

SEMANTIČKI WEB – IDEJA I TEHNOLOGIJE

Ljubiša Milivojević

Informacione tehnologije,
Beograd

Kontaktни naslov:
ljubisa_milivojevic@yahoo.com

Rezime

Semantički web je tehnologija koja će možda imati ključnu ulogu u razvoju informatike u narednim godinama. Na sličan način na koji je klasični web učinio dokumente i multimedijски sadržaj dostupnim čoveku, semantički web bi trebalo da najrazličitije podatke učini dostupnim različitim softverskim komponentama, da bi se tako otklonili neki nedostaci i proširilo polje primene weba. U članku se objašnjava zašto se u okviru tehnologija postojećeg weba ne mogu postići navedeni ciljevi i daje se prikaz principa i najvažnijih tehnologija na kojima se zasniva semantički web.

Ključne reči

semantički web, internet, softverska arhitektura, model podataka, rečnik podataka, ontologija

Izvlaček

Semantični splet je tehnologija, ki bo verjetno imela ključno vlogo pri razvoju informatike v naslednjih letih. Na podoben način kot je klasični splet naredil, da so dokumenti in multimedijске vsebine dostupne človeku, bi moral semantični splet omogočiti, da so najrazličnejši podatki dostopni različnim komponentam programske opreme, da bi se tako odpravile neke pomanjkljivosti in bi se razširila uporaba spleta. V članku pojasnjujemo, zakaj se v okviru tehnologije obstoječega spleta navedeni cilji ne morejo doseči, in predstavljamo principe in najpomembnejšo tehnologijo, na kateri je zasnovan semantični splet.

Ključne besede

semantični splet, internet, arhitektura programske opreme, model podatkov, slovar podatkov, ontologija

Abstract

The semantic web is a technology that in next few years will most probably play a key role in the development of informatics. Similar to what the standard web has done in the field of public access to documents and multimedia content, also the semantic web should do and allow that a variety of data be accessible to different software components in order to rectify certain deficiencies and expand the scope of web use. The article explains why the said goals cannot be achieved in the context of the existing web technology; it also presents the principles and the most important technologies the semantic web is being designed on.

Key words

semantic web, internet, software arhitecure, data model, data dictionary, ontology

UVOD

Personalni računari, internet i mobilne komunikacije su tehnologije koje su obeležile kraj prošlog i početak ovog veka. Međutim, jedan od internet servisa, web, uspeo je da preraste svoje tehnološke okvire postavši kulturni i društveni fenomen koji je ušao praktično u sve pore našeg života. Web koristimo za posao, zabavu, informisanje, preko njega kupujemo i prodajemo, sklapamo poznanstva i družimo se. Web je jedini slobodan medijum na kome mogu da se izraze praktično svi koji imaju nešto da kažu,

a i oni koji to nemaju. Ova sloboda donosi i neke probleme: iz obilja informacija koje postoje na webu, teško je izdvojiti one koje su nam potrebne, jer često nemamo mehanizam za proveru verodostojnosti informacija, niti možemo da kombinujemo informacije iz različitih izvora. Ništa manje važni su razni etički i pravni problemi koji su povezani sa webom, ali se u članku njima nećemo baviti.

Ovaj članak se bavi semantičkim webom, koji predstavlja pokušaj da se primena weba još više proširi, da on od komunikacionog medija postane globalni informacioni

sistem. Pored toga, on bi trebalo da ublaži neke od navedenih problema vezanih za primenu postojećeg weba. Semantički web počiva na ideji da informacije na webu postanu mašinski čitljive. Znači da se umesto dokumenata povezanih hiperlinkovima (u koje može biti ugrađen različiti multimedijски sadržaj), koriste međusobno povezani podaci koji imaju specifikovanu strukturu i značenje. Pri tom je cilj očuvati slobodu koja vlada na postojećem webu.

Članak započinjemo primerom koji ilustruje ograničenja postojećeg weba i pokazuje da se ona ne mogu rešiti u okviru tehnologija koje se u njemu koriste. U nastavku se objašnjavaju principi semantičkog weba, odnos prema postojećem webu i njegova arhitektura. Veći deo članka posvećen je novim tehnologijama za predstavljanje informacija i znanja koje su u osnovi semantičkog weba. U prikazu tehnologija trudili smo se da ne ulazimo u tehničke pojedinosti, ali smo takođe pokušali i da ih predstavimo dovoljno detaljno, kako bi čitalac stekao opštu sliku o njima.

PRIMER POTREBE ZA NOVIM WEBOM

U nedelju po podne setili smo se da uveče treba da odemo na rođendan prijatelju. Odlučili smo se da mu poklonimo knjigu, znamo da voli špijunske romane i da nije pročitao nijedan od novijih romana Džona le Kareja (John le Carré). S obzirom na to da se ovde bavimo tehnologijama i njihovim nedostacima, a ne problemom razvoja weba u Srbiji, pretpostavimo da naše knjižare imaju svoje web lokacije i na tim lokacijama kataloge knjiga koje prodaju. Budući da nemamo vremena za obilazak knjižara, preko weba pokušavamo da pronađemo knjižaru u našem gradu (npr. u Beogradu), koja radi nedeljom po podne i koja ima neki od poslednjih le Kareovih romana. Pogledajmo sa kakvim se problemima susrećemo.

Najpre treba da ustanovimo da li je kod nas u poslednjih nekoliko godina objavljen neki le Kareov roman. To možemo proveriti u nekoj bibliografskoj bazi podataka i ubrzo, npr. korišćenjem kataloga Narodne Biblioteke Srbije, saznajemo da je 2006. godine objavljen roman "Savršeni prijatelji", 2007. "Misionarska pesma", a 2008. "Brižni baštovan". Za početak nije loše, međutim, uskoro ćemo se susresti sa znatno većim problemima.

Sada nam predstoji da pronađemo spisak knjižara koje u Beogradu rade nedeljom po podne. Naivno postavljen upit Guglu (Google) "*knjizara beograd*" nudi 304.000 pogodaka. Pokušavamo da koristimo neke od imenika na webu i uviđamo da su podaci na njima zastareli i nepotpuni. Verovatno bi većina nas u ovom trenutku odustala i u najbližoj prodavnici kupila kao poklon piće ili bombonjeru, ali na isti način na koji smo pretpostavili da knjižare

imaju web lokacije sa katalogima knjiga koje prodaju, možemo zaključiti da postoji i ažuran adresar knjižara. Drugim rečima, neažurne web lokacije i imenici na webu posledica su organizacionih problema, a ne principijelnih i tehnoloških ograničenja weba.

Nastavljamo našu potragu za idealnim poklonom i posećujemo jednu po jednu web lokaciju knjižara, proveravamo njihovo radno vreme i pretražujemo katalog artikala. Kada pronađemo knjižaru, koja radi nedeljom i poseduje neki od romana koje tražimo, prekidamo pretraživanje.

