Prilagodba aplikacije pisane u jeziku Java temeljene na obrascu model-izgled-kontroler za semantički web

Rovan, Lidia

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Lidia Rovan

Prilagodba aplikacije pisane u jeziku Java temeljene na obrascu model-izgled-kontroler za semantički web

DOKTORSKA DISERTACIJA

Zagreb, 2010.

Doktorska disertacija je izrađena u **Zavodu za primijenjeno računarstvo Fakulteta elektrotehnike i računarstva u Zagrebu.**

Mentor: **Prof.dr.sc. Mirta Baranović**

Disertacija ima 272 stranice.

DISERTACIJA BR.:

Povjerenstvo za ocjenu doktorske disertacije:

1. Dr.sc. Damir Kalpić, redoviti profesor
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva
2. Dr.sc. Mirta Baranović, izvanredna profesorica
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva
3. Dr.sc. Mladen Varga, redoviti profesor
   Sveučilište u Zagrebu Ekonomski fakultet

Povjerenstvo za obranu doktorske disertacije:

1. Dr.sc. Damir Kalpić, redoviti profesor
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva
2. Dr.sc. Mirta Baranović, izvanredna profesorica
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva
3. Dr.sc. Mladen Varga, redoviti profesor
   Sveučilište u Zagrebu Ekonomski fakultet
4. Dr.sc. Bojana Dalbelo Bašić, redovita profesorica
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva
5. Dr.sc. Zoran Skočir, redoviti profesor
   Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva

Datum obrane disertacije: 25. ožujka 2010. godine

"Nitko ne može odsvirati simfoniju sam. Za simfoniju treba cijeli orkestar"
Halford. E. Luccock

Zahvaljujem svima koji su mi pružali potporu tijekom izrade ove disertacije i koji sa mnom dijele ostvaren uspjeh.

SADRŽAJ

1.Uvod

U znanosti nije toliko bitno doći do novih činjenica, koliko otkriti nove načine razmišljanja o njima.

William Bragg

Web je jedan od najbrže prihvaćenih tehnoloških izuma i utjecao je na ljudski život više nego ijedna druga tehnologija u bližoj povijesti. Iako su načela na kojima se web temelji ostala ista od njegovog samog početka, izgled se izrazito promijenio. Inicijalno zamišljen kao sustav za distribuiranje statičkog sadržaