML formatu i podrazumevanom srpskom jeziku.

Ako se zahtev odnosi na *Expression* vraća se najnoviji dokument u podrazumevanom *Akoma Ntoso* XML formatu.

Ako se zahtev odnosi na *Manifestation* vraća se dokument u traženom formatu. Trenutno podržani formati su *Akoma Ntoso* XML, XHTML i PDF. U bazi se čuvaju samo XML dokumenti na kojima se posle vrše transformacije. XML se sa XSLT transformacijom pretvara u XHTML, a XHTML se pomoću *pdfkit*-a pretvara u PDF.

Ažuriranje je složena operacija i sastoji se od kompozicije operacija brisanja i upisivanja.

Slanjem PUT zahteva na URL koji se sastoji od URL-a servera i URI-ja individue vrši se ekstrakcija URI-ja iz linka i proverava se da li individua sa tim URI-jem postoji. Ako individua postoji vrši se pretraživanje RDF grafa u bazi i brišu se svi tripleti u kojima ona direktno ili sporedno učestvuje.

Posle uspešnog brisanja sledi upisivanje u bazu. Prvi korak je validacija prosleđenog XML-a XML šemom. Nakon uspešne validacije kreira se RDF graf koji se zatim ponovo validira sa RDF šemom. Ako je validacija uspešna RDF graf se upisuje u bazu.

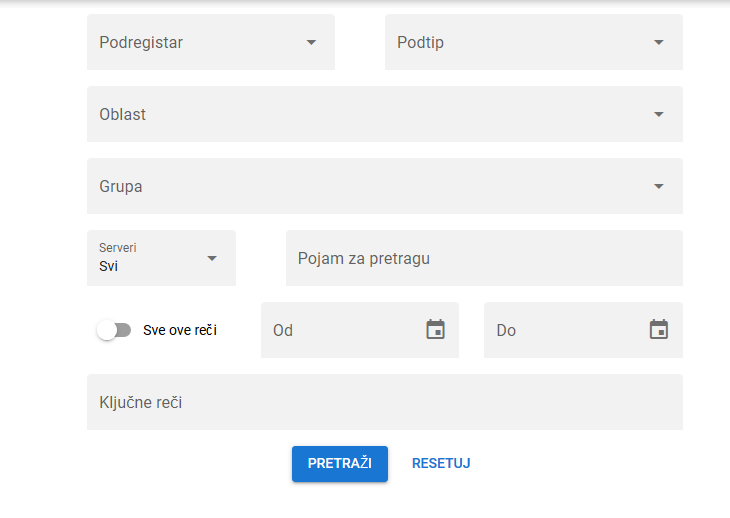
Slanjem DELETE zahteva na URL koji se sastoji od URL-a servera i URI-ja individue vrši se ekstrakcija URI-ja iz linka i proverava se da li individua sa tim URI-jem postoji. Ako individua postoji vrši se pretraživanje RDF grafa u bazi i brišu se svi tripleti u kojima ona direktno ili sporedno učestvuje.

Sistem nudi i tri mogućnosti za pretragu podataka u bazi: jednostavnu pretragu, pretragu preko formulara i pretragu preko SPARQL upita., koja predstavlja najfleksibilniji vid pretrage. Upit biva direktno prosleđen bazi koja vraća RDF graf koji ga zadovoljava. Ovakva vrsta pretrage namenjena je korisnicima koji su upoznati sa tehnikama semantičkog veba i SPARQL upitima i omogućava vrlo korisne i kompleksne pretrage.

