Pretraživanje semantičkog web-a

Pranjić, Darko

Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
«Dr. Mijo Mirković»

DARKO PRANJIĆ

PRETRAŽIVANJE SEMANTIČKOG WEBA

Završni rad

Pula, 2015.

Sveučilište Jurja Dobrile u Puli
Fakultet ekonomije i turizma
«Dr. Mijo Mirković»

DARKO PRANJIĆ

PRETRAŽIVANJE SEMANTIČKOG WEBA

Završni rad

JMBAG: 0303033580, izvanredni student
Studijski smjer: Informatika

Predmet: Upravljački IS
Mentor: Prof. dr. sc. Vanja Bevanda

Pula, rujan 2015.

Sadržaj

1 UVOD

U ranim danima weba čuvši za njegove mogućnosti tj. povezivanja dokumenata poveznicama skeptici su se pitali tko će kreirati toliki sadržaj i hoće li to uopće funkcionirati u praksi. Čim je web infrastruktura „legla na svoje mjesto“ ljudi su počeli sami dijeliti dokumente bilo putem službenih ili neslužbenih web stranica o toj temi. U webu je svakome dozvoljeno reći bilo što o bilo kojoj temi (AAA – eng. *Anyone can say Anything about Any topic*) i tako ljudi mogu kreirati svoje stranice u kojima mogu izražavati mišljenje i napisati što god požele. Kako je trend rastao tako se sve više ljudi uključilo na strani kreatora stranica, ali i čitatelja njenog sadržaja.

Trenutni način web infrastrukture bazira se na distribuciji web stranica koje služe za prezentaciju sadržaja povezanim linkovima (URL[1]-eng. *Uniform Resource Locator*). Ideja semantičkog weba je podržati web na razini podataka a ne prezentacije te umjesto da se stranice povezuju jedna na drugu, podaci se mogu povezivati jedan na drugog koristeći globalne identifikatore (URI[2]-eng. *Uniform Resource Identifier*) . Model podataka kojim se semantički web služi za distribuciju podataka je RDF[3] (eng. *Resource Description Framework*). U semantičkom webu, podaci moraju biti označeni na način da se mogu kombinirati sa drugim podacima iz različitih vanjskih izvora i tako davati korisne informacije.

U početku semantičkog weba samo su kreatori ideje semantičkog weba i njima bliski imali interesa za njegovu tehnologiju. Kako je sve više podataka bilo dostupno u RDF obliku tako je počelo biti sve više interesa i počele su se pisati aplikacije koje koriste ove podatke. Neke od velikih izvora javnih podataka dostupnih u RDF-u može se pronaći na stranici Dbpedia koja koristi podatke s Wikipedije ili Freebase baza podataka koju koristi Google.

Cilj ovog rada je pružiti uvid u razlike između pretraživanja u semantičkom webu, kada su podaci povezani i pretraživanja po ključnoj riječi. Objašnjena je povijest semantičkog weba pa sve do opisnog dijela svakog pojedinog sloja na kojemu počiva njegova struktura. Za posljednja poglavlja su ostavljene semantičke tražilice i semantički preglednici, te načini na koje koriste semantički web.

[1] Adresa do sadržaja na Internetu
[2] Jedinstveni identifikatori različitih resursa
[3] Jezik koji služi za opisivanje resursa i daje mogućnost njegovog spajanja s drugim resursima

2 POVIJEST WEBA I UVOD U SEMANTIČKI WEB

2.1 Web 1.0 – Mreža informacija

Web 1.0 koji je nastao 1991. bio je u biti izvor informacija kreiran od malog broja autora za jako veliki broj korisnika. Sadržavao je statičke web stranice bez mogućnosti prave komunikacije između korisnika i može se reći da je funkcionirao kao knjižnica referentnih knjiga. Do pojave Web-a 2.0 nije se pričalo o verzijama weba tako da je izraz web 1.0 nastao tek nakon izlaska weba 2.0 kako bi se pomoglo razlikovati informacijsku od društvene mreže.

2.2 Web 2.0 – Društvena mreža

U sljedećih desetak godina situacija na Internetu se mijenjala pojavom većeg broja web stranica koje su davale mogućnost interakcije između korisnika na njima (blogovi, slanje web poruka – primjer Facebook, Twitter, razne društvene mreže). Kreator stranice više nije sam autor informacija na web stranici nego je ona otvorena i drugim korisnicima kako bi upravljali njenim sadržajem.

„Do naziva web 2.0 došlo se tijekom brainstorminga za naziv internet konferencije koja je bila usmjerena na najučinkovitije načine korištenja interneta gdje je napomenuto da je web važniji nego ikad, sa svim novim web stranicama i aplikacijama koje su iznenađujuće ispravno funkcionirale. Naziv konferencije postao je „Web 2.0“ a tijekom sljedećih godinu i pol dana se očito zadržao, citiran preko 9.5 milijuna puta na Google-u. „[4]

2.3 Web 3.0 – Semantička mreža

Jedan od problema weba 2.0 je kako pronaći korisnu informaciju na njemu. Rješenje koje je bilo ponuđeno su tražilice u koje upisujemo riječ i one nam vraćaju rezultat ukoliko se ta riječ spominje negdje u sadržaju web stranice. Ali web je ogroman i ponekad je teško pronaći nama

[4] http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html

potrebnu informaciju zbog rezultata pretrage, odnosno svih stranica koje sadrže tu riječ. Godine 2000-e je na 6,127,700,430 stanovnika bilo 413,425,190 povezanih korisnika na Internet[5] (6.7% populacije), godine 2014-e je na 7,243,784,121 ljudi u svijetu bilo 2,925,249,355 korisnika s pristupom Internetu što je 40.4% korisnika od ukupne ljudske populacije. Vidljivo je da je ogromna razlika u broju korisnika stvorena tijekom godina i normalno je da se i količina informacija na webu povećala sa povećanim brojem korisnika.

Web postaje veći svaku minutu, u tablici 1. je izvučeno par podataka istraživanja tvrtke DOMO. Istraživanje se baziralo na mjerenje količine podataka postavljenih na web u jednoj minuti, podaci su za 2012 i 2014 godinu.

Tablica 1. Broj podataka stavljenih na web u jednoj minuti za 2012 i 2014 godinu

Podaci tvrtke DOMO – generirani podaci u jednu minutu		
	2012	**2014**
Podignutih novih Youtube videa	48 sati	72 sata
Google pretraga	Preko 2,000,000	Preko 4,000,000
Poslanih e-mailova	204,166,667	200,000,000
Podijeljenih Instagram slika	3,600	216,000
Kreirane web stranice	571	/
Podijeljeni sadržaj Facebook korisnika	684,478	2,460,000

Izvor: http://www.visualnews.com/2012/06/19/how-much-data-created-every-minute

Trenutno je oko 3,100,000,000 ljudi spojeno na Internet i izmjenjuju informacije, ali za olakšanje ubrzanog života ljudi počeli su se na Internet spajati uređaji (IoT eng. *Internet of things*"). Pametne kuće, pametni satovi, hladnjaci, pametne utičnice su samo jedan mali dio IoT-a, koji spajanjem na web također šalju informacije i izmjenjuju poruke. Procjene su da će do 2020. godine na Internet biti spojeno između 20 milijardi[6] i 50 milijardi[7] uređaja, stoga možemo zamisliti do kuda će sezati broj objavljenih podataka.

Jedan od problema sa tolikom količinom podataka je kako izvući ono nama bitno. Uzmimo na primjer web stranicu na stranom jeziku koji ne razumijemo. Vidimo njen sadržaj, ali ne znamo

[5] http://www.internetlivestats.com/internet-users/#trend
[6] http://www.gartner.com/newsroom/id/2636073
[7] http://www.cisco.com/web/solutions/trends/iot/portfolio.html

što znači pojedini dio na stranici. Koji dio je reklama, koji je informacija, koji dio je suvišan. Ako je ipak riječ o informaciji dali nama nešto znači i dali joj se može vjerovati. Ljudima je lako riješiti predstavljeni problem tako da se nauči jezik, apsorbira određena informacija te ju kasnije prosljeđuju drugima. Ali računalo drugačije funkcionira, budući web stranice kodirane su HTML-om (eng. *HyperText Markup Language*). HTML je jezik koji ima ulogu poveznice i opisuje samo izgled web informacije ali ne opisuje što predstavljena informacija ustvari znači i kakvo značenje ima korisniku.

Bez tražilica bi se izgubili na webu, ali predstavljeni problem je pronalaženje informacija. Uzmimo za primjer da upitom želimo pronaći životinju „pile" i informacije o njoj. Upisom „pile" dolazimo do životinje „pile" ali i mnogo nama nebitnih informacija kao što su „pile – franc. baterija", „pile – eng. hrpa", „pile – Japanska pjevačica". Upisom sinonima od „pile" – „pilić", dobijemo druge rezultate koji su nama potrebni i do kojih nećemo doći upitom „pile". Pretraživač nam daje rezultate stranica gdje je tražena riječ prisutna ali ne i rezultate gdje je označen sinonim koji ima isto značenje te s toga ne dobivamo sve moguće odgovore. Tradicionalno pretraživanje po ključnoj riječi ponekad može dovesti do previše irelevantnih informacija.

Većina podataka trenutno je povezana poveznicom prema nekoj drugoj web stranici, stoga se predstavlja još jedan problem, a to je održavanje. Postavlja se pitanje što ako ta druga web stranica prema kojoj nas poveznica upućuje više ne postoji ili je sadržaj na njoj izmijenjen. Tu dolazimo do pojave semantičkog weba.

Za uspješnu verbalnu komunikaciju potrebno je da je informacija točno prenesena, ispravno interpretirana i da pošiljatelj i primatelj na isti način shvaćaju rečenicu. Na toj pretpostavci bazira se ideja semantičkog weba.

„Riječ semantika dolazi od grčke riječi *semantikos* koja se prevodi kao *onaj koji daje znakove, značajan, simptomatičan*, odnosno u korijenu svega je riječ *sema* što znači *znak, značenje.* Možemo reći da je semantika jedan od triju aspekata svakog smislenog pojma, dakle da uz sintaksu[8] i pragmatiku[9] daje smisao, odnosno značenje pojmu. Znamo da na primjer jedno te

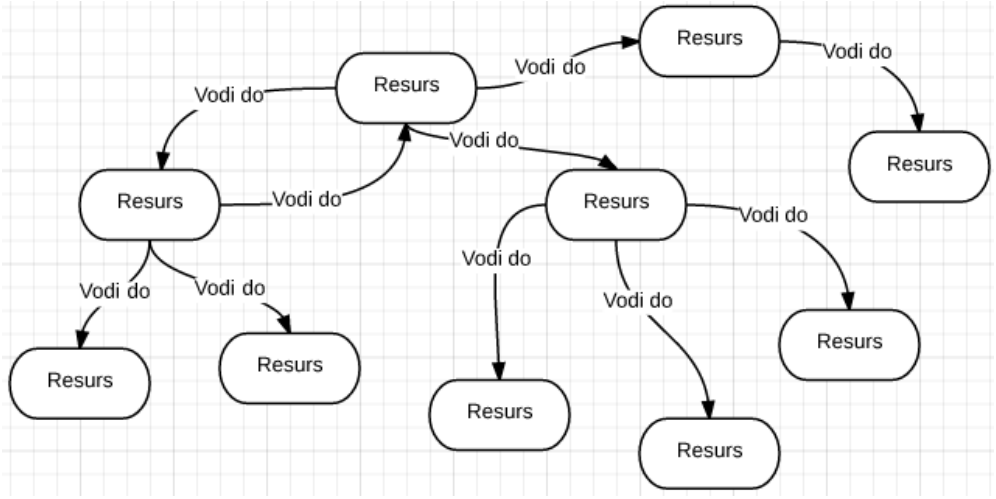
[8] Pravila pisanja i konstruiranja same riječi; dio gramatike koji proučava pravila ustrojstva i raščlambu rečenice, tj. poredak, razmještaj i međusobno prilagođivanje riječi i njihovih skupina u rečenici, njezinim dijelovima, rečeničnim sklopovima.

[9] Proučavanje znakova u situaciji; dio semiotike i općenito lingvistike i teorije komunikacije koji se bavi odnosima između znakova i njihovih tumača u odnosu na situaciju u kojoj se oni nalaze, na njihove potrebe, ciljeve i sl

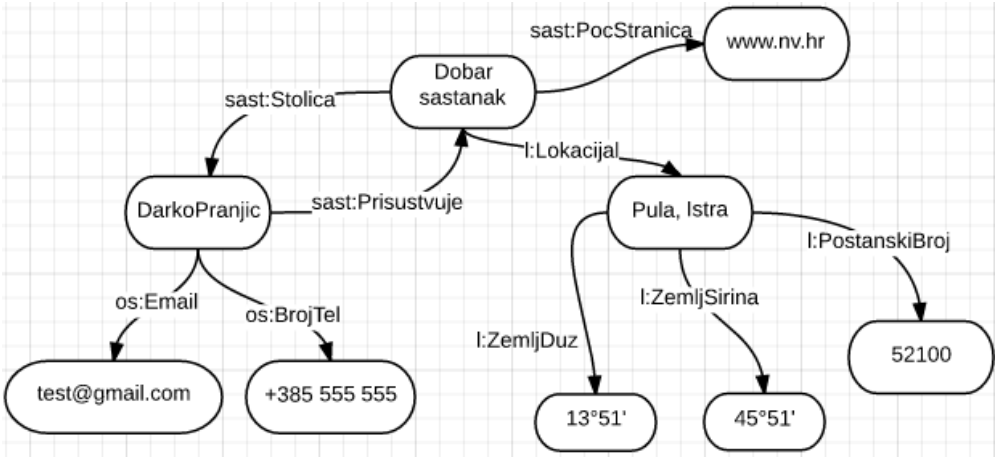
ista riječ može imati nekoliko značenja, što znači da je njezina semantika u tim slučajevima različita.“[10]

Promoviranje i razvoj semantičkog weba započelo je 1990-ih godina od strane W3C konzorcija, a izumitelj weba Tim Berners Lee počeo je 1998. godine promovirati semantički web. On definira semantički web kao „web podataka“ koji mogu biti direktno ili indirektno obrađeni od strane računala. On nije odvojeni web nego nadopuna na trenutni, omogućava računalima i ljudima da rade u suradnji te u kojem je informaciji dano jasno definirano značenje. Sve ovo je moguće jedino ako je sadržaj web stranice čitljiv i razumljiv računalu. U semantičkom webu sadržaj je označen semantičkim koji ga kodiraju na način da bude čitljiv i točno protumačen sa strane računala.

Slika 1. Trenutni način povezivanja podataka



Slika 2. Povezivanje podataka u semantičkom webu



Na slici 1. prikazan je još uvijek, na većini tražilica, način povezivanja podataka gdje je vidljivo da se podaci povezuju preko web stranica ne znajući što ih povezuje niti na što će se naići. Na

[10] http://autopoiesis.foi.hr/wiki.php?name=KM+-+Tim+22&parent=NULL&page=semantika

slici 2. prikazan je način povezivanja podataka na semantičkom webu. U primjeru vidimo resurs s imenom „Dobar sastanak“ koji ima računalu povezane i čitljive podatke označavajući ga jedinstvenim. Može postojati drugi sastanci sa tim imenom, ali biti će na drugoj lokaciji ili osoba „Darko Pranjic“ neće biti prisutna. Osoba „Darko Pranjic“ također je jedinstvena svojim podacima.

3 SLOJEVI SEMANTIČKOG WEBA

Prilikom prezentacije semantičkog weba Tim Berners Lee je predstavio i slojeve web standarda koji će se koristiti.

- Osnovni sloj obuhvaća Unicode i URI koji služe za korištenje internacionalnog skupa znakova i identifikaciju.
- Na drugom sloju se nalazi XML (eng. *Extensible Markup Language*) te omogućava korisnicima da izradu vlastitih oznaka kod kreiranja web dokumenata. XML ne daje nikakvu semantičku vrijednost XML dokumentu te daje mogućnost integracije dokumenta s raznim programima.
- RDF je jezik poznat računalu, povezuje podatke te daje mogućnost slaganja tvrdnji koje opisuju resurse i relacija između povezanih resursa. RDF Shema omogućuje definiranje hijerarhije između resursa i daje značenje vezama.
- Ontologija je nadogradnja na prethodni sloj. Resursi definirani RDF-om i RDF shemom se većim brojem veza između podataka detaljnije specificiraju, definiraju odnosi, ograničavaju te detaljnije opisuju njihova svojstva i klase
- SPARQL (eng. *SPARQL Protocol and RDF Query Language*) služi za postavljanje upita nad semantičkim bazama podataka te omogućuje spremanje, izmjene i izvlačenje podataka iz njih

3.1 URI

URI služi za identifikaciju različitih tipova resursa koji mogu biti bilo što sa definiranim identitetom. Resurs (eng. *resource*) može biti knjiga, lokacija, osoba, veza između objekata itd. URI se mora prilagoditi i razlikovati od resursa do resursa na način da tvori jedinstveno URI ime.

URI kombinira sa dva načina označavanja: adresom i identitetom. Adresa (URL) ili lokacija označava gdje resurs može biti pronađen na webu, ali problem je da se adresa može promijeniti pa veza na njega može postati veza na nešto nebitno. Također važno je napomenuti da URL može biti URI ali ne može biti suprotno. Dajući naziv (URN-eng. *Uniform Resource Name*) resursu on dobiva jedinstven identitet koji se ne mijenja. Ali za pronaći nešto moramo znati gdje se nalazi, kako je moguće da se promijenila lokacija mora se pitati nekoga gdje se nalazi i pronaći adresu. Primjer sa URL i URN načinom označavanja:

URL – http://oet.unipu.hr/ - web stranica fakulteta ekonomije i turizma

URN – urn: isbn:0451450523 – knjiga „Posljednji jednorog“ iz 1986. godine.

Predstavljena su dva pravila označavanja URI-jem:

1. URI može identificirati samo jedno, isti URI ne smije označavati stvarni objekt i web stranicu.
2. Ukoliko se zatraži URI, računalo treba razumjeti da se traži opis resursa te vratiti korisniku čitljivi prikaz.

Recimo da želimo znati kada je „Stonehenge“ izgrađen, logično bi bilo da se otiđe na URI koji ga označava http://www.stonehenge.co.uk/ i čiji bi sadržaj trebao imati korisne informacije. Ali pregledavajući sadržaj nigdje se ne nalazi ono što tražimo. Gledajući format strukturiranja stranice vidi se da meta oznake predstavljaju HTML stranicu, a pretraživač sukladno s tim prevodi HTML format i prezentira sadržaj da ga ljudi mogu iščitati. Kao sve na tradicionalnom webu navedena stranica i njoj srodne su web dokumenti.

Za pristup zatraženoj informaciji web pretraživač i serveri koriste HTTP[11] protokol (eng. *Hypertext Transfer Protocol*). Kada pretraživač da HTTP zahtjev serveru na adresu

[11] Protokol za prijenos datoteka na webu

„Stonehenge“-a tada označi i traži željenu datoteku a server odgovara da li je zahtjev uspješan i sadržaj se šalje klijentu. Primjer slanja:

```
GET: http://www.stonehenge.co.uk/index.php HTTP/1.1
HOST: www.stonehenge.co.uk
ACCEPT: text/html
```

Odgovor servera:

```
HTTP/1.1 200 OK
Content-Length: 3700
Content-Type: text/html..
```

Recimo da želimo vidjeti informacije o osobi Li Yang, znamo da web stranicom dobijemo i informacije koje nas ne zanimaju stoga se postavlja pitanje kako pronaći osobu a ne web stranicu.

Nude se dva rješenja identificiranja stvarnih objekata: URI „303 vidi drugo“ (eng. *303 See Other*) i URI označen znakom ljestvi (eng. *hash URI*).

3.1.1 Hash URI

URI može sadržavati fragmentaciju, dijelove koji su odvojeni od URI-ja znakom ljestvi. Potrebno je napomenuti da bilo koji URI koji sadržava fragmentaciju označava stvarni objekt, a ne stranicu te se tako izbjegava dvosmislenost.

„Ako zatražimo URI `http://www.liyangyu.com/foaf.rdf#liyang` dobijemo opis resursa Li Yang-a, tj. osobu koja je identificirana s njim, znamo njegov mail, što ga zanima, gdje radi. Kada se šalje zahtjev serveru URI-jem ljestvi HTTP protokol traži da se odstrani dio nakon ljestve prije slanja serveru i zahtjev izgleda ovako“[12]:

```
GET /foaf.rdf HTTP/1.1
Host: www.liyangyu.com
```

[12] Liyang Yu, A developer's guide to the semantic web, 2011. str. 419

3.1.2 303 See Other URI

Način na koji se odvija komunikacija „303 Vidi drugo“ između pretraživača i servera je pregovorom o sadržaju (eng. *content negotiation*). Potrebno je imati URI koji nema informacija na njemu, ali označava resurs. Prilikom slanja zahtjeva pretraživač uključuje taj URI u HTTP zaglavlje i navodi koji tip datoteke preferira za pregled sadržaja. Server pregledava zaglavlje, vidi koji resurs se traži i vraća HTTP kod „303 Vidi drugo“ te preusmjerava klijenta na dokument koji daje uvid u odgovarajući sadržaj. Ako pretraživač podržava RDF, server će najvjerojatnije preusmjeriti na RDF dokument. Ako pretraživač ne podržava RDF, nego samo HTML, tada ćemo biti preusmjereni na HTML dokument.

Primjer je „Stonehenge“ koji nas otvaranjem URI identifikatora „Stonehenge“-a (`http://dbpedia.org/resource/Stonehenge`) preusmjerava na URI (`http://dbpedia.org/page/Stonehenge`) što prikazuje sadržaj u formatu lako čitljivom ljudima. URI `http://dbpedia.org/data/Stonehenge` identificira prikaz u RDF/XML formatu ali može se otvoriti samo ako imamo pretraživač koji podržava prikaz RDF dokumenata.

3.2 XML

XML je jezik dizajniran za označavanje podataka. Uglavnom je obična tekstualna datoteka koju mogu čitati tekstualni i razni programski editori. Razlika između HTML i XML jezika je ta da se HTML jezik koristi elementima označavanja kako bi se dizajnirala web stranica i podaci u njoj.

Svrha XML-a je da elementima označava, prenosi i sprema podatke između aplikacije i baze podataka. Kada je opisana struktura i značenje podataka, zahvaljujući ovome formatu, omogućeno je i njegovo ponovno korištenje na različite načine.

Dok HTML ima određen broj elemenata koji se koriste, XML može imati beskonačan broj mogućih elemenata. Proširiv (eng. *extensible*) znači da imamo mogućnost sami stvarati oznake (eng. *tag*) kao što su <ime>, <prezime> ili <adresa>. Svi elementi imaju otvarajuću i zatvarajuću oznaku, dok u HTML-u postoje elementi koji ne moraju imati zatvarajuću oznaku,

npr.
. Sve oznake su osjetljive velikim ili malim slovom tako da oznaka <Ime> i <ime> predstavljaju različiti element. XML dokument sastoji se od Unicode znakova pa tako podržava sva svjetska slova pa i kineska ili hrvatska (č, ć, ž, đ, š).

Primjer kreiranja XML dokumenta:

```
<?xml version="1.0" encoding="UTF-8" ?>
<prijevozno_sredstvo>
      <auto type="poslovni">
            <marka>Zastava</marka>
            <model>750</model>
            <nadimak>Fićo</nadimak>
      </auto>
      <auto type="osobni">
            <marka>Hyundai</marka>
            <model>i30</model>
      </auto>
</prijevozno_sredstvo>
```

Ako dokument nije oblikovan po strogim pravilima od kojih se XML struktura sastoji, tada isti neće raditi. Prvo i početno pravilo je da dokument mora imati XML deklaraciju koja označava XML verziju dokumenta. Zatim svaki element mora imati korijenski element, u našem primjeru je to `<prijevozno_sredstvo>`. Ostali elementi moraju biti u odnosu roditelj – dijete (eng. *parent-child*) <auto> i <marka> ili braća i sestre (eng. *siblings*) <marka> i <model>. Također je moguće se dokument sadrži i atribute koji se označavaju sa „type“ čija je namjena da pruže dodatne informacije o elementu.