Naravno, idealno bi bilo da nekom pretraživaču možemo postaviti pitanje "Koje knjižare u Beogradu, koje rade nedeljom po podne, imaju neki od Le Kareovih romana koji su objavljeni u poslednjih nekoliko godina?", uz zahtev da rezultate sortira, na primer, po ceni pronađenih knjiga i udaljenosti knjižara od mesta gde stanujemo. Pogledajmo koja su to tehnološka ograničenja koja onemogućavaju postojanje ovakvih pretraživača:

- Pretraživač bi trebalo da razlikuje prezentacije knjižara od drugih dokumenata u kojima se knjižare samo pominju.
- Pretraživač bi iz prezentacije morao da izdvoji informaciju o adresi i radnom vremenu – problem je što su ovi podaci skriveni negde u tekstu i to često ne na istoj strani prezentacije.
- Potrebno je da postoji jedinstven jezik za pretraživanje kataloga knjižarskih artikala. Takođe, rezultati upita moraju biti u obliku koji pretraživač može da protumači.

Rešenje navedenih problema može se potražiti i u unifikaciji prezentacija (dokumenata). Može se, na primer, tražiti da sve početne stranice knjižara budu označene na odgovarajući način, da se podaci o radnom vremenu i adresi nalaze na odgovarajućim stranama i da budu u obliku koji omogućava njihovo mašinsko izdvajanje iz strane, kao i da sve knjižare koriste isti upitni jezik i isti oblik predstavljanja rezultata pretraživanja kataloga. Ovo rešenje je krajnje nepraktično (teško je zamisliti organizaciju koja bi propisivala gde će i šta knjižare držati na svojim prezentacijama) i kosi se sa osnovnim principima na kojima web počiva i koji su zaslužni za njegovu popularnost. Razmatranje ovih principa pomoći će nam da shvatimo motive koji stoje iza koncepta i tehnologija semantičkog weba.

PRINCIPI NA KOJIMA POČIVA WEB

Razvoj weba zasnovan je na sledećim tehnološkim i organizacionim principima:

- jednostavnost,
- modularni dizajn,

- decentralizacija (distribucija),
- tolerancija,
- distribucija.

Prva dva principa odnose se na softverski inženjering i za nas nisu posebno zanimljivi. Druga dva principa objašnjavaju pravac razvoja semantičkog weba.

Decentralizacija se ne odnosi samo na činjenicu da se informacije nalaze na različitim lokacijama na internetu, već i na to da je svako odgovoran za informacije koje postavi na web – i to kako za njihov sadržaj, tako i za oblik u kome se one nalaze. Ne postoji nikakvo regulativno telo koje kaže šta može, a šta ne može da se postavi na web (naravno pod uslovom da je u okviru zakona). Šta više, decentralizacija se ne odnosi samo na sadržaj, već i na tehnološki razvoj weba – proizvođači softvera i pružaoči različitih usluga na webu mogu u njega da uključuju sopstvena tehnološka rešenja. Tržište, odnosno korisnici, treba da daju konačan sud o tome šta će od tih rešenja preživeti, a šta neće.

Sledeći važan princip razvoja weba jeste tolerancija pod kojom se podrazumeva da nove tehnologije ne smeju onemogućiti korišćenje starih tehnologija. Time se obezbeđuje postepen, evolutivan razvoj weba.

IDEJA I OBLICI SEMANTIČKOG WEBA

Ideja semantičkog weba je jednostavna, a to je iskoristiti znanja, principe i tehnologije, koje su u osnovi običnog weba, za novi web koji bi bio univerzalan medijum za razmenu podataka, informacija i znanja.¹ Analizom navedenog primera može se uočiti da je problem postojećeg weba u tome što je oblikovan tako da njegov korisnik bude čovek, a ne mašina. Videli smo da stvari koje su za nas kao korisnike veoma jednostavne, npr. da prepoznamo prezentaciju knjižare ili da u toj prezentaciji prepoznamo i protumačimo adresu i radno vreme, za pretraživač, bez obzira na to koliko savršen on bio, predstavljaju nerešiv problem. Zato je potrebno uvesti potpuno nov način predstavljanja podataka na webu, koji će biti mašinski čitljiv i omogućiti različitim programima da izdvoje značenje iz njih, a pri tom biti dovoljno fleksibilan da pokrije različite primene i da omogući sopstveni razvoj.

Pre nego što pređemo na tehnologije semantičkog weba, potrebno je da razmotrimo u kakvoj će vezi on biti sa postojećim, nesemantičkim webom. Predviđa se da će postojati dva oblika semantičkog weba. To su:

1. Klasičan web u kome će dokumenti, pored oblika prilagođenog čoveku, sadržati i mašinski čitljive informacije. Drugim rečima, u obične HTML/XHTML stranice upisuju se podaci koji omogućavaju da se

iz njih programski izdvoji značenje. Na ovaj način, mi semantički web integrišemo u postojeći web; isti dokumenti se koriste za prikaz korisnicima i za računarsku obradu. Pored tehnologija, koje su specifične za semantički web (RDF(S)), za ovu namenu danas se dosta koriste različite varijante mikroformata.²

2. Semantički web kao poseban web podataka, pri čemu mogu postojati reference iz semantičkog na dokumente običnog weba i obrnuto.

Ova dva oblika semantičkog weba ne isključuju se međusobno, iako drugi oblik (zaseban web podataka) ima neosporne prednosti, te je verovatno da će, bar u početku, postojati oba oblika.

Kada govorimo o migraciji sa običnog na semantički web, treba napomenuti da se ulažu veliki naponi u razvoj tehnologija za automatsko izdvajanje značenja iz nestrukturiranog teksta. Ove tehnologije bi trebalo znatno da olakšaju migraciju.

PRETPOSTAVKA O OTVORENOM SVETU

Ako ne može da dođe do neke informacije, svaki korisnik weba intuitivno prihvata da ne može izvoditi nikakve zaključke o njenoj istinitosti. Recimo, ako ne možemo da pronađemo adresu knjižare u našem susedstvu, koja radi nedeljom po podne, to ne znači da ta knjižara ne postoji. Za takve sisteme kažemo da su zasnovani na pretpostavci o otvorenom svetu.