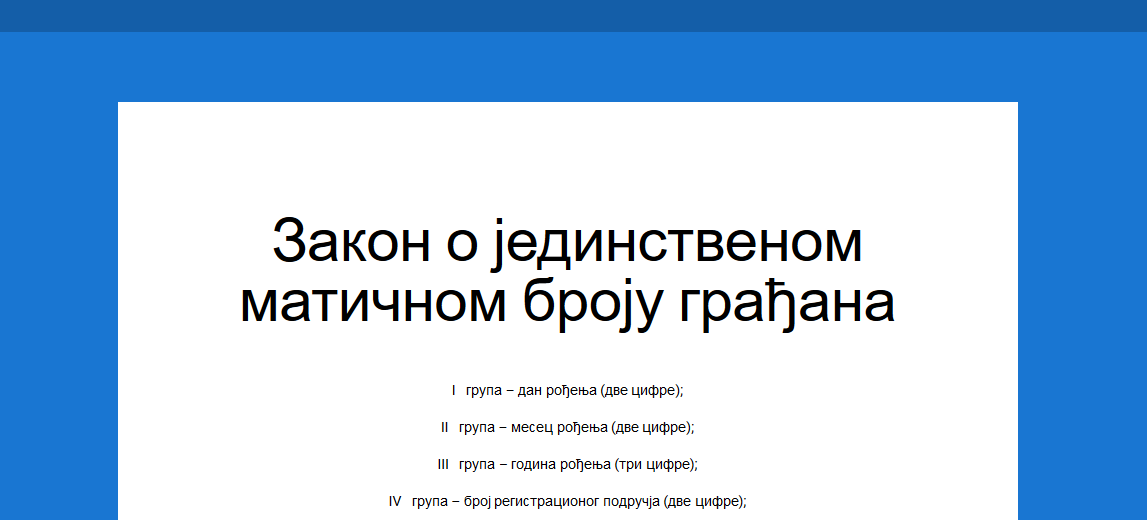
**5. Demonstracija**

Korisnički interfejs sastoji se iz sledećih stranica: stranica za dodavanje (slika 6), uklanjanje i izmenu dokumenata i stranica za jednostavnu (slika 3) i naprednu pretragu (slika 5). Iz stranice za jednostavnu pretragu, korisnik ima mogućnost pretraživanja svih aktivnih servera, tako što unese upit koji pretražuje dokumente po nazivu. Takođe, nudi mu se i mogućnost da bazu pretražuje po rečima koje je uneo pojedinačno. Klikom na dugme pretraži, izlistavaju mu se pronađeni propisi. U poslednjoj koloni tabele, nakon osnovnih informacija o propisima dostupni su mu linkovi za pristup dokumentu u sledećim formatima: XHTML, XML i PDF. Klikom na poslednja dva, pokreće se preuzimanje dokumenta, dok se klikom na prvo dugme otvara prozor za prikaz dokumenta (slika 4). U okviru prozora za prikaz dokumenta, korisniku je omogućeno da se navigira na povezane dokumente klikom na linkove.  
Stranice za dodavanje, uklanjanje i brisanje ekvivalentne su, zahtevaju od korsnika da izabere dokument kojim želi da manipuliše, zatim da izabere URI nad kojim bi želeo da izvrši izabranu operaciju.

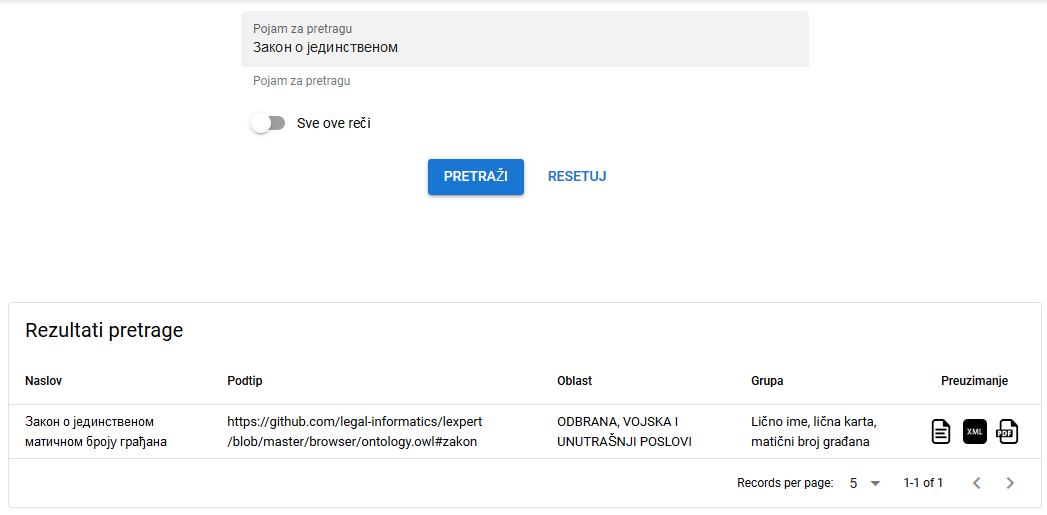
**6. Zaključak**



Slika 5- Stranica za naprednu pretragu



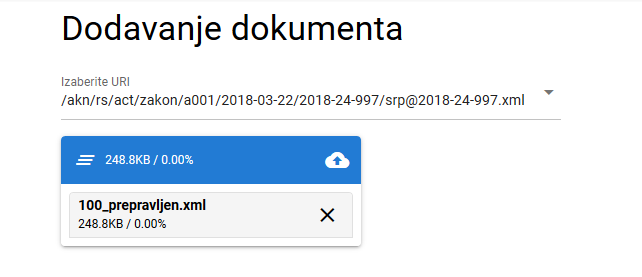
Slika 4- Stranica za prikaz dokumenta



Slika 3- Prikaz stranice za osnovnu pretragu

**6. Zaključak**

U ovom radu predstavljen je servis za pretraživanje pravnih propisa Republike Srbije. Opisani servis razlikuje se od postojećih po tome što pravne propise predstavlja korišćenjem ontologija i tehnika semantičkog veba. Servis nudi sve funkcionalnosti koje nude drugi servisi ovog tipa, a osim toga nudi i napredniji mehanizam pretrage i bolju integraciju između povezanih dokumenata. Pored toga, integracijom sa servisom *DBpedia*, odlazi i korak dalje u težnji ka mašinskoj čitljivosti dokumenata, čime obezbeđuje bogatiji kontekst propisima koje skladišti, ali i predstavlja osnovu za povezivanje ovog sistema sa drugim. Ontologija je realizovana kompozicijom drugih postojećih ontologija, ali i kreiranjem klasa, svojstava i objektnih svojstava koje modeluju specifičnosti pravnog sistema Republike Srbije. Kombinovanjem tehnologija semantičkog veba sa veoma rasprostranjenim tehnologijama RESTful servisa, omogućeno je to da prednosti ontologija ne ostanu samo na papiru kao što sa njima često bude slučaj, već da dođu u kontakt sa krajnjm korisnicima.