3.2.1 DTD

DTD (eng. *Document Type Description*) predstavlja pravila koja se moraju pridržavati prilikom kreiranja XML dokumenta. Opcionalno je da li će se koristiti, no ako se koristi tada može biti izražen u XML dokumentu ili pozvan kao vanjska datoteka. Određuje kakva vrsta sadržaja je dopuštena i gdje se sadržaj smije pojaviti.

```
<!DOCTYPE prijevozno_sredstvo [
<!ENTITY tekst „Samo najbolji auto">
<!ELEMENT prijevozno_ sredstvo(auto)*>
<!ELEMENT auto (marka, model, nadimak)>
```

```
<!ELEMENT marka (#PCDATA)>
<!ELEMENT model (#PCDATA)>
<!ELEMENT nadimak (#PCDATA)>
]>
```

Pravila ovog DTD-a određuju da korijen XML dokumenta treba započeti sa elementom <prijevozno_sredstvo> unutar kojega može biti jedan ili više elemenata <auto> čiji pod elementi moraju biti <marka>, <model>, <nadimak>, a njihov sadržaj može biti bilo koji tekst. Vrijednost entiteta imena tekst je „Samo najbolji auto“ te ukoliko želimo tu vrijednost negdje upisati možemo samo pozvati entitet sa „&tekst“. DTD je lako za napisati, ali nije baš fleksibilan i moćan, npr. ne može se limitirati sadržaj elementa na telefonski broj ili brojeve.

3.2.2 XML Shema

Kao DTD tako i XML Schema predstavlja pravila kreiranja XML dokumenta, samo što je njegov nasljednik i mnogo je moćnija od DTD-a. Jedna od najvećih snaga joj je da pruža podršku za vrste podataka stoga je lakše:

- provjeriti ispravnost podataka
- raditi sa podacima iz baze podataka
- lakše je definirati ograničenja na podacima
- definira format podataka

XML Schema osigurava komunikaciju sa podacima pa slanjem podatka od pošiljatelja do primatelja obje strane dobiju istu vrijednost sadržaja. U nekim zemljama datum 12-04-1988 znači 4. prosinac.1988, a taj problem se uz pomoć sheme rješava da se označi element sa tipom „date“ koji zahtjeva format „YYYY-MM-DD“. Sljedećim primjerom je XML dokument iz prijašnjeg primjera nadovezana s XML shemom, prikazan je samo osnovni koncept i opisan ispod koda:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema>
<xsd:element name="prijevozno_sredstvo"/>
      <xs:complexType>
            <xs:sequence>
                  <xs:element name="auto" type="autoType"
```

```
                minOccurs="0" maxOccurs="unbounded" />
            </xs:sequence>
        </xs:complexType>
        <xs:complexType>
            <xs:sequence>
                <xsd:element name="marka" type="xs:string"/>
                <xsd:element name="model" type="xs:string"/>
                <xsd:element name="nadimak" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
</xs:schema>
```

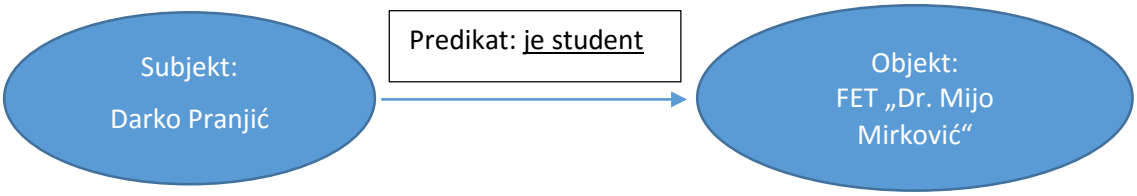
Svaka XML shema treba sadržavati korijenski <schema> element koji u sebi može sadržavati atribute. Određeno je da dokument započinje elementom <prijevozno_sredstvo> koji može imati neodređen broj pod elemenata <auto>. Svaki element <auto> mora imati atribut type="..." i mora imati zadana tri pod elementa. Upravo zadani dozvoljeni formati omogućavaju programima pregled primljene informacije i izbjegavanje grešaka.

3.3 RDF

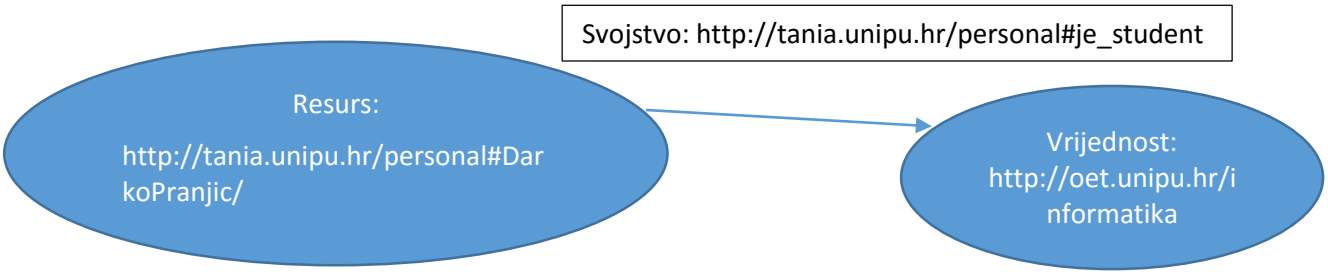
Do sada je prikazano kako se stvara i označava individua, ali pitanje je kako predstaviti neku tvrdnju za tu individuu? Uz pomoć XML-a možemo označiti podatak, ali ne i povezati ga sa drugim. Za tu svrhu stvoren je RDF sloj kojim je omogućeno povezivanje između podataka. Resurs, koji je ranije objašnjen u tekstu mora biti jedinstven i većinom je identificiran URI-jem. Resursi moraju biti opisani uz pomoć svojstva kojima su definirani i vezama s drugim resursima. Te se veze između njih prikazuju grafovima. Kombinacijom web protokola, jezika označavanja podataka i jedinstvenih identifikatora (URI, HTTP,XML...), definiraju se sve dopuštene veze između resursa.

Ideja RDF modela je rastaviti sadržaj na manje dijelove, tako da svaki dio ima definiranu semantiku kako bi ga računalo moglo razumjeti i napraviti nešto korisno s njim. Taj rastavljeni dio je tvrdnja. Znanje je u RDF-u izraženo kao niz tvrdnji koje se sastoje od tri komponente odnosno trojca (eng. *triple*): *subjekt*, *predikat*, *objekt* odnosno *resurs*, *svojstvo*, *vrijednost*. Subjekt i objekt predstavljaju resurse a predikat je poveznica koja ih veže.

Ovo nam daje odgovor na pitanje. RDF-om predstavljamo tvrdnje i temelj je definiranja strukture podataka za semantički web, ali sam po sebi ne opisuje semantiku niti značenje tih podataka. Trojac je prikazan u rečenici Darko Pranjić je student Fakulteta ekonomije i turizma „Dr. Mijo Mirković“.

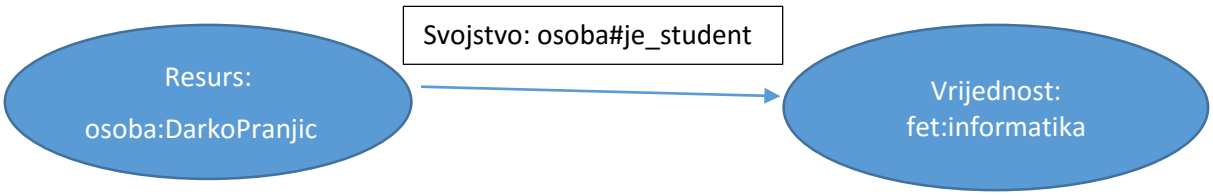


Subjekt i objekt su resursi stoga su označeni većinom URI-jem, predikat predstavlja vezu koja je URI. Predikat se može shvatiti kao ograničenje na jedan od atributa subjekta. Stoga je moguće prikazati: predikat je student je i *svojstvo resursa*, a objekt je *vrijednost* tog svojstva.

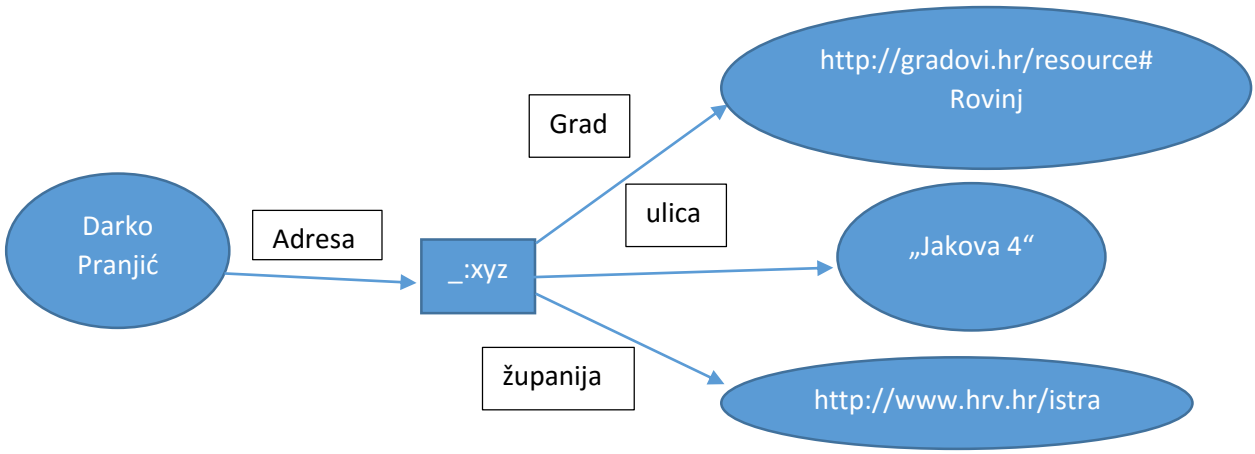


Vrijednost svojstva mogu biti znakovi, brojke (eng. *literal*) ili neki drugi resurs URI. Ukoliko je vrijednost znakovi ili brojke to se može označiti na način da se vidi jezik na kojem je tekst napisan npr. „Semantics“@en , „Semantički“@hr, a on također može označavati i tip podatka `''0.57''^^http://www.w3.org/2001/XMLSchema#float`. Uzevši ovo u obzir, u prethodnom primjeru vrijednost mogu biti slova „FET 'Dr. Mijo Mirković'“.

Kako bi se izbjegli mogući problemi sa konfliktom imena elemenata u XML-u moguće je dodati prefiks koji će označavati element. Prefiks može biti i web adresa, a ta oznaka naziva se imenski prostor (eng. *namespace*). Ako se stavi imenski prostor na URI „Darko Pranjića“ xmls:osoba= http://tania.unipu.hr/personal# i na URI vrijednost xmls:fet=http://oet.unipu.hr/ tada RDF graf izgleda ovako:



Čvor (subjekt ili objekt) može biti i prazan čvor (eng. *blank node*). Prazan čvor označava postojanje resursa koji ima određena svojstva, ali resurs nije definiran URI-jem. Prazan čvor se označava sa _:nekakvoIme.



3.3.1 RDF serijalizacija

Prethodno prikazani način trojaca naziva se čvor – veza – čvor (eng. *node-edge-node*) gdje RDF koristi strukture grafova za semantičke upite. RDF serijalizacija je proces konvertiranja trojaca u formu da ih računala mogu spremati, čitati i prenositi.

3.3.1.1 N-Triples

N-Triples predstavlja najjednostavniji način zapisivanja trojca. Trojac subjekt, predikat i objekt zapisuje se u istom redu točno tim redom. Na kraju svakoga trojca stavlja se točka. Svaki URI se zapisuje unutar izlomljenih zagrada <URI>, a ako ima znakovnih vrijednosti one se pišu unutar navodnika.

<http://example.org/#spiderman> <http://www.perceive.net/schemas/relationship/enemyOf> <http://example.org/#green-goblin> .

3.3.1.2 Turtle

Turtle kombinira prikaz trojca iz N-Triple i mogućnost označavanja imenskim prostorom.

- Trojac subjekt, predikat i objekt zapisuje se u istom redu točno tim redom.
- Svaki URI se zapisuje unutar izlomljenih zagrada <URI>,
- Znakovne vrijednosti se pišu unutar navodnika.
- Ignorira prazna mjesta i prijelom retka izvan identifikatora.
- Prefiksom se može definirati URI tako da ga se ne mora ponavljati.

URI koji pozivamo i koji identificira „Turtle“ format je `http://www.w3.org/2008/turtle#turtle`, a ekstenzija datoteke je .ttl:

Moguće je olakšati ispis da se ne moraju stalno ponavljati subjekt, predikat i objekt .Ako ponavljamo subjekt i predikat onda objekte odvajamo zarezom.

```
@prefix ab: http://learnrdfturtle.com/resource/podaci#

ab:darko ab: posjecena_mjesta ab: Austrija,
                              ab: Mađarska,
                              ab: Irska.
```

Ako se ponavlja subjekt, tada se na kraju trojca umjesto točke upisuje točka zarez.

```
@prefix gr:...