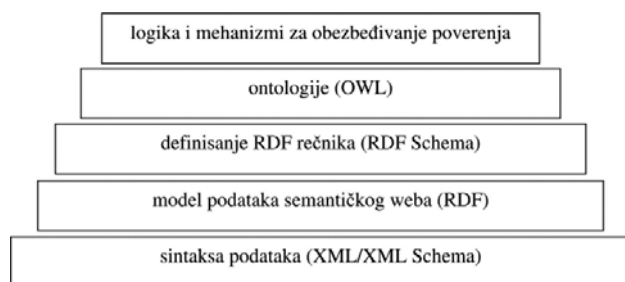
S druge strane, u većini drugih informacionih sistema, nepostojanje neke informacije ekvivalentno je tvrdnji da ta informacija nije istinita. Ako ste prodavac, iz činjenice da u svom informacionom sistemu nemate evidentiranu uplatu nekog kupca, zaključićete da kupac tu uplatu nije ni izvršio. Na delu je pretpostavka o zatvorenom svetu.

Bez obzira na to što bi trebalo da se ponaša kao ogroman informacioni sistem, semantički web se zasniva na pretpostavci o otvorenom svetu. Ovo je direktna posledica principa decentralizacije i može nam pomoći da shvatimo motive koji leže iza izbora tehnologija koje se u njemu koriste.

ARHITEKTURA SEMANTIČKOG WEBA

Kao i većina informacionih tehnologija, semantički web ima slojevitou arhitekturu. Glavni elementi ove arhitekture su prikazani na slici 1.

S obzirom na to da će semantički web koristiti postojeću mrežnu infrastrukturu (internet), na slici nisu prikazani mrežni protokoli.



Slika 1: Arhitektura semantičkog weba

U nastavku članka opisaćemo tehnologije koje čine pojedine slojeve prikazane arhitekture.

XML I XML SCHEMA

Osnovu svih tehnologija semantičkog weba čini Extensible Markup Language (XML). XML je već sada zrela tehnologija koja nalazi široku primenu na webu, ali i u brojnim drugim oblastima računarstva. Na semantičkom webu, XML bi trebalo da bude opšti format za razmenu informacija.³

Na slici 2 su data dva moguća XML zapisa sa informacijama o adresi i radnom vremenu neke hipotetične knjižare.

```
<xml>
<prodavnica vrsta='knjizara'>
  <nazivo>Tristero</nazivo>
  <adresa>
    < mesto='Beograd' />
    < ulica='Pinčonova' />
    < broj='49' />
  < radno_vreme>
    < radni_dan od='9' do='22' />
    < subota od='9' do='22' />
    < nedelja od='12' do='22' />
  < /radno_vreme>
< /prodavnica>
< /xml>
```

```
<xml>
<knjizara naziv='Tristero'>
  <adresa mesto='Beograd' ulica=' Pinčonova' broj='49' />
  <radno_vreme>
    <ponedeljak od='9' do='22' />
    <petak od='9' do='22' />
    <subota od='9' do='22' />
    <nedelja od='12' do='22' />
  </radno_vreme>
</xml>
```

Slika 2: Dva moguća opisa hipotetičke knjižare

Ovde se nećemo baviti opisom XML-a, jer će svima koji su se susretali sa HTML-om navedeni primeri biti jasni. Ono što želimo da pokažemo je to kako se iste informacije u XML-u mogu prikazati na sasvim različite načine. Razlike mogu biti formalne (da li ćemo neki podatak staviti u atribut ili ćemo ga staviti u tekstualni deo datog elementa), ali i suštinske – prvi XML u primeru pretpostavlja da je knjižara podvrsta prodavnice, u drugom slučaju to je poseban entitet. Aplikacija koja treba da koristi XML opis knjižare mora da pozna strukturu XML-a. U nekim primenama ova struktura može da se podrazumeva. Kada je reč o webu, čak i ako postoji neka dogovorena struktura podataka, aplikacije se mogu susretati sa dokumentima koji koriste različite verzije ove strukture, zbog čega XML poseduje mehanizme da se u okviru dokumenta eksplicitno specifikuje njegova struktura.

XML Schema je mehanizam za definisanje strukture XML dokumenta. Glavni element ovog mehanizma je jezik za definisanje sheme dokumenata – XML Schema

Definition Language (XSD). Na osnovu XML sheme program može da odredi da li je neki dokument formalno ispravan, na primer, da li sadrži samo definisane elemente, kao i to da li ovi elementi imaju ispravne attribute i podelemente. Takođe se mogu postaviti ograničenja koja se tiču tipova podataka koji se mogu koristiti u nekom kontekstu i broja pojavljivanja elementa (kardinalnosti) u okviru nekog drugog elementa. Ovde je reč o elementima koji se tiču semantike podataka, međutim XML i XML Schema nisu pogodni da se direktno koriste za semantički web. Problem je u tome što su to dosta krute tehnologije – mala promena u shemi može zahtevati izmenu u svim dokumentima koji se na njoj zasnivaju. U slučaju malog broja shema, koje održavaju eksperti, može se kontrolisati kompatibilnost sa prethodnim verzijama, ali to nije izvodljivo u semantičkom webu.

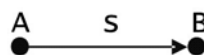
Drugi problem sa XML i XML Schema tehnologijama je u tome što one omogućavaju da se u istom dokumentu jednostavno koristi više shema.

MODEL PODATAKA SEMANTIČKOG WEBA – RDF

Najjednostavniji način za postizanje fleksibilnosti informacionih tehnologija jeste njihovo pojednostavljivanje. Web svoj uspeh duguje tome što je, bar u početku, bio veoma jednostavan. Postojao je tekstualni dokument sa najosnovnijim naredbama za formatiranje i mogućnošću da se u njega ugrađuju multimedijiski sadržaj i linkovi na druge dokumente. Sistem je bio jednostavan za upotrebu, dokumente je bilo moguće publikovati i pomoću običnih programa za uređivanje teksta, a postojale su implementacije programa za pregled web sadržaja koje su se mogle koristiti čak i na alfanumeričkim terminalima. Vremenom su tehnologije postale složenije, što je znatno proširilo polje njihove primene.

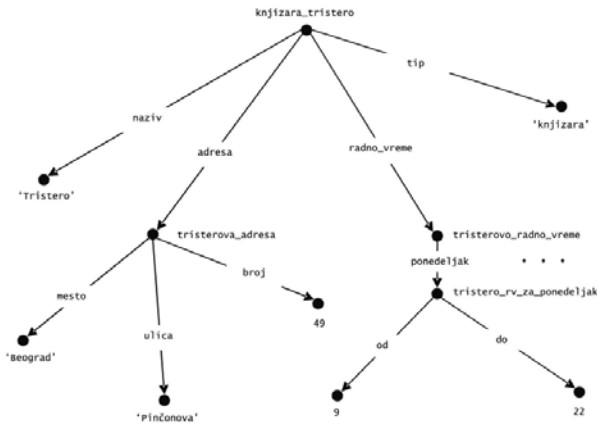
Resource Definition Framework (RDF) trebalo bi da u semantičkom webu bude ono što je prvobitni HTML bio u običnom webu. Osnovna razlika je u sadržaju koji se postavlja na web – to više nisu međusobno povezani dokumenti nego proizvoljni podaci.