Slika 6- Stranica za dodavanje dokumenata

Nedostatak sistema predstavlja zavisnost o postojanju specifično formatiranih dokumenata u kojima se očekuje bogat skup metapodataka. Drugi servisi koji su prethodno opisani, iako imaju skromniji skup funkcionalnosti, imaju izvesnu prednost nad ovim sistemom, zbog toga što imaju pristup većem broju dokumenata. Nije realno očekivati da se izuzetno velik broj propisa koji su u upotrebi u okviru pravnog sistema Republike Srbije prenese u ovaj format ručno. Međutim, softversko rešenje koje bi dopunilo servis opisan u ovom radu, značajno bi ga poboljšalo, jer bi na taj način omogućilo automatizovanu konverziju propisa i ustupilo na raspolaganje korisnicima sve prednosti koje su njime ostvarene. Još jedna mogućnost za poboljšanje prethodno opisanog rešenja leži u tome da se dokumenti semantički bogatije opišu. To bi omogućilo unapređivanje prethodno opisanih funkcionalnosti kao što je pretraga, a najviše bi doprinelo naprednijem referenciranju među dokumentima, pri čemu bi korisnik klikom na link koji jedan propis povezuje sa drugim mogao dobiti onu informaciju koju je tražio, a ne ceo dokument.

**Reference:**

1. Akoma Ntoso | Akoma Ntoso Site. 2020. Akoma Ntoso | Akoma Ntoso Site. [ONLINE] Available at: http://www.akomantoso.org/. [Accessed 01 June 2020].
2. Правно-информациони систем РС. 2020. Правно-информациони систем РС. [ONLINE] Available at: <http://www.pravno-informacioni-sistem.rs/>. [Accessed 01 June 2020].
3. Paragraf Lex. 2020. Paragraf Lex | Elektronska pravna baza | Online baza propisa. [ONLINE] Available at: <https://www.paragraf.rs/>. [Accessed 01 June 2020].
4. vocab.org. 2020. Expression of Core FRBR Concepts in RDF. [ONLINE] Available at: https://vocab.org/frbr/core. [Accessed 01 June 2020].
5. World Wide Web Consortium (W3C). 2020. SKOS Simple Knowledge Organization System Namespace Document 30 July 2008 "Last Call" Edition. [ONLINE] Available at: https://www.w3.org/TR/2008/WD-skos-reference-20080829/skos.html. [Accessed 01 June 2020].
6. DBpedia. 2020. DBpedia. [ONLINE] Available at: https://wiki.dbpedia.org/. [Accessed 01 June 2020].
7. Beckett, Dave (6 October 2011). "What does SPARQL stand for?". [semantic-web@w3.org](mailto:semantic-web@w3.org).
8. Semantic Web Activity News. 2020. Eleven SPARQL 1.1 Specifications are W3C Recommendations. [ONLINE] Available at: https://www.w3.org/blog/SW/2013/03/21/eleven-sparql-1-1-specifications-are-w3c-recommendations/. [Accessed 01 June 2020].
9. www.w3.org. 2020. SPARQL Query Language for RDF. [ONLINE] Available at: https://www.w3.org/TR/rdf-sparql-query/. [Accessed 01 June 2020].
10. www.w3.org. 2020. SPARQL Protocol for RDF. [ONLINE] Available at: https://www.w3.org/TR/rdf-sparql-protocol/. [Accessed 01 June 2020].
11. www.w3.org. 2020. SPARQL By Example. [ONLINE] Available at: https://www.w3.org/2009/Talks/0615-qbe/. [Accessed 01 June 2020].
12. Apache Jena. 2020. Jena documentation overview. [ONLINE] Available at: https://jena.apache.org/documentation/. [Accessed 01 June 2020].
13. Flask. 2020. Flask Documentation (1.1.x). [ONLINE] Available at: https://flask.palletsprojects.com/en/1.1.x/. [Accessed 01 June 2020].
14. GitHub. 2020. pallets/flask: The Python micro framework for building web applications.. [ONLINE] Available at: https://github.com/pallets/flask. [Accessed 01 June 2020].
15. GitHub. 2020. Search · stars:>100 · GitHub. [ONLINE] Available at: https://github.com/search?l=Python&q=stars%3A%3E100&s=stars&type=Repositories. [Accessed 01 June 2020].