@prefix sv:...
gr: Rovinj  sv: nalaziSe           gr: Istra ;
            sv: brStanovnika       „12 000“ ;
            sv: imaGradonačelnika  gr: Giovanni_Sponza .
```

U Turtle formatu mogu se označiti i prazni čvorovi označavanjem uglatim zagradama [ ] u koje se stavlja subjekt i svojstvo ili svojstvo i objekt.

3.3.1.3 RDF/XML

Jednostavnost strukturiranja XML-a i velikog broja programa koji razumiju njegovu strukturu dovelo je do toga da se za serijalizaciju RDF-a preporuča RDF/XML sintaksa, tj. da se RDF graf predstavlja kao XML dokument.

Kako bi se definirala RDF/XML sintaksa generiran je skup URI-ja kojima su dana određena značenja. Ovaj skup URI-ja je RDF-ov rječnik uvjeta i naziva se „*RDF vocabulary*“. Svaki URI u tom rječniku dijeli `http://www.w3.org/1999/02/22-rdf-syntax-ns#` kao vodeći URI te se kao njegov prefiks uobičajeno koristi „rdf:“ . Razumjeti RDF/XML sintaksu znači razumjeti izraze iz RDF rječnika te načine korištenja prilikom kreiranja RDF grafova unutar XML formata.

Način na koji se kreira RDF/XML datoteka je taj da prva oznaka definira XML dokument i njegovu verziju nakon čega se kreira korijenski „`rdf:RDF`“ element koji označava da XML dokument predstavlja RDF model i unutar kojeg se kreiraju željeni imenski prostori. Potrebno je da se kao prvi imenski prostor stavi RDF „`xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#`“.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF    xmlns:rdf = „http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:osoba= http://tania.unipu.hr/personal#>
<rdf:Description rdf:about="http://tania.unipu.hr/personal#DarkoPranjic">
      <osoba:je_student rdf:resource="http://oet.unipu.hr/informatika"/>
</rdf:Description>
</rdf:RDF>
```

Unutar početne i zatvarajuće oznake RDF modela upisuju se oznake iz RDF rječnika i definirani trojci. Početak tvrdnje označava se s oznakom „rdf:Description“ koja označava da će se nešto opisivati, „rdf:about“ označava subjekta koji se opisuje, svojstvo subjekta je u primjeru „osoba:je_student“, `a` vrijednost tog svojstva je URI koji je zapisan unutar „rdf:resource“.

Pozivanje praznog čvora izvršava se koristeći rdf:parseType=“Resource“, kao što je prikazano u primjeru niže[13]. Subjekt je članak koji se zatim usmjerava na prazan čvor, koji ima svojstvo ime urednika članka i svojstvo web stranice urednika članka.

[13] Primjer je uzet sa stranice http://www.w3.org/TR/REC-rdf-syntax/#section-Introduction

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:dc="http://purl.org/dc/elements/1.1/"
            xmlns:ex="http://example.org/stuff/1.0/">
  <rdf:Description rdf:about="http://www.w3.org/">
    <ex:osnivač rdf:parseType="Resource">
      <ex:ime>Tim Berners Lee</ex:fullName>
      <ex:webStranica rdf:resource="http://www.w3.org/People/Berners-
Lee/"/>
    </ex:osnivač>
  </rdf:Description>
</rdf:RDF>
```

3.3.2 RDF Schema

Opisivanjem RDF-a opisano je kako se predstavljaju tvrdnje sa zadanim svojstvima, tj prikazuju se trojci.

U prethodnom primjeru kao resurs je postavljena „W3“ web stranica, a predstavljena je po svojstvu „osnivač“. Ona ima i druga svojstva koja ju predstavljaju. Pitanje koje se postavlja je kako možemo znati koja druga svojstva postoje? Resurs „Tim Berners Lee“ postoji, ali kako je on definiran, koja su njegova svojstva, da li postoje klase koje su definirane kao njegove podklase ili nadklase?

Ako je resurs predstavljen od strane više ljudi i svi ga opisuju po različitim svojstvima, onda neće biti zajedničkog jezika i u skladu s tim dolazi do manje korisnih informacija. Nadalje, program može definirati i koristiti RDF podatke samo ako zna koje uvjete i klase koristiti, ali sa dosad prikazanim nema mogućnosti stvaranja klasa. Zajednički rječnik bi riješio predstavljeni problem. RDF shema (eng. *schema*) predstavlja upravo takav rječnik.

RDF shema definira klase i hijerarhiju između njih, veze između klasa i njihova ograničenja. Prva verzija RDF sheme izašla je 1998. godine, a završna 2004. godine. Kao RDF tako i RDF Shemu pozivamo upisivanjem odgovarajućeg URI-ja „http://www.w3.org/2000/01/rdf-schema#“, dok joj je prefiks „rdfs“. Jednostavni prikaz RDF sheme je u primjeru niže. Kreirana je klasa „zivotinje“, kao njena podklasa kreirana je klasa „konj“. Nakon nje ponovilo se isto i sa klasom „pas“. Obično su dokumenti s RDF shemom mnogo kompleksniji i sadrže uvjete kao „rdfs:domain“, „rdfs:range“ te tako ograničavaju kojim klasama se subjekt odnosno objekt može koristiti.

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://www.zivotinje.fake/zivotinja#">
<rdfs:Class rdf:ID="zivotinje"/>
<rdfs:Class rdf:ID="konj">
      <rdfs:subClassOf rdf:resource="#zivotinja"/>
</rdfs:Class>
<rdfs:Class rdf:ID="pas">
      <rdfs:subClassOf rdf:resource="#zivotinja"/>
</rdfs:Class>
</rdf:RDF>
```

Nedostaci[14] RDF sheme su:

- „rdf:range“ definira raspon svojstva, recimo jedu, za sve klase. RDF shema ne može deklarirati raspon restrikcije samo za neke podklase. Stoga se ne može reći „krave jedu biljke“, dok druge životinje jedu meso.
- Nemoguće je razdvojiti klase. Ponekad želimo reći da su klase razdvojene, npr. „muško“ i „žensko“ su razdvojeni, a povezani su istom klasom „osoba“
- Ograničenja kardijalnosti. Ponekad želimo staviti ograničenje koliko vrijednosti svojstvo mora ili može uzeti. Npr. Osoba ima točno dva roditelja ili predmet podučava barem jedan profesor

3.3.3 RDFa

Prije pojašnjenja RDFa tehnologije potrebno je znati čemu služe mikropodaci. To su oznake koje se upisuje u HTML sadržaj kako bi se znalo o čemu je sadržaj dokumenta. Daje opis elementima i tako daju semantiku dokumentu. Svaki mikropodatak opisuje specifičnu domenu

[14] Steffen Staab, Rudi Studer, Handbook on Ontologies, 2004 str. 69

(osobu, lokaciju, film, događaj) i dodaje semantičke oznake web stranicama tako da se mogu izdvojiti i obraditi od strane aplikacija.

Kao što su mikropodaci, tako i RDFa služi za opisivanje RDF podataka i RDF tvrdnji unutar XHTML1, HTML4, HTML5, XHTML5, XML, SVG, ePub, OpenDocument dokumenta, samo s mnogo više mogućnosti opisa. RDFa znači RDF atribut (eng. *attribute*), a omogućuje niz novih atributa koji se mogu iskoristiti za označavanje HTML elemenata. Njihovim dodavanjem čini ih se razumljivijima računalima, pa tako i ljudima.

3.3.4 Dublin Core

Dublin Core je mali rječnik koji služi za opis resursa uz pomoć opisa poput „title“, „language“, „subject“, „format“. Pozivamo ga URI-jem „http://www.purl.org/metadata/dublin-core#“, a prefiks koji se koristi je „dc“. Sastoji se od seta od 15 elemenata čiji redoslijed nije bitan.

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:dc= "http://purl.org/dc/elements/1.1/">

<rdf:Description rdf:about="http://www.w3schools.com">
  <dc:description>W3Schools - Free tutorials</dc:description>
  <dc:publisher>Refsnes Data as</dc:publisher>
  <dc:date>2008-09-01</dc:date>
  <dc:type>Web Development</dc:type>
  <dc:format>text/html</dc:format>
  <dc:language>en</dc:language>
</rdf:Description>

</rdf:RDF>
```

3.4 SPARQL

Napravljen je po uzoru na SQL[15] da se može pristupati i manipulirati bazama znanja[16]. Sastoji se od upitnog jezika, protokola HTTP za slanje upita serveru i XML formata gdje se upiti vraćaju. Postoje trenutno dvije verzije, verzija 1.0 počela s korištenjem 15.1.2008 i verzija 1.1 koja je izašla u korištenje 21.3.2013.

SPARQL 1.0 omogućuje:

- Upite nad RDF grafom, samo čitanje podataka,
- Izvlačenje podataka (Literala, URI) iz baza znanja,
- Izvršavanje kompleksnih operacija pridruživanja u jednom upitu,
- Transformaciju RDF podataka iz jednog rječnika u drugi,
- Stvaranje novih RDF grafova.

SPARQL 1.1 omogućuje:

- Dodatne mogućnosti upita kao što su podupiti (subquery), skup funkcija, negacije
- Ažuriranje RDF grafova i potpunu manipulaciju nad njima, upisivanje novih podataka
- Logičan slijed za RDF, RDFS, OWL

RDF baza podataka ili baza znanja (eng. *RDF data store ili Triplestore*) je sistem baze podataka kreiran za unos i prihvat samo RDF trojaca koristeći jezik upita SPARQL koji se prema bazi odnosi kao SQL prema relacijskoj bazi podataka.

SPARQL je baziran na „Turtle“ načinu strukturiranja, a njegovi upiti sadrže trojac koji se naziva uzorak grafa. Potrebno je definirati varijable koje će označavati subjekt, svojstvo ili objekt ovisno o onome što se traži, te se rezultat nakon upita bazi znanja dostavlja kao vrijednost varijable. One se kreiraju jednostavno stavljanjem upitnika (?) ispred njenog naziva „`?imevarijable`“. Povratni rezultati se vraćaju kao tablica sa redovima i stupcima u kojima se nalaze rezultati pretraživanja. Način na koji SPARQL pretraživanje funkcionira je traženje podudarajućih uzoraka između uzorka grafa i spremljenih RDF grafova koji sadrže kolekcije trojaca .

[15] Jezik za upravljanje relacijskim bazama podataka
[16] Baze podataka za spremanje i izvlačenje trojaca

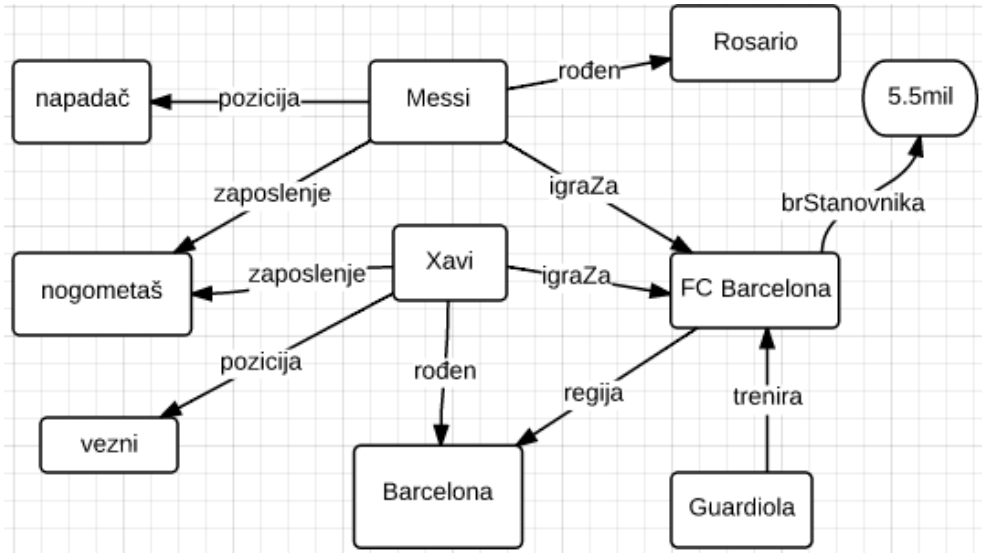
SPARQL pruža 4 različita oblika upita[17]:

1. SELECT - iz izvora podataka izdvaja one elemente koji se podudaraju sa zadanim uzorcima
2. ASK - vraća samo logičku vrijednost istine ili laži (eng. *true ili false*) u ovisnosti postoje li elementi koji se podudaraju po zadanim uzorcima u upitu
3. CONSTRUCT - stvara RDF graf prema predlošku zadanom u upitu te koristi „where“ dio upita da bi zamijenio varijable u predlošku s konkretnim vrijednostima
4. DESCRIBE – vraća jedan RDF graf s podacima o URI-u; URI može biti konstanta ili varijabla čija se vrijednost dobije iz „where“ dijela upita

Kao u SQL-u „SELECT“ naredbom govorimo o kojim varijablama želimo dobiti informacije nakon čega se u „WHERE“ naredbu upisuje uzorak trojca. Naredbom „FROM“ može se odabrati jedan ili više RDF grafova na koje će se upit odnositi nakon čega se mogu upisati dodatni upitni modifikatori.

Jednostavan primjer naveden je na slici:

Slika 3. RDF graf



Izvor: http://0agr.ru/wiki/index.php/SPARQL

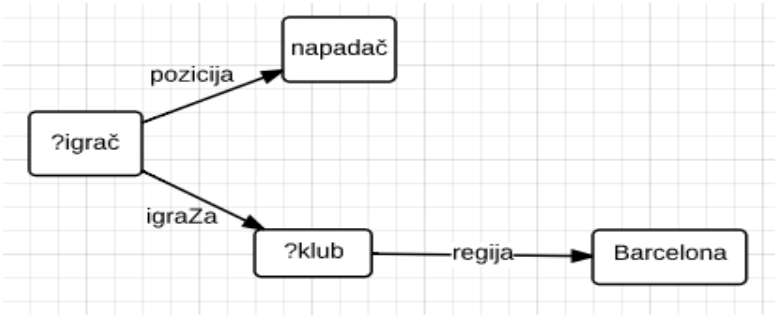
Ako uzmemo npr. da želimo pronaći „napadača“ iz kluba „FC Barcelona“ SPARQL upit bi izgledao ovako:

[17] https://www.fer.unizg.hr/_download/repository/10._Semanticki_web_-_SPARQL.pdf