Jedna od najčešće korišćenih struktura za predstavljanje podataka u informatici jesu grafovi.⁴ Recimo, informaciju da je vrednost svojstva “s”, objekta “A”, neki objekat “B” možemo prikazati sledećim grafom:



Slika 3: Jednostavan graf koji govori da objekat “A” ima svojstvo “s” čija je vrednost “B”

Slika 4 prikazuje informacije o adresi i radnom vremenu knjižare iz našeg primera u obliku grafa.



Slika 4: Osnovni podaci o knjižari u obliku grafa

Centralno mesto u ovom grafu ima čvor sa oznakom “knjizara_tristero”, za koji smo rekli da je tipa “knjizara”, da mu je naziv Tristero i dali smo neke dodatne informacije o njemu. Ovakav graf za predstavljanje podataka mogao bi da se koristi u nekoj centralizovanoj aplikaciji, međutim, za web se ne može upotrebiti. Osnovni objekat u grafu je “knjizara_tristero”, međutim, niko ne garantuje da na webu ne postoji još neki objekat sa ovim nazivom (pri tom, to uopšte ne mora biti knjižara). Takođe, graf nam kaže da naša *knjizara_tristero* ima svojstva *tip*, *naziv*, *adresa* i *radno_vreme* i da ova svojstva imaju neke vrednosti. Međutim, ponovo imamo sličan problem, program koji bi pristupio ovim podacima ne bi mogao da zna o kojim je svojstvima zapravo reč, jer termine *tip*, *naziv*, *adresa* i *radno_vreme* različite organizacije na webu mogu da koriste za označavanje potpuno različitih pojmova. Potrebno je pronaći način da se u semantičkom webu izbegne upotreba homonima. Srećom, na internetu već postoji mehanizam koji omogućava jedinstveno imenovanje objekata i svojstava. To je *Uniform Resource Identifier* (URI).

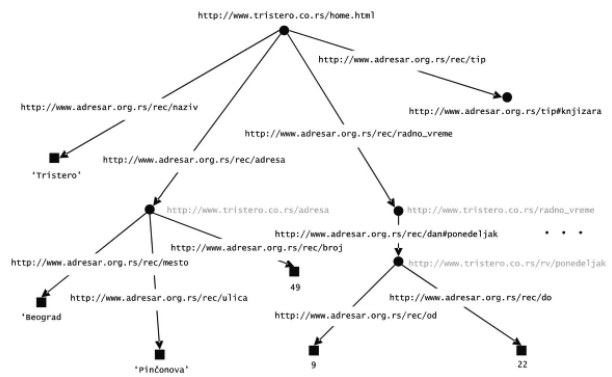
Kao što mu ime govori, URI se bavi jedinstvenim imenovanjem resursa. Pod resursom se ovde podrazumeva bilo koji objekat koji može, mada ne mora, biti povezan sa nečim što postoji na internetu. Adrese dokumenata na webu (URL) su URI sa kojima se najčešće susrećemo. Ovde se nećemo baviti opštom sintaksom URI-a, međutim, za nas je bitno to da u njemu na neki način mora postojati jedinstvena oznaka autoriteta koji je zadužen za dato ime, odnosno organizacije koja je zadužena da se ovo ime koristi samo u jednom, tačno određenom značenju. Mi ćemo koristiti URI u obliku URL-a koji sadrži domen čiji je vlasnik autoritet zadužen za dato ime. U našem primeru, recimo da imamo dva autoriteta koja daju imena resursima. To bi mogla biti sama knjižara (npr. da poseduje internet domen *tristero.co.rs*) i neka hipotetička

organizacija koja u Srbiji unifikuje informacije o preduzećima (domen *adresar.org.rs*).

S obzirom na to da je to u njenoj nadležnosti, knjižara za identifikator centralnog objekta u grafu, koji je predstavlja, može da izabere i adresu početne strane svoje web prezentacije (npr. <http://www.tristero.co.rs/home.html>). Međutim, ona to ne mora da uradi, jedino je važno da URI počinje sa *http*: (što u ovom slučaju nije oznaka protokola nego indikacija da je URI u obliku web adrese) i da u adresnom delu koristi domen čiji je vlasnik. Podrazumeva se da ovaj URI knjižara neće koristiti za označavanje drugih resursa.

Organizacija, koja je zadužena za unifikaciju podataka o preduzećima, može URI u svojoj nadležnosti evidentirati tako da počinje sa <http://www.adresar.org.rs>.

Zamenom proizvoljno datih imena URI-ima u grafu sa slike 4 dobijamo RDF graf (slika 5).



Slika 5: Osnovni podaci o knjižari u obliku RDF grafa

Primećujemo da su oznake svih grana i većine čvorova u grafu nazivi resursa (URI). Međutim, postoje neki čvorovi čije su oznake date kao neposredne vrednosti – nizovi znakova ili brojevi (mogu biti i datumi). Ove čvorove smo nacrtali kao kvadrate,⁵ a za njihove oznake se kaže da su date u obliku neposredno datih vrednosti, odnosno da su *literal*i. Literali ne mogu imati svojstva – iz njih ne mogu izvirati strelice.

Pri opisu tipa prodavnice koristili smo oznaku “<http://www.adresar.org.rs/tip#knjizara>” koja se sastoji iz URI-a i nekog dodatnog identifikatora međusobno razdvojenih znakom “#”. Ovaj dodatni identifikator naziva se *fragment*. Slično je i sa danima u nedelji čije smo oznake formirali od URI-a “<http://www.adresar.org.rs/rec/dan>” i fragmenta “*ponedeljak*”, “*utorak*”, ... Ovaj generalizovani oblik URI-a naziva se URI referenca ili skraćeno URIref.⁶

Oznake nekih čvorova (“<http://www.tristero.co.rs/adresa>”, “http://www.tristero.co.rs/radno_vreme” i niza

“<http://www.tristero.co.rs/rv/ponedeljak>” ...)”⁷ ispisali smo sivom bojom. Oznake ovih čvorova ne nose nikakvu korisnu informaciju. Čvor koji smo označili sa “<http://www.tristero.co.rs/adresa>” služi samo da bi objedinio informacije o mestu, ulici i broju. Njegova oznaka nam ne govori ništa, čak ni to da je poznata adresa knjižare – ova informacija je sadržana u tome što postoji svojstvo “<http://www.adresar.org.rs/rec/adresa>” iz čvora koji predstavlja knjižaru.⁸ Ovakvi čvorovi se ne označavaju u RDF grafu i nazivaju se neimenovani čvorovi (*blank nodes*).