```
PREFIX barca: <http://example.com/barca#>
SELECT ?igrac ?klub
      WHERE {
            ?igrac barca:pozicija barca:napadač.
            ?igrač barca:igraZa ?klub.
            ?klub barca:regija barca:Barcelona.
            }
```

Upit je sam po sebi graf pa ga takvim možemo prikazati:

Slika 4. SPARQL uzorak grafa



Izvor: http://0agr.ru/wiki/index.php/SPARQL

Dobiveni rezultat biti će samo Messi koji je napadač i FC Barcelona klub.

Novosti koje su uvedene s SPARQL 1.1 daju veću mogućnost manipuliranja podacima. Sada se SELECT naredbom mogu npr. povezati varijable i spremiti ih u jednu novu varijablu naredbom „SELECT fn:concat(?ime, " ", ?prezime) AS ?imeOsobe“[18]. Također moguće je zbrajanje, oduzimanje, množenje i sl. Omogućeno je skraćivanje koda po svojstvu stoga se primjerice:

```
where {
?ja foaf:mbox <mailto:zgajo@net.hr>.
?ja foaf:knows ?prijatelj.
?prijatelj foaf:name ?ime.
}.
```

Ne mora pisati stavljajući za svaki trojac novi red, može se napisati puno kraće:

```
where {
?ja foaf:mbox <mailto:zgajo@net.hr>.
?ja foaf:knows/foaf:ime ?ime.
}
```

Kao što je prije navedeno dodana je mogućnost ažuriranja trojaca i potpuna manipulacija nad njima. Za manipulaciju RDF grafom u bazi znanja uveden je SPARQL 1, neke od njegovih naredbi su sjedeće:

[18] Liyang Yu, A developer's guide to the semantic web, 2011. str. 283

„INSERT DATA“ kreira graf ukoliko isti ne postoji, također može kopirati RDF tvrdnje iz drugih RDF dokumenta u trenutno korišteni. „DELETE DATA“, briše RDF tvrdnju, a može se uz uvjet „WHERE“ limitirati što da se briše.

Način na koji se SPARQL upiti vrše je uz SPARQL Endpoint. SPARQL Endpoint je programsko sučelje preko kojeg se mogu slati upiti i prikazati rezultati upita. Jedan od najpoznatijih SPARQL Endpoint sučelja je Virtuoso SPARQL Query Editor kojeg koristi DBpedia. Upiti se upisuju u „Query Text“, format prikaza može se birati između HTML, RDF/XML, JSON, Javascript, N-Triples, Turtle, xls i ostalih formata. Ukoliko imamo pretraživač koji podržava prikaz RDF/XML datoteke može se vidjeti rezultat u tom formatu. U zaglavlje (eng. *header*) datoteke spremaju se varijable upita koje se ispisuju unutar `<results>` oznaka. HTML prikazom dobili bi ime varijable i tabelu s rezultatom.

Primjer RDF/XML sadržaja izgleda ovako[19]:

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">

  <head>
    <variable name="x"/>
    <variable name="hpage"/>
    <variable name="name"/>
    <variable name="age"/>
    <variable name="mbox"/>
    <variable name="friend"/>
  </head>

  <results>

    <result>
      <binding name="x">
            <bnode>r2</bnode>
      </binding>
      <binding name="hpage">
            <uri>http://work.example.org/bob/</uri>
      </binding>
      <binding name="name">
            <literal xml:lang="en">Bob</literal>
      </binding>
      <binding name="age">
            <literal
      datatype="http://www.w3.org/2001/XMLSchema#integer">30</literal>
      </binding>
      <binding name="mbox">
            <uri>mailto:bob@work.example.org</uri>
      </binding>
    </result>
    ...
  </results>
</sparql>
```

[19] http://www.w3.org/TR/rdf-sparql-XMLres/

3.5 ONTOLOGIJA

Filozofska definicija ontologije navodi je to disciplina koja se bavi problemom bitka i biti bivstvujućeg, zakonima i strukturom bivstvujućeg uopće; sastavni (prvi) dio metafizike, pored psihologije, teologije i kozmologije[20].
Ontologija je središnja grana metafizike te istražuje odnose o vrstama stvari u svijetu i samim njima. Znanost o ontologiji potiče iz grčke filozofije gdje su se tretirale ontološke ideje kao klasifikacije stvari i pitanja zaključka (silogizam[21]).

Premise: *Svi ljudi su smrtni.*

Sokrat je čovjek.

Zaključak: *Sokrat je smrtan.*

Motivacija filozofima da se bave ontologijom je da se predstavlja znanje i utvrdi novo.

Do sada je prikazano kako se povezuju RDF resursi te kako se definiraju semantički odnosi između njih pomoću RDF sheme. Ono što ontološkim jezikom OWL (eng. *Web ontology language*) čini je isto kao i s RDF shemom, definiraju se klase i veze između njih za određenu domenu. Ali ti odnosi su detaljnije definirani dajući jednakosti i nejednakosti između klasa i svojstva ali i uvođenjem više mogućnosti za opis između njih. Svi nedostaci RDF Sheme pokriveni su unutar OWL-a. Stoga se mogu izraditi web aplikacije s većom mogućnošću logičnog zaključivanja. OWL sadrži skup pojmova koji se koriste za opisivanje i predstavljanje određene domene znanja (školovanje, automobilizam, medicina) . Definicija[22] ontologija u semantičkom „Webu“ bila bi:
Ontologije su formirane definicije rječnika koje omogućuju definiranje novih veza između složenih struktura kao i između članova definiranih klasa.
Jezici koji se koriste su: RDF za iskazivanje tvrdnji, RDF Shema za opis podataka i OWL koji omogućava jasnije definiranje veza i podataka.

Postoje dvije verzije ontološkog jezika, OWL 1 i OWL 2. OWL 1 se počeo koristiti 2004. godine, a OWL 2 je postao dio W3C standarda 2009. godine. OWL je baziran na jeziku računalne logike tako da znanje izraženo u OWL-u sa računalnim programom može biti

[20] http://hjp.novi-liber.hr/index.php?show=search_by_id&id=eFlkUBQ%3D
[21] Oblik logičkog deduktivnog zaključivanja u kojem se iz dvaju ili više sudova (premisa) prema određenim pravilima izvodi zaključak
[22] LeRNING Sparql, Bob DuCharme, O'Reilly, 2011. str 39

obrazloženo da provjerava konzistentnost tog znanja ili da implicitno[23] znanje pretvara u eksplicitno[24]. Sve ontologije kreirane koristeći OWL 1 čitljive su od aplikacija koje razumiju OWL 2.

Postoje tri podjezika OWL-a, a svaki proširuje prethodni. OWL Lite, OWL DL i OWL Full.

- OWL Lite – Kardinalnost ograničenja je svedena na vrijednosti 0 i 1, koristi se kod jednostavnijih ontologija, nema mogućnosti izjave „owl:equivalentClass“ koja služi za povezivanje anonimne klase.
- OWL DL –Sadrži sve konstruktore ali ima ograničenja na načine na koje se OWL konstruktori i RDF mogu koristiti. Klasa ne može biti istovremeno član druge klase. Postavljena su ograničenja na svojstva. Ime DL je dano zbog njegove korespodencije s deskriptivnom logikom, područjem istraživanja koje čini temelj OWL-a
- OWL Full – dizajniran da sačuva kompatibilnost s RDF shemom. Sadrži sve mogućnosti OWL-a.

Svi pojmovi imaju sljedeći URI kojim ih se poziva: http://www.w3.org/2002/07/owl#, a kojem se uobičajeno daje imenski prostor „owl:“.

Tvrdnja koja je označena OWL ontologijom naziva se aksiom, npr. klasa student je podklasa od klase osoba. Svaki pojedini dio trojca u aksiomu naziva se entitet. Kombinacije entiteta kako bi se oformili novi aksiomi nazivaju se izrazi (eng. *expressions*).

Postoji par sintaksi pisanja i dijeljenja ontologija:

- Functional-style sintaksa
- RDF/XML sintaksa – Jedina sintaksa koja je podržana od svih OWL alata. Opisana u RDF poglavlju
- Manchester sintaksa
- OWL/XML

U OWL 1 koristi se „owl:Thing“ kao korijen svih klasa a klase se označavaju sa „owl:Class". Ukoliko se želi ograničiti klasa potrebno je staviti oznaku „owl:Restriction“ u koju se upisuje svojstvo i što je i ograničenje.

[23] Nesvjesno pamćenje, npr. vožnja biciklom
[24] Svjesno pamćenje činjenica ili događaja

Dopuštene su dvije vrste ograničenja: po vrijednosti i kardijalnosti. Po vrijednosti ograničava raspon koji se pretražuje, a kardinalnosti broj vrijednosti koje može zadovoljiti svojstvo.

Ako program razumije ontologiju znači da može parsirati i stvoriti popis aksioma na temelju oznaka ontologije nakon čega se stvara prikaz sadržaja RDF dokumenta koji sadrži samo ono bitno.

OWL ima raznih mogućnosti za izgradnju dokumenta ograničavajući instance klase, označavanjem različitosti klasa i označavanjem jednakosti klasa.

```
<owl:Class rdf:about="http://www.primjer.com/primjer#igrač">
      <rdfs:subClassOf rdf:about="http://www.primjer.com/primjer#nogIgrač">
            <owl:Restriction>
                  <owl:onProperty rdf:resource="#napadač"/>
                        <owl:Cardinality rdf:datatype=
                  "http://www.w3.org/2001/XMLSchema#nonNegativeInteger">
                              1
                        </owl:minCardinality>
            </owl:Restriction>
      </rdfs:subClassOf>
</owl:Class>
```

Definirana je klasa „igrač“ koji je podklasa od „nogIgrač“, svojstvo mu je „napadač“, te je postavljeno ograničenje na mogućnost pronalaska samo jedne instance klase „napadač“.

3.5.1 FOAF

Definicija kojom bi se mogla opisati FOAF sintaksa je:

„*FOAF (eng. Friend Of A Friend) predstavlja računalu čitljive metapodatke koji opisuju osobe, njihove aktivnosti i omogućuju povezivanje s drugim osobama.“*

Osnovan je sa idejom stvaranja računalu poznatih podataka u području osobnih „Web“ stranica i društvenog umrežavanja. FOAF sintaksa je napisana uz OWL ontologiju, stoga se stvarajući dokument koriste RDF za strukturiranje dokumenta i FOAF ontologija da podaci u sadržaju dobiju značenje.

Postoji mnogo web stranica u kojima se nalaze osobni podaci, email, slike ili URL veze prema stranici od prijatelja, ali da se pronađe prijatelj ili prijatelj od prijatelja s kojim imamo zajednički interes može biti frustrirajuće. Tada je za pretragu potrebno otvarati web stranice od svakoga posebno i čitati čak nebitne informacije samo da bi došli do traženog podatka.

Ubacivanjem računalu čitljivih metapodataka u takvu stranicu omogućuje se da osobe budu povezane i može se jednostavno vidjeti tko od prijatelja ima posao u istoj branši i tako tražiti posao ili radnika ukoliko je to cilj.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

    a foaf:Person ;
    foaf:name "Darko Pranjić" ;
    foaf:mbox <mailto:test@gmail.com> ;
    foaf:homepage <http://www.darko-pranjic.com> ;
    foaf:nick "Nicky" ;
    foaf:depiction <http://www. darko-pranjic.com/darko_small.jpg> ;
    foaf:interest <http://www.semantic.org> ;
    foaf:knows [
        a foaf:Person ;
        foaf:name "Obama"
    ] .
```

Primjer prikazuje označavanje u turtle formatu. Označena je osoba „Darko Pranjić“, njegova email adresa, web stranica i oznaka da poznaje osobu „Obama“. Moguće je napraviti osobu „Obama“ sa identifikatorom email adresom pa umjesto da smo napisali „`foaf:name "Obama"`“ veza može biti i email adresa tj. „`foaf:mbox mailto:obama@gmail.com`“ što će računalo znati da je prijatelj „Obama“. Način na koji se povezuje krug prijatelja je sljedeći:

1. Kreira se FOAF dokument koji treba sadržavati foaf:knows i foaf:seeAlso
2. Web stranicu treba povezati sa FOAF dokumentom
3. FOAF koristi agenta za indeksiranje koji skuplja sve FOAF dokumente
4. FOAF održava repozitorij i pazi da su podaci ažurirani.
5. FOAF pruža korisničko sučelje kako bi mogli pretraživati prijatelje i mnogo drugih aktivnosti.

3.6 LINKED DATA

Web 2.0 povezuje web stranice preko URL-ova, ali računala ne razumiju što web stranica na koju vodi poveznica znači za stranicu sa koje idemo. Semantički web daje značenje podacima, a povezani podaci (eng. *Linked data*) označavaju podatke na način da web stranica dobije značenje od strane drugih stranica kojima je povezana.

Tim Bernes Lee postavio je četiri načela za povezivanje podataka:

- Korištenje URI-ja za imenovanje informacija. Sve mora biti povezano URI-jem
- Da bi ljudi mogli potražiti informacije potrebno je koristiti HTTP protokol. Kada se potraži informacija iz URI-ja ili ćemo dobiti informaciju čitljivu ljudima ili čitljivu računalu. To ostvarivo preko HTTP pregovora o sadržaju.
- Kada se pretražuje URI trebaju pružiti korisne informacije koristeći RDF, SPARQL.
- Uključiti poveznice drugih URI-ja tako da i oni mogu otkrivati nove informacije. Povezati RDF reference među podacima između različitih izvora podataka, kako bi se našla informacija povezana po sadržaju.

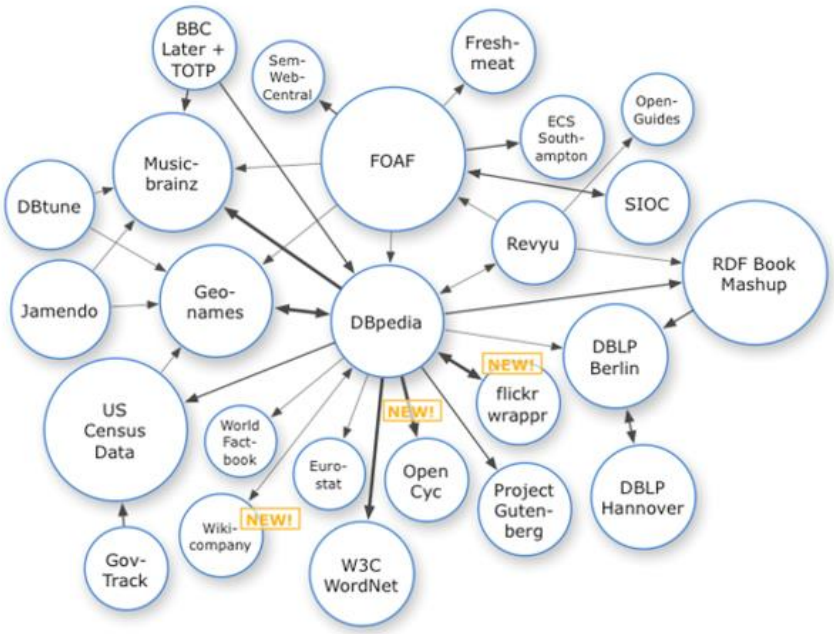
Kako bi se prikazala ideja povezanih podataka kreiran je Dijagram povezanih podataka u oblaku (eng. *Linked Data Diagram of the Cloud*) gdje su prikazane veze između web stranica čiji su podaci međusobno povezani u „Linked data“ formatu. Za dijagram su zaslužni Richard Cyganiak i Anja Jentzsch te članovi zajednice javno dostupnih podataka (eng. *Linked Open Data Community*) i drugih organizacija koji su povezali podatke na ovaj način. Definicija kojom LinkedData.org predstavlja povezane podatke je:

„Web nam omogućuje da povezujemo srodne dokumente. Slično nam omogućuje da povežemo srodne podatke. Izraz povezanih podataka (Linked Data) odnosi se na skup najboljih praksi prilikom objavljivanja i povezivanja strukturiranih podataka na Webu. Ključne tehnologije koje podržavaju povezane podatke su URI (identificira identitete ili koncepte u svijetu), HTTP (jednostavan ali i univerzalan mehanizam za prijem resursa ili opisa resursa) i RDF (Model podataka baziran na grafu s kojim se strukturira i povezuju podaci koji opisuju stvari u svijetu).“ [25]

[25] Alex Williams, 18.1.2011 , http://readwrite.com/2011/01/18/the-concept-of-linked-data

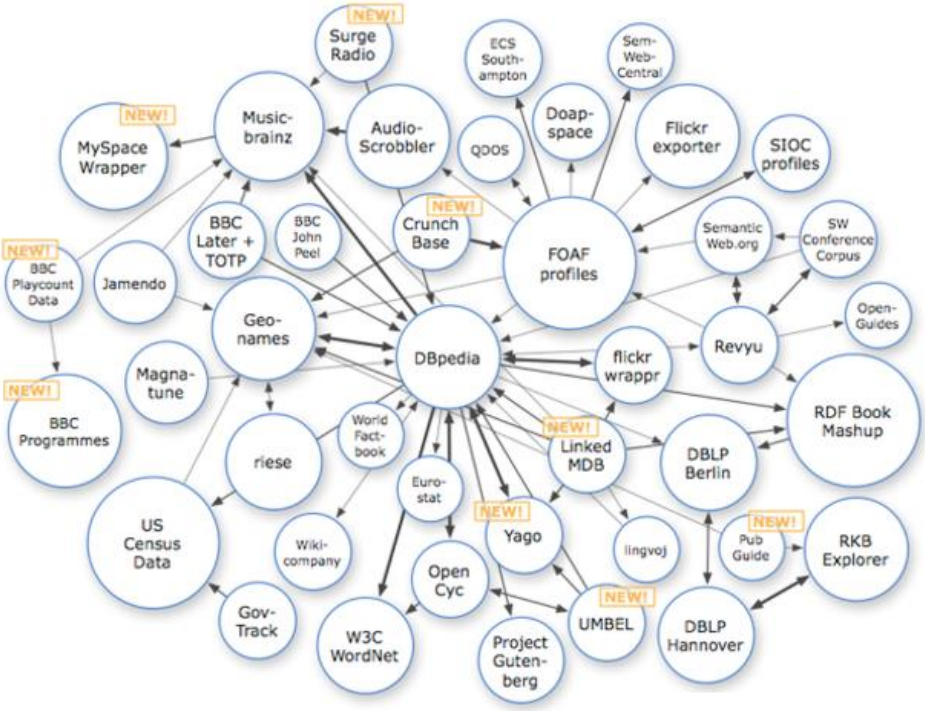
Veličina „oblaka povezanih podataka“ po godinama:

Slika 5. Veličina „oblaka povezanih podataka“ 2007. godine



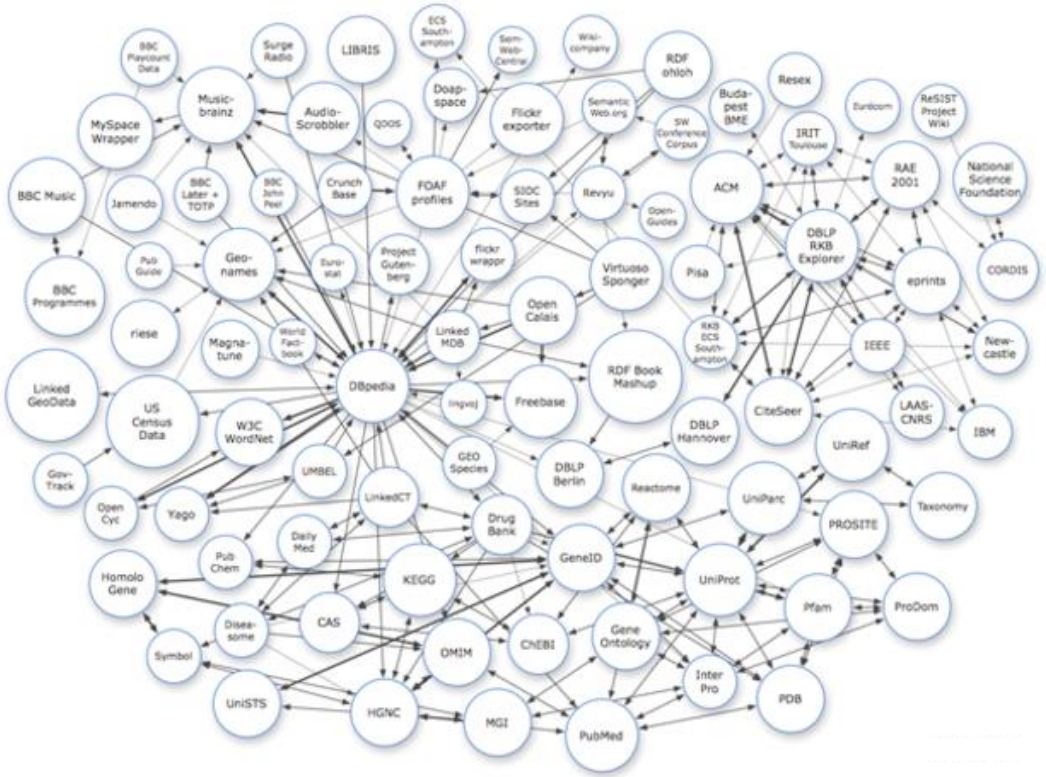
Izvor: http://readwrite.com/2011/01/18/the-concept-of-linked-data

Slika 6. Veličina „oblaka povezanih podataka“ 2008. godine



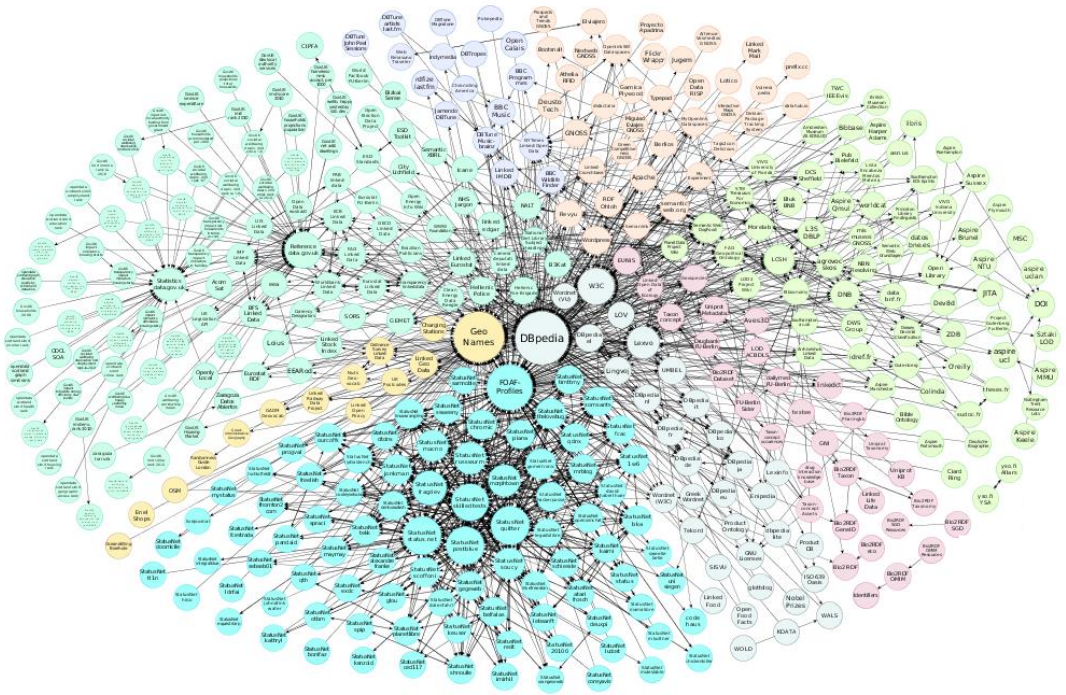
Izvor: http://readwrite.com/2011/01/18/the-concept-of-linked-data

Slika 7. Veličina „oblaka povezanih podataka“ 2009. godine



Izvor: http://readwrite.com/2011/01/18/the-concept-of-linked-data

Slika 8. Veličina „oblaka povezanih podataka“ 30.8.2014



Izvor: http://readwrite.com/2011/01/18/the-concept-of-linked-data

Tablica prikazuje tri najčešće korištena vezujuća predikata za svaku pojedinu domenu:

Tablica 2. Tri najčešće korištena predikata za povezivanje po kategoriji.

Kategorija	Predikat	Korišten	Kategorija	Predikat	Korišten
Društvene mreže	foaf:knows	60.27%	znanost	owl:sameAs	52.17%
	foaf:based_near	35.69%		rdfs:seeAlso	48.48%
	sioc:follows	34.34%		dct:creator	21.74%
publikacije	owl:sameAs	32.20%	vlada	dct:publisher	47.57%
	dct:language	25.42%		dct:spatial	30.10%
	rdfs:seeAlso	23.73%		owl:sameAs	24.27%
sadržaj stvoren od strane korisnika	owl:sameAs	53.13%	geografija	owl:sameAs	64.29%
	rdfs:seeAlso	21.88%		skos:exactMatch	21.43%
	dct:source	18.75%		skos:closeMatch	21.43%
medij	owl:sameAs	81.25%	više domena	owl:sameAs	80.00%
	rdfs:seeAlso	18.75%		rdfs:seeAlso	52.00%
	foaf:based near	18.75%		dct:creator	20.00%

Izvor: http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/

Povezani podaci ne trebaju biti otvoreni svima za korištenje, mnogo je važnih podataka koji bi se trebali koristiti samo interno. Godine 2010.-e Tim Bernes Lee izdao je kriterij od 5 zvjezdica koji se dodjeljuje ovisno na koji način su naši podaci dostupni. Javno dostupni povezani podaci napravljeni su sa besplatnom licencom i njima se dodjeljuju zvjezdice. Kada se objavi javni

podatak znači da se može kopirati ili modificirati, tj koristiti na željene načine. Način dodjele zvjezdica je sljedeći:

* dostupno na webu u bilo kojem formatu ali ne može se upravljati s njim. Primjer slika ili pdf dokument

** dostupno kao računalu čitljivi strukturirani podaci (Excel za razliku slike sa tablicom)

*** kao i dvije zvjezdice uz dodatak da je podatak čitljiv od velikog broja aplikacija (csv umjesto Excela)

**** Sve prije sa dodatkom korištenja W3C standarda URI za označavanje podatka

***** Sve navedeno sa dodatkom da se povežu podaci sa podacima na drugim lokacijama

3.6.1 DBPEDIA

Kao što se vidi na dijagramu povezanih podataka na svim dijagramima se u sredini i prema kojoj ide najviše poveznica nalazi se DBpedia. To je i prvi javno objavljeni skup semantičkih podataka. Wikipedija je najveća svjetska enciklopedija, a njezina semantička verzija je DBpedia.

Wikipedija sama po sebi nudi jako limitirane mogućnosti upita i pretraživanja, a sadrži mnogo strukturiranih podataka koje Dbpedia sprema u svoju bazu u RDF formatu. Npr. teško je pronaći sve rijeke koje se ulijevaju u Jadransko more ili sve Hrvatske pjevače iz 20 stoljeća. Dbpedia takve pretrage izvršava i vraća veoma korisne podatke. Na Wikipediji se ti strukturirani povezani podaci nalaze se u tablici sa desne strane. Uzmimo na primjer pretraživanje podataka o gradu Berlinu, utipkavši naziv grada pojavljuju nam se podaci koji ga karakteriziraju, od njegovih koordinata, države, gradonačelnika, površine i ostalog. Rezultat na DBpediji će dati mnogo više podataka, ali prikaz je fokusiran samo na sve povezane podatke s njim.

Način na koji Wikipedia i Dbpedia funkcioniraju je sljedeći. DBpedijino izvlačenje sastavljeno je od četiri faze:

- Wikipedijina stranica se čita od vanjskog izvora,

- Svaka Wikipedijina stranica je parsirana i kod je transformiran u stablo apstraktne sintakse,
- Transformirano stablo je proslijeđeno Dbpedijinim ekstraktorima za razne svrhe. Svaki ekstraktor iz stabla izvlači RDF tvrdnje,
- Skupljene RDF tvrdnje se sinkroniziraju i spremaju.

4 TRAŽILICE SEMANTIČKOG WEBA

U današnje vrijeme postoji mnogo web tražilica koje dohvaćaju informacije, ali dohvaćanje smislenih informacija je teško. Za prevladavanje ovog problema semantička tehnologija igra veliku ulogu.

Tražilice obično pretražuju web stranice pokušavajući naći podudaranje upita sa sadržajem web stranica, takav način pretraživanja naziva se pretraživanje po ključnoj riječi. Naprednim algoritmom filtriraju stranice, ali filtar prolaze čak i one pretražene, a nebitne. Na inteligentne upite korisnika ne daju tako inteligentne rezultate jer su ovisne o informacijama koje su dostupne na web stranici. Takvim pretraživanjem dovodi se do netočnih ili nepouzdanih rezultata pa tako do korisnikovog nezadovoljstva rezultatom.