Za razliku od XML-a koji je jezik za predstavljanje podataka, RDF ne govori ništa o obliku podataka (jeziku) u kome će se oni čuvati i razmenjivati, on se bavi isključivo modelom podataka. RDF graf je apstraktna sintaksa koja nije praktična za direktnu primenu. Ovaj graf se može transformisati na više načina u tekstualni oblik, odnosno postoji više RDF sintaksi. Najvažniji od njih su:

- Sintaksa uređenih trojki. Ovo je najjednostavniji način za tekstualno predstavljanje RDF podataka, koji leži u osnovi druga dva RDF jezika.
- RDF/XML je XML sintaksa za RDF podatke. Već je rečeno da se predviđa da će XML biti osnova semantičkog weba. Videli smo da XML dozvoljava da se na mnogo načina, korišćenjem različitih XML shema, prikažu informacije. RDF/XML definiše stroga pravila (shemu) predstavljanja podataka iz RDF grafa.
- N3 je proširenje sintakse uređenih trojki tako da bude pogodnija za praktičnu primenu. Pri tom, RDF/XML je pogodniji za programsku obradu, a N3 je prilagođen čoveku.

Ovde ćemo objasniti osnovne elemente sintakse uređenih trojki. RDF graf ovde je prestavljen kao skup uređenih trojki, pri čemu svakoj grani odgovara jedna uređena trojka koju čine oznaka izvorišnog čvora (resurs), oznaka grane (svojstvo) i oznaka čvora odredišta (vrednost svojstva). Na slici 6 prikazan je deo RDF grafa iz našeg primera sa slike 5 u sintaksi uređenih trojki.

```
<http://www.tristero.co.rs/home.html> <http://www.adresar.org.rs/rec/tip> <http://www.adresar.co.rs/tip#knjizara> .
<http://www.tristero.co.rs/home.html> <http://www.adresar.org.rs/rec/naziv> "Tristero" .
<http://www.tristero.co.rs/home.html> <http://www.adresar.org.rs/rec/adresa> ??? .
??? <http://www.adresar.org.rs/rec/mesto> "Beograd" .
??? <http://www.adresar.org.rs/rec/ulica> "Pinčonova" .
??? <http://www.adresar.org.rs/rec/broj> 49 .
```

Slika 6: RDF sintaksa trojki

Sintaksa uređenih trojki je jednostavna, ali zahteva neka objašnjenja. Umesto u matematici uobičajenog označavanja uređene trojke sa “(a, b, c)”, koristi se jednostavno navođenje elemenata bez zagrada i zareza kao delimitera, pri čemu se na kraju trojke obavezno stavlja prazno mesto i tačka. URI reference se stavljaju u izlomljene zagrade (“<”, “>”). Srednji element trojke, koji predstavlja granu grafa, često se izražava glagolskom frazom te se elementi trojke nazivaju i *subjekat*, *predikat* i *objekat*.

Iz primera sa slike 6 vidimo da upotreba punih URI referenci u sintaksi uređenih trojki dovodi do veoma dugačkih linija. Zbog toga je u svim RDF jezicima omogućeno skraćivanje URI izraza. Umesto dela URI-a “<http://www.tristero.co.rs/>”, možemo da koristimo, npr. prefiks “*tristero:*”, a umesto “<http://www.adresar.org.rs/rec/>” prefiks “*recnik:*”. Ovi delovi URI-a koje zamenjujemo prefiksom, nazivaju se prostori imena (*namespaces*). Kada se koriste prefiksi, URI reference se ne pišu u izlomljenim zgradama.

Prefikse možemo dodeljivati proizvoljno, mada se za obeležavanje nekih veoma značajnih i opšte poznatih prostora imena uobičajeno koriste uvek isti prefiksi. Tako se, na primer:

- prefiks “*rdf:*” koristi za prostor imena “<http://www.w3.org/1999/02/22-rdf-syntax-ns#>”,
- prefiks “*rdfs:*” koristi za “<http://www.w3.org/2000/01/rdf-schema#>”,
- prefiks “*owl:*” koristi za “<http://www.w3.org/2002/07/owl#>”,
- prefiks “*dc:*” koristi za “<http://purl.org/dc/elements/1.1/>”,
- prefiks “*ex:*” koristi za “<http://www.example.org/>” ili “<http://www.example.com/>”.

O prva tri prostora imena biće više reči kasnije. Četvrti (koji se obeležava sa “*dc:*”) je prostor imena za elemente Dablinskog jezgra (*Dublin Core*). Dablinsko jezgro je skup elemenata (metapodataka) koji služe za opis dokumenata na internetu, i to na način koji će omogućiti njihovo jednostavno i efikasno indeksiranje i pretraživanje. Poslednji prefiks (“*ex:*”) koristi se u različitim primerima, kada je potrebno da imamo pravilno formiran URI, ali nam nije bitno šta on predstavlja.

Neimenovani čvor, koji predstavlja adresu knjižare, označili smo sa “???”. Ako u grafu postoji više neimenovanih čvorova, ovakav način obeležavanja dovodi do kolizije imena. To se rešava tako što se neimenovanim čvorovima dodeljuje proizvoljna oznaka u prostoru imena sa prefiksom “*_:*”. Ovaj prostor imena rezervisan je za neimenovane čvorove. Na slici 7 prikazana je modifikovana RDF sintaksa.

```
tristero:home.html recnik:tip <http://www.adresar.co.rs/tip#knjizara> .
tristero:home.html recnik:naziv "Tristero" .
tristero:home.html recnik:adresa _:adresa .
_:adresa recnik:mesto "Beograd" .
_:adresa recnik:ulica "Pinčonova" .
_:adresa recnik:broj 49 .
```

Slika 7: Modifikovana RDF sintaksa trojki

RDF REČNICI I JEZIK ZA DEFINISANJE REČNIKA – RDF SCHEMA

RDF nam je omogućio da na jedinstven i jednostavan način predstavimo podatke o našoj knjižari. Razmotrimo šta bi hipotetički servis za pretraživanje semantičkog weba trebalo da radi kada mu postavimo zahtev da pronade knjižare u Beogradu koje rade nedeljom po podne. Izvršavanje ovog upita moglo bi da se odvija u sledećim koracima:

1. U svim RDF grafovima izdvojimo one koji imaju neki čvor (nazovimo ga, na primer, $?x$) koji je povezan sa čvorom “<http://www.adresar.co.rs/tip/knjizara>” granom “<http://www.adresar.org.rs/rec/tip>”. Na ovaj način izdvojili smo sve knjižare.
2. Suzimo rezultat pretraživanja iz koraka 1 na one grafove koji imaju vezu od čvora $?x$ (izdvojenog u prvom koraku) do nekog čvora granom “<http://www.adresar.org.rs/rec/adresa>” i od tog čvora do čvora označenog literalom “Beograd” granom “<http://www.adresar.org.rs/rec/mesto>”. Izdvojili smo sve knjižare u Beogradu.
3. Suzimo rezultat pretraživanja iz koraka 2 na one grafove koji imaju put od čvora $?x$ do nekog čvora $?y$ granama “http://www.adresar.org.rs/rec/radno_vreme” i “http://www.adresar.org.rs/rec/dani_nedelja” (čvor preko koga ovaj put prelazi nas ne interesuje). Na ovaj način izdvojili smo sve knjižare u Beogradu koje imaju podatak o nedeljnom radnom vremenu.
4. Ako pretpostavimo da frazu “rade po podne” naš pretraživač tumači kao “rade između 14 i 18 časova” u poslednjem koraku suzićemo rezultat pretraživanja iz koraka 3. na one grafove koji imaju vezu između čvora $?y$ i nekog čvora označenog literalom manjim ili jednakim 14, granom “<http://www.adresar.org.rs/rec/od>” i istovremeno ima vezu sa čvorom označenim literalom koji je veći ili jednak 18, granom “<http://www.adresar.org.rs/rec/do>”.