Za procjenu kvalitete primljenih informacija tražilice najčešće koriste preciznost (eng. *precision*) i povlačenje (eng. *recall*). Preciznost označava postotak primljenih stranica koje su relevantne, povlačenje označava postotak relevantnih stranica koje su primljene

Ako se napravi pretraga za filmovima na kojima je auto na naslovnici postotak bi se izračunao ovako:

	Relevantno	Irelevantno
Primljene	8	4
Nisu primljene	2	10

Povlačenje = 8/10= 0.8 * 100% = 80% - primljeno je 80% relevantnih stranica

Preciznost = 8/12 = 0.66 * 100% = 66% - je postotak od ukupno primljenih relevantnih stranica

Fokus ovih tražilica je na rješavanju upita sa približnim rezultatom u što kraćem vremenu. Semantička tehnologija daje približne do željenih rezultata. Cilj joj je dakako poboljšati dohvat informacija.

Semantičkom pretragom možemo nazvati pretragu koja uključuje entitete označene semantičkim metapodacima, dohvat datoteka sa sadržajem temeljenim na entitetima te mogućnost dobivanja sličnih rezultata povezanih semantičkim vezama sa entitetom.

Ono što semantička tražilice poboljšavaju prilikom pretrage je:

1. Sam upit je proširen sinonimima za dobivanje istog rezultata

2. Daje rezultate koji su vezani s upitom po sadržaju, a ne sadrže u sebi riječi iz upita
3. Korisniku je omogućeno proširivanje znanja prateći povezane podatke sa stranica
4. Pretraživanje razumije upite pa daje odgovore tražeći povezanost svih entiteta iz upita

Postoje četiri pristupa semantičke pretrage, a većina semantičkih tražilica koristi kombinaciju više njih, pristupi su sljedeći:

1. Kontekstualna analiza: analizira upit za shvaćanje u kojem kontekstu se traži odgovor
2. Razumijevanje: funkcionira na način da izvlači dodatne tvrdnje iz postojećih, npr. ako sustav zna tko su djeca od osobe „X“ i zna tko su djeca od njegove djece, tada može utvrditi od koga je osoba „X“ pradjed.
3. Razumijevanje ljudskog jezika: analiziraju se entiteti iz upita koristeći njihove oznake, te tako razumije da postavljeni upit predstavlja npr. osobu.
4. Korištenje ontologije: Na ovaj način proširuje se sam upit i znanje o domeni. Npr. upitom „auto“, sustav razumije da „auto“ označava „vozilo“ i proširuje upit na njega.

4.1 Hakia

Hakia je ugašena 2014. godine, ali do tada je bila jedna od najpopularnijih semantičkih tražilica koja je prihvaćala i pitanja kao upit. Korišten je algoritam koji je dekodirao upit na način da se odredi njegovo značenje koristeći redom OntoSem, QDEX i algoritam semantičkog rangiranja. OntoSem kategorizira sve vezano za koncept upita, čak i sinonime riječi iz upita, koji su korišteni u drugim tvrdnjama. QDEX izdvaja sadržaj dobivenih web stranica kao skup upita, pokušavajući pronaći podudaranje sa sadržajem zadanog upita. Rangiranje web stranica ovisi o trenutnom upitu, njegovom značenju pa i starosti sadržaja u skladu s kojim se prikazuju rezultati.

4.2 DuckDuckGo

Tražilica koja odvaja klasične rezultate i rezultate o informaciji. Jedinstvenost tražilice je kada se postavi upit koji ima više značenja, tražilica omogućava korisniku da odabere značenje upita. Ako je upit „banana“ , rezultat će biti voće, država, osoba, film, a nama se nudi mogućnost odabira klase i dobivanje informacija o tome.

4.3 Sindice

Sindice je jedna od semantičkih tražilica koje odbacuju tekst i koncentrirane su prema semantičkim oznakama. Sindice projekt trajao je 5 godina, točnije pokrenut je 2007. godine, a ugašen je 2012. godine. Dizajniran je sa idejama da pruža jednostavnost, brzi odaziv, ažuriranje u realnom vremenu i održavanje jednake brzine povrata rezultata sa povećanjem broja trenutno povezanih korisnika.

U početku su stvorili svoj način indeksiranja[26], koji je radio na način da otkriva web stranice uz pomoću web pauka, koji su stalno ažurirali URL stranica u kojima se spominje URI, a kopije takvih stranica su se spremale na njihov server.

Nakon nekog vremena za eksperimentiranje se počela koristiti Apache Lucene, besplatna biblioteka za dohvat informacija. Indeksiranje je konstantno je davalo brza ažuriranja za koja su postojale tehnike rangiranja.

Za pretraživanja trojaca iz RDF modela pokrenuo se projekt SIREn (eng. *Semantic Information Retrieval Index*) koji je omogućavao ne samo URI pretragom nego i pretrage oznaka.

Zahvaljujući ranijim rezultatima prvih verzija Sindice-a DERI institut im je dao novčanu potporu kojom su resursi za obavljanje operacija proširen na klaster[27] od 60 računala. Cilj je bio istražiti što se može napraviti u RDF i domeni weba podataka sa big data tehnikama Hadoop i HBase, koji su davali mogućnosti pohranjivanja i obrade velikih količina povezanih podataka.

[26] Razdvajanje dijelova stranice i njeno ubacivanje u bazu podataka kako bi ona bila dostupna tražilici i usporedbi sa drugim stranicama
[27] Skup povezanih računala koja rade zajedno na način da se mogu gledati kao jedno računalo.

Izgradnjom vlastitog softvera za procesiranje dokumenata (eng. *pipeline*) dobila se mogućnost manipuliranja dokumentom na način da se pretraživao podatak u dokumentu te se indeksirao na Hadoopu. Obrade podataka Hadoop-a uključivale su otkrivanje sadržaja u oznakama i ispravljanje njihovih grešaka, izdvajanje RDF-a, dohvaćanje ontologija što se na kraju šalje u SIREn.

Kako bi što više aplikacija moglo izdvajati RDF na isti način, „kod" koji se koristi kod izdvajanja iz Hadoopa učinjen je javno dostupnim kao online alat Any23 (eng. *Anything to triple*). Zatim je stvorena aplikacija „Sindice inspektor" bazirana na Any23 koja je omogućavala korisnicima prikaz trojaca putem grafa ili odnosa roditelj – dijete i braća - sestre.

Kako bi rezultati bili razumljivi i lako čitljivi običnim korisnicima stvoren je preglednik „Sig.ma" koji je opisan u poglavlju s preglednicima.

Način na koji je Sindice ostvario komercijalnu vrijednost je na način da su vlasnicima stranica, koje imaju semantičke oznake, pružali uslugu na web stranici (Sindice Site Services).

Sindice bi pokupio oznake sa stranice, izvršio detaljnu pretragu iste te držao kopiju podataka sa nje stalno ažuriranom. Zatim bi kombinirao podatke web stranice sa sličnim podacima, te nudio usluge koje su korisne i povezane sa stranicom.

Jedna od usluga u praksi je slična načinu rada kao na stranicama eBay, IMDB i sličnima. Ukoliko se odabere film na stranici IMDB u jednom dijelu stranice nude se ostali slični filmovi za pogledati. Tako nam se i na eBay stranici nude slični proizvodi sa onim odabranim.

Usluge koje su nudili funkcionirale su na način da se u web stranicu ubaci javascript kod, koji je čitao oznake na stranici i slao ih na server, nakon čega su se se nudile druge stranice na toj domeni isto kao što je opisano za IMDB.

Sindice tim je prešao u startup tvrtku SindiceTech gdje su nastavili izradu SIREn-a. Također ostvarili su suradnju s Google-om te izradili Frebase i Graf znanja (eng. *Knowledge graph*) koje Google koristi. Freebase je repozitorij strukturiranih podataka koji sadrži informacije o raznim tipovima podataka (ljudima, mjestima, knjigama, filmovima itd) i svojstvima (datum rođenja, GPS i sl.), a graf znanja koristi te podatke i prikazuje ih u tablici koja se može vidjeti sa desne strane rezultata pretraživanja.

4.4 Google i semantička pretraga

Stvaranjem prvih verzija Google tražilica lako su se nalazili načini kako prevariti tražilicu da misli kako određenu stranicu mora dobro rangirati. Evolucijom Google-a stvarali su se načini kako smanjiti varanja i poboljšavale pretrage da se korisniku vrate što relevantnije informacije na vrhu rezultata pretrage. Prije izlaska „Caffeine“ algoritma 2009. godine rangiranja stranica rijetko su se osvježavala pa bi prva stranica ostala netaknuta po par tjedana, no s njegovim izlaskom rezultati pretrage mijenjali su se nekoliko puta na dan. Tri velike promjene desile su se u Google algoritmu pretraživanja u posljednjih par godina, svaki algoritam je nazvan svojim imenom: „Panda“, „Penguin“ i „Hummingbird“.

Pokrenut 2011. godine, svrha „Panda“ algoritma bila je bolje rangiranje kvalitetnije napravljenih i degradiranje nekvalitetno napravljenih „Web“ stranica. Radi na način da gleda što se na stranici nalazi, točnije da li ima duplirane članke, pravopisne greške, malo izmijenjene ponovljene tekstove. Pogodio je rangiranje mnogo stranica koje su sadržavale ukradene tekstove kako bi se nalazili na što boljem mjestu prilikom pretrage.

Penguin algoritam započeo je s korištenjem 24.4.2012 a cilj mu je smanjiti povjerenje prema stranicama koje su sadržavale poveznice (eng. *unnatural links*) s kojima se manipulira rangiranjem pretraživanja. Način na koji funkcionira rangiranje je sljedeći: ukoliko respektirana stranica od strane Google-a napravi poveznicu prema našoj stranici to donosi do nemjerljivo bolje pozicije nego kada to napravi ne respektirana stranica, ali ako veliki broj nerespektiranih stranica daje poveznicu to pozitivno utječe na poziciju.

Prilikom obavijesti o izlasku „Hummingbird“-a dana 26.9.2013 on je već bio u funkciji mjesec dana, ali je rijetko tko uspio zapaziti promjenu. Također je rečeno je da od izlaska „Google Caffeina“ algoritam nije bio tako puno ažuriran. Cilj Hummingbird algoritma je bolje razumjeti korisnikov upit i semantiku upita. Trenutno još ne daje dobre rezultate na hrvatskom jeziku ali na upite engleskog jezika vraća uglavnom relevantne podatke. Na upit „What is the best place to eat pizza“ ima mogućnost da raspozna da s „place“ korisnika najvjerojatnije interesira „restoraunt“, tj. algoritam razumije sinonime. Ima mogućnost brzo analizirati duža, kompleksnija pitanja i pružiti najbolji odgovor korisniku sa što manje klikova.

Mogućnosti i tehnike pretraživanja su sljedeće:

1. Ovisno o pitanju koristi semantičku pretragu (uzima u obzir zašto, kada, tko) za razliku od prijašnjih pretraga po ključnim riječima

2. Ista pretraga može donijeti različite rezultate za različite korisnike. Prijašnja je donosila svima iste rezultate
3. Uzima korisnikove informacije kao lokaciju, prijašnja skidanja sa „Web“-a i sl.
4. Koristi Google Knowledge Graph za odgovarati na duga kompleksna pretraživanja.
5. Koristi govornu pretragu i kreće sa odgovaranjem na pitanje za razliku od prošlih verzija algoritama kada je korisnik dobivao više opcija rezultata i nastavljao pretraživanje za odgovorom.

Ažuriranjem na Hummingbird počelo se koristiti glasovno pretraživanje (eng. *voice recognition search*). Hummingbird proširuje upotrebu Knowledge grapha za omogućavanje usporedbe između pretraživanih objekata (eng. *"compare moon vs mars"*).

Odgovora na kompleksni upit i poboljšava proces pitanja jedno za drugim. Tako da ukoliko pretražimo Google glasovnim pretraživanjem „Show me the pictures of Eiffel tower“ on nam prikaže slike, a pitanjem nakon toga „How tall is it“ Google već zna da je se priča o Eiffelovom tornju i daje odgovor na njegovu visinu i sve to korištenjem Knowledge Grapha.

5 SEMANTIČKI PREGLEDNICI

Web preglednik omogućava pregledavanje web stranica na „Web“-u. No kako arhitektura semantičkog weba ne koristi HTML, standardni format kojim se tradicionalni web preglednik koristi, stvarani su preglednici semantičkog weba koji traže izričito RDF podatke sa Web servera.

Napravljene su tri vrste preglednika: preglednik kao samostalna aplikacija, preglednik kao dodatak na postojeći web preglednik, te preglednik unutar web preglednika (web stranica). Funkcioniraju na dva načina prikaza rezultata:

- Preglednici bazirani na tekstu: Ovi preglednici koriste tablice i liste kako bi prikazali entitete, svojstva i veze.
- Vizualni preglednici: Preglednici koji koriste grafove, slike, mape kako bi prikazali povezane podatke

5.1 Disco

Disco preglednik je preglednik koji se instalirao kao dodatak na web preglednik. Pretraga se izvršavala na način da mu kao upit zadao URI „a“ koji se pretraživao. On bi zatim uključio svaki URI „b“ sa kojim je traženi URI povezan po svojstvu „rdfs:seeAlso“, tj. svojstvom koje označava da bi se mogle naći dodatne informacije o URI „a“ („a“ rdfs:seeAlso „b“). Pretraga bi nastavila traženje trojaca gdje god se URI „a“ spominje, te se tako stvorio graf kojeg bi Disco prikazivao u tekstualnom obliku.

5.2 Sig.ma

Sig.ma preglednik je dohvaćao povezane podatke sa više izvora omogućavajući navigaciju stranicama. Nudi prikaz svih tvrdnji koje se nalaze na dohvaćenim stranicama, te daje

mogućnost pretrage po tekstu i URI-ju, što je prednost na prema preglednicima koji pružaju pretragu samo po URI-ju. Dobivši upit pregledao se Sindice za resursom kojeg upit označava, utvrdila svojstva i vrijednosti te prikazali rezultati.

5.3 LodLive

LodLive je vizualni preglednik koji je u funkciji, a funkcionira na način da stvara RDF graf uz koristeći RDF i SPARQL. Koristi se na način da se kao upit koristi URI od kojeg se stvori početni čvor, te od kojeg se poveznicama otvaraju novi čvorovi čineći vidljivi RDF graf.

6 ZAKLJUČAK

Tema je poznata već godinama ali semantički web je tek posljednjih godina postao u upotrebi i nezaobilazna tema u svijetu weba. Tehnologije se još razvijaju budući da ovo nije prvobitna ideja vizionara, uzevši u obzir prevelik broj poveznica. Fokusiranjem kreatora stranica na označavanje podataka, dobije se informacija korisna tražilicama i ljudima. Tražilice uz pomoć web pauka posjećuju stranice koje sadrže strukturirane podatke i indeksiraju ih u svoju bazu. Pretragom korisnika podaci se povlače iz baze i poveznicama prema drugim podacima tvore korisnu pretragu.

Od strane Google-a stvoren je alat kojim se može testirati kako su podaci na web stranici strukturirani, alat se zove „Structured Data Testing Tool“, a testira HTML dokumente koji su označeni mikropodacima sa schema.org, RDFa, RDFa Lite i JSON-LD (eng, *JavaScript Object Notation for Linked Data*). JSON-LD predstavlja označavanje podataka u „javascriptu“. Sam Googleov alat je vrlo jednostavan za korištenje i ubacivanjem URL-a stranice koju bi htjeli pogledati vidimo sve oznake koje su korištene u njoj.

U radu je dan naglasak na razliku pretraga između tradicionalne i semantičke pretrage baš iz načina kojim nam semantička pretraga daje bolje rezultate pretraga, te načina kojim možemo proširiti znanje slijedeći povezane podatke.

Biti će zanimljivo vidjeti semantičke odgovore kompleksnih pitanja na hrvatskom jeziku, te daljnje razvijanje semantičkih tehnologija.

Jedan problem semantičkog weba su programeri, odnosno ljudi, koji su lijeni, lažu ili ne znaju. Pokušavati će pronaći najlakši način da varaju kako bi bili bolje pozicionirani i da omoguće računalima da komuniciraju umjesto da provedu vrijeme proučavajući i učeći RDF i OWL način strukturiranja.

Semantički web je nadopuna na tradicionalni u kojemu su podaci jasno strukturirani gdje računala mogu sama izvlačiti, raspoznavati i razlikovati podatke jedne od drugih i povezati ih sa vanjskim podacima. Semantičke tražilice pronalaze te podatke te ih spremaju, da nam se omogući brže pretraživanje i bolji rezultati.

7 LITERATURA

a) KNJIGE

1. Liyang Yu, (2011): A developer's guide to the semantic web, Springer, Heidelberg, Njemačka
2. Dean Allemang, James Hendler, (2011): Semantic Web for the Working Ontologist, Second Edition. Elsevier, Waltham, Massachusetts, SAD
3. Steffen Staab, Rudi Studer, (2004): Handbook on Ontologies, Springer, Heidelberg, Njemačka
4. Pascal Hitzler, Markus Krotzsch, Sebastian Rudolph, (2010): Foundations of Semantic Web Technologies, Chapman & Hall, Boca Raton, SAD
5. Bob DuCharme, (2011): Learning SPARQL, O'Reilly Media, Sebastopol, Kalifornija, SAD

b) PUBLIKACIJE

1. Christian Bizer, Tom Heath, Tim Berners Lee, (2009): Linked Data - The Story So Far
2. G. Sudeephi, G.Anuradha, Prof. M. Surendra Prasad Babu, (2012): A Survey on Semantic Web Search Engine
3. R.Guha, Rob McCool, E. Miller, 2003: Semantic Search
4. R. Yus, V. Mulwad, T. Finin, E. Mena: Infoboxer – Using Statistical and Semantic Knowledge to Help create Wikipedia Infoboxes

c) INTERNET IZVORI

1. Statistika internet korisnika: http://www.internetlivestats.com/internet-users/#trend, posjećeno (srpanj 2015)
2. Podaci na internetu u minuti: http://www.visualnews.com/2012/06/19/how-much-data-created-every-minute/ , posjećeno (srpanj 2015)
3. Linked Data: http://linkeddata.org/ , posjećeno (srpanj 2015.)
4. Dataversity: http://www.dataversity.net/end-support-sindice-com-search-engine-history-lessons-learned-legacy-guest-post/ , posjećeno (kolovoz 2015)
5. SPARQL: http://www.w3.org/TR/rdf-sparql-query/ , posjećeno (srpanj 2015)
6. Ontologija: http://www.w3.org/TR/vocab-org/ , posjećeno (srpanj 2015)

7. Seo By Sea: http://www.seobythesea.com/2013/09/google-hummingbird-patent/, posjećeno (kolovoz 2015)
8. W3C Frequently Asked Questions: http://www.w3.org/2001/sw/SW-FAQ, posjećeno (srpanj 2015)
9. Search engine journal: http://www.searchenginejournal.com/semantic-search-engines/9832/, posjećeno (rujan 2015)
10. Linked Data Tools: http://www.linkeddatatools.com/semantic-web-basics, posjećeno (srpanj 2015)