Dobijeni grafovi, odnosno njihovi čvorovi $?x$ predstavljaju tražene knjižare. Jasno je da pretraživač obilazeći RDF grafove, korisniku može jednostavno da izdvoji nazive i adrese knjižara koje zadovoljavaju postavljene uslove.

U analizi načina na koji radi naš hipotetički pretraživač, možemo da primetimo da je on morao da poznaje značenje određenih termina (URI referenci) koji se pojavljuju u RDF grafovima. Recimo, morao je da zna da se “<http://www.adresar.org.rs/rec/adresa>” koristi za obeležavanje grane koja povezuje čvor koji predstavlja preduzeće sa čvorom koji predstavlja njegovu adresu. Pri tom, pretraživač treba da prepozna samo termine za koje je odgovorna organizacija koja održava registar preduzeća (URI sa domenom “www.adresar.org.rs”). Knjižare proizvoljno mogu da biraju termine za označavanje ostalih čvorova i grana.

Skup termina (URI referenci), koji se koristi u nekom kontekstu, naziva se RDF rečnik.

RDF rečnike ponekad vezujemo za prostore imena. S obzirom na to da je u RDF grafu veoma jednostavno koristiti termine iz različitih prostora imena, rečnici se jednostavno mogu proširiti dodavanjem termina iz drugih prostora imena. U našem primeru, rečnik koji koristi hipotetički pretraživač sastoji se od termina iz “<http://www.adresar.org.rs/rec/...>” (prefiks “*recnik*”) i “<http://www.adresar.org.rs/tip/...>” (ovim URI referencama za sada nismo dodelili prefiks). Pored ovih, knjižara koristi i URI reference iz prostora imena “<http://www.tristero.co.rs/>” (prefiks “*tristero*”), ali može koristiti i termine iz drugih rečnika, na primer, katalog knjiga kojima raspolaze može se opisati terminima nekog rečnika za bibliografske podatke i rečnika koji se koristi za podatke o cenama.

Pri analizi rada našeg hipotetičkog pretraživača, zaobišli smo pitanje neimenovanih čvorova, pretražujući RDF graf po putanjama u kojima nije bilo bitno kroz koje se čvorove prolazi.

Zamislimo na trenutak da smo, umesto neimenovanog čvora koji predstavlja adresu knjižare, greškom stavili identifikaciju vlasnika knjižare. Ako bi ovaj pogrešno postavljeni čvor imao svojstvo “*recnik:mesto*” sa vrednošću “Beograd”, naš algoritam ne bi mogao da uoči grešku i smatrao bi da dati objekat zadovoljava postavljene uslove. Očigledno je da nije dovoljno da rečnik bude samo skup termina (URI referenci) koji mogu da se koriste, već da bi trebalo da sadrži mehanizam kojim se opisuje u kom kontekstu se neki termin može koristiti. U našem primeru, ovaj mehanizam bi trebalo da zabrani da se URI, koji označava vlasnika, koristi kao vrednost svojstva “*recnik:adresa*”.

Već na nivou RDF-a postoji jedan sintaksički konstrukt koji predstavlja podršku za definisanje rečnika.⁹ Naime, čvorovima RDF grafa može se dodeliti tip, odnosno može se reći da je tip nekog čvora drugi čvor. Sledeća RDF trojka kazuje da je *ex:T* tip čvora *ex:A*.

```
ex:A rdf:type ex:T .
```

Kada smo, u našem primeru, definisali da je “*tristero:home.html*” knjižara, koristili smo svojstvo “*recnik:tip*”. Umesto toga, mogli smo da koristimo “*rdf:type*” i da na taj način izbegnemo nepotrebno kreiranje sinonima za termine koji nam već stoje na raspolaganju. Videćemo da bi nam ovo donelo dodatne pogodnosti.

RDF Schema (RDFS) je osnovni mehanizam za definisanje RDF rečnika. RDFS nam omogućava da u RDF-u kreiramo hijerarhiju klasa i da definišemo svojstva.

Klase nam služe da grupišemo slične objekte. Tako, klasa “*http://www.adresar.co.rs/tip#knjizara*” grupiše sve objekte koji su knjižare. Međutim, u jeziku hipotetičke organizacije (“*http://www.adresar.org.rs*”), koja unifikuje adresar naših preduzeća, pored knjižara postoje i drugi tipovi prodavnica, a pored prodavnica i drugi tipovi preduzeća. Pogledajmo kako bi mogao da izgleda deo opisa jezika ove organizacije u RDFS.

U cilju jednostavnosti, uvedimo još jedan prefiks “*tip:*” za prostor imena “*http://www.adresar.co.rs/tip#*”. Najpre moramo kazati da su preduzeće, prodavnica, knjižara i, recimo, banka, klase. To radimo tako što za tip ovih objekata navodimo “*rdfs:Class*”:

```
tip:preduzece rdf:type rdfs:Class .
tip:banka rdf:type rdfs:Class .
tip:prodavnica rdf:type rdfs:Class .
tip:knjizara rdf:type rdfs:Class .
```

Svojstvo “*rdfs:subClassOf*” omogućava nam da definišemo hijerarhiju klasa:

```
tip:banka rdfs:subClassOf
tip:preduzece .

tip:prodavnicardfs:subClassOf
tip:preduzece .

tip:knjizarardfs:subClassOf
tip:prodavnica .
```

Kada u RDF-u napišemo:

```
tristero:home.htmlrdf:type
tip:knjizara .
```

pored toga što time označavamo da je “*tristero:home.html*” tipa “*tip:knjizara*”, ovo povlači sa sobom da je ovaj objekat istovremeno i tipa “*tip:prodavnica*” i “*tip:preduzece*”¹⁰. Naš pretraživač na pitanje “koje prodavnice u Beogradu rade nedeljom?”, sada može da ponudi odgovor koji uključuje i našu knjižaru, mada u njenom RDF opisu nigde nismo eksplicitno naveli da je reč o prodavnici. Na neki način, naš pretraživač postaje “svestan” značenja termina koje koristimo.

Kao što nam “*rdfs:Class*” i “*rdfs:subClassOf*” omogućavaju da definišemo tipove čvorova RDF grafa, “*rdfs:Property*” i “*rdfs:subPropertyOf*” omogućavaju da definišemo hijerarhiju tipova njegovih grana, odnosno svojstva. Kada je reč o granama, zanimljiviji je mehanizam za definisanje tipova čvorova između kojih grana može da se

javi. Pogledajmo sledeći fragment RDFS definicije:

```
ex:Knjiga rdf:type rdfs:Class .

ex:Izdavac rdf:type rdfs:Class .

ex:izdao rdf:type rdfs:Property .

ex:izdao rdfs:domain ex:Knjiga .

ex:izdao rdfs:range ex:Izdavac .
```

Definisali smo dve klase (“*ex:Knjiga*” i “*ex:Izdavac*”) i jedno svojstvo (“*ex:izdao*”). Zatim smo naveli kontekst u kome ovo svojstvo može da se javlja. Svojstvo “*ex:izdao*” može da se javi u izrazima oblika:

```
ex:A ex:izdao ex:B
```

gde je objekat “*ex:A*” tipa “*ex:Knjiga*”, a objekat “*ex:B*” tipa “*ex:Izdavac*”. Drugim rečima, u RDF grafu, ovako označena grana može da povezuje čvor koji predstavlja neku konkretnu knjigu sa čvorom koji predstavlja nekog konkretnog izdavača.

Ako je vrednost nekog svojstva literal, RDFS omogućava da definišemo tip literala (recimo ceo broj, niz znakova, datum ...).¹¹

ONTOLOGIJE I OWL (WEB ONTOLOGY LANGUAGE)

Videli smo da nam je RDFS omogućila da definišemo rečnik opisujući svojstva i klase RDF resursa. Pri tom smo se koristili samo jednim tipom relacija – generalizacijom. Semantički web zahteva da u mašinski čitljivom obliku predstavimo znatno potpuniju sliku sveta kojim se bavimo. Recimo, u našem primeru treba omogućiti da se može reći da objekat koji opisujemo mora da ima jednoznačno definisan tip (ne može biti istovremeno i knjižara i banka) i da mora imati najviše jednu adresu, ali da ta adresa ne mora biti specifikovana. Ovakav semantički bogatiji opis oblasti kojom se bavimo naziva se *ontologija*. Možemo reći da ontologiju čine rečnik i određena pravila te ograničenja u upotrebi termina ovog rečnika.

U semantičkom webu, ontologije definišemo u jeziku koji se zove *Ontology Definition Language* (OWL).¹² Slično kao RDFS i OWL se zasniva na RDF sintaksi. Ovde se nećemo detaljno baviti osobinama ovog jezika, već ćemo, kao ilustraciju, navesti neke njegove mogućnosti. U OWL-u možemo:

- definisati kardinalnost (minimalnu i maksimalnu) nekog svojstva u odnosu na neku klasu, odnosno mo-

žemo specificirati koliko puta se dato svojstvo može javiti u instanci date klase;¹³

- definisati sinonime za objekte, klase i svojstva;
- za svojstva definisati njihove karakteristike, na primer, tranzitivnost, simetričnost, inverzno svojstvo i slično;
- definisati ograničenja na vrednosti koje neko svojstvo može da uzima;
- definisati nove klase primenom skupovnih operacija nad postojećim klasama.

Ovde smo prikazali samo neke mogućnosti OWL-a koje bi trebalo da pomognu da se shvati šta termin ontologija znači u kontekstu semantičkog weba.

DEDUKTIVNO ZAKLJUČIVANJE NA OSNOVU ONTOLOGIJE

Na osnovu ontologije, iz podataka koje imamo u semantičkom webu, dedukcijom se mogu izvoditi određeni zaključci. Recimo da imamo ontologiju (sa prostorom imena *sr:*) u kojoj je definisana tranzitivna i simetrična relacija *sr:uSrodstvuSa*. Ako imamo sledeće činjenice:

```
os:Petar sr:uSrodstvuSa os:Ana.
os:Ana sr:uSrodstvuSa os:Vesna.
os:Dragan sr:uSrodstvuSa os:Vera.
```

Na upit:¹⁴

```
os:Vesna sr:uSrodstvuSa ?.
```

sistem će, na osnovu svojstava relacije *sr:uSrodstvuSa*, dati ispravan odgovor *{os:Ana, os:Petar}*. Ako bi data relacija bila samo simetrična, a ne i tranzitivna, rezultat bi bio *{os:Ana}*. Ako u ontologiji ne bi bilo specificovano nijedno svojstvo relacije *sr:uSrodstvuSa* (ili ako bi bilo specificovano da je relacija samo tranzitivna), rezultat bi bio prazan skup.

Vidimo da ontologija principijelno omogućava da sistem, na osnovu nekih činjenica koje su eksplicitno date, generiše nove činjenice. Ovo podseća na sisteme veštačke inteligencije i ponekad se koristi u kritikama protiv semantičkog weba sa argumentom da, poput oblasti veštačke inteligencije koja nije dala očekivane rezultate, to ne može učiniti ni semantički web. Ova kritika nije opravdana iz sledećih razloga:

1. Cilj veštačke inteligencije je oponašanje inteligentnog ponašanja čoveka, a cilj semantičkog weba je da omogući mašinsku obradu podataka koji se generišu i nalaze na različitim mestima.
2. Većina aplikacija na semantičkom webu ne koristi nikakve sofisticirane obrade na osnovu ontologija, za

njih su ontologije samo sredstvo za specificaciju podataka. Možemo reći da glavni korisnici ontologija nisu programi, već programeri.

3. Za razliku od veštačke inteligencije, semantički web koristi jednostavnije i teorijski dobro proučene mehanizme zaključivanja. Ovo se posebno odnosi na OWL. Složenije metode zaključivanja su takođe predmet interesovanja, ali su one u arhitekturi semantičkog weba (slika 1) izdvojene na poseban nivo.

OSTALE TEHNOLOGIJE SEMANTIČKOG WEBA

Spisak tehnologija semantičkog weba ovim nije ni približno iscrpljen. Mislimo da bi trebalo pomenuti bar još tri tehnologije (ili grupe tehnologija) koje igraju veliku ulogu u ovoj oblasti:

- SPARQL upitni jezik za RDF. SPARQL omogućava da na jedinstven način postavljamo upit u različite RDF podatke i ujedno definiše oblik rezultata ovih upita. SPARQL bi, u primeru s početka članka, trebalo da omogući da, uz informaciju o adresi i radnom vremenu knjižara, pretražujemo i njihov katalog knjiga. Poseban značaj SPARQL je u tome što dozvoljava da se podaci, koji nisu u obliku RDF-a, uključe u semantički web. Implementacijom posebne softverske komponente, adaptera, SPARQL upiti mogu se izvršavati u proizvoljnom izvoru podataka. U ovom slučaju, adapter je odgovoran i za prevođenje dobijenih rezultata u propisani oblik. To znači da se različite baze podataka (na primer, relacione i bibliografske) mogu uključiti u semantički web, a da pri tom nije potrebno konvertovati njihove podatke.
- Mehanizmi logičkog zaključivanja. Pored ontologije, za neke primene, mogu se definisati i složenija pravila zaključivanja koja omogućavaju da se iz neposredno datih podataka na semantičkom webu izvedu nove informacije. Za ovo se koriste različite metode koje su razvijene u oblasti veštačke inteligencije, kao što su ekspertski sistemi i logičko programiranje.
- Mehanizmi za obezbeđivanje poverenja. Svi smo svesni složenosti problema koji se tiču zaštite podataka na postojećem webu. Ti su problemi na semantičkom webu još složeniji, jer ne mora da postoji korisnik (čovek) koji će, zahvaljujući svom iskustvu, odrediti da li i koliko se može imati poverenja u neki izvor informacija ili servis. Kritičari semantičkog weba posebno ističu problem mogućeg narušavanja privatnosti, i to upravo zbog njegove sposobnosti da integriše podatke iz različitih izvora. Pri tom, naše kupovne navike predstavljaju verovatno najmanji problem.

ZAKLJUČAK

U ovom trenutku ne znamo kakva je zapravo budućnost semantičkog weba. Veliki broj kompanija i institucija ulaže velike napore da se obezbedi softverska infrastruktura potrebna za razvoj i funkcionisanje aplikacija i servisa zasnovanih na semantičkom webu. Međutim, za njegov uspeh ili neuspeh, od presudnog uticaja neće biti softver, već korisnici. Pitanje je kad će (skeptici kažu i da li će) semantički web dostići broj korisnika koji će motivisati nove korisnike da počnu da ga koriste. Optimističke prognoze govore da će semantički web za 5 do 7 godina izaći iz eksperimentalne faze i početi praktično da se primenjuje, a da će između 2018. i 2020. godine doživeti punu ekspanziju. Međutim, već je sada razvoj semantičkog weba dao podsticaj mnogim drugim disciplinama softverskog inženjeringa, a posebno onima koji se tiču konceptualnog oblikovanja i predstavljanja znanja. Budućnost će pokazati kakav će uticaj semantički web imati na društvo i da li će se ostvariti vizije Tima Berners-Lija (Tim Berners-Lee) i drugih zagovornika ove ideje.

Opombe

- 1 U ovom članku se nećemo baviti razlikom između pojmova podatak, informacija i znanje. Ova razlika igra određenu ulogu u teorijskim radovima vezanim za semantički web, međutim, nama to ovde nije interesantno.
- 2 Mikroformat je metod označavanja postojećeg sadržaja HTML/XHTML stranica (npr. preko atributa class, ili nekog drugog atributa koji korisnik definiše), tako da je omogućena njegova automatska obrada.
- 3 Ovo ne znači da će XML biti jedini format koji će se koristiti. Videćemo kasnije da RDF ne zavisi od XML-a, već da se informacije mogu predstaviti i na drugim jezicima. Takođe, na webu jedan drugi format za razmenu podataka, JSON, postaje sve popularniji. Ipak, kako sad izgleda, XML će imati dominantnu ulogu.
- 4 Graf je matematička struktura koja se sastoji od skupa čvorova i skupa veza između ovih čvorova. Mi ovde koristimo označene, usmerene grafove, što znači da čvorovi i veze imaju svoje nazive (oznake) i da je za vezu bitno koji je čvor prvi, a koji drugi. U grafičkoj predstavi ovakvih grafova, veze imaju naziv i strelicu koja pokazuje na drugi čvor.
- 5 Ovo nije standardna oznaka u RDF grafovima, uveli smo je da bismo graf učinili čitljivijim.
- 6 S obzirom na to da URI reference obuhvataju i "obične" URI-e, u daljem tekstu biće uglavnom korišćen termin URI referenca.
- 7 Na slikama 4 i 5, u cilju uštede prostora, prikazani su podaci o radnom vremenu samo za ponedeljak; jasno je kako izgleda izostavljeni deo grafa sa podacima o ostalim danima.
- 8 Ovde je važno primetiti da je oznaka grane u nadležnosti organizacije koja se bavi standardizacijom adresara (adresar.org.rs), dok je oznaka čvora privatna stvar knjižare (tristero.co.rs).

- 9 RDF poseduje i neke druge mehanizme (na primer, za definisanje nekih struktura podataka – kontejnera i kolekcija), ali to su tehnički detalji kojima se ovde nećemo baviti.
- 10 Drugim rečima – knjižara je užu pojam od pojma prodavnica, prodavnica je užu pojam od preduzeća.
- 11 RDF Schema nema ugrađene tipove podataka, već se koriste tipovi iz nekog drugog rečnika, najčešće iz XSDL-a.
- 12 OWL ima više varijanti: OWL Lite, OWL DL, OWL Full. Ove varijante imaju različite mogućnosti u iskazivanju semantičkih relacija između termina rečnika.
- 13 Za objekat kažemo da je instanca neke klase, ako je njegov tip data klasa.
- 14 Ovde smo iskoristili sintaksu SPARQL upitnog jezika, ali smo prikazali samo WHERE deo kompletnog upita, koji je nešto kompleksniji.

Reference

- [1] Walton, Christopher D. (2007). *Agency and the Semantic Web*. New York, Oxford University Press.
- [2] McGuinness, Deborah; L., Frank van Harmelen [ur.] (2004). *OWL Web Ontology Language – Overview*; W3C Recommendation. Dostupno na: <http://www.w3.org/TR/owlfeatures/>.
- [3] Prud'hommeaux, Eric; Andy Seaborne [ur.] (2008). *SPARQL Query Language for RDF*; W3C Recommendation. Dostupno na: <http://www.w3.org/TR/rdf-sparql-query/>.
- [4] Manola, Frank; Eric Miller [ur.] (2004). *RDF Primer*; W3C Recommendation. Dostupno na: <http://www.w3.org/TR/rdfprimer/>.
- [5] Antoniou, Grigoris; Frank van Harmelen (2008). *A semantic Web primer*. 2nd ed. London, MIT Press.
- [6] Smith, Michael K. i dr. [ur.] (2004). *OWL Web Ontology Language – Guide*; W3C Recommendation. Dostupno na: <http://www.w3.org/TR/owl-guide/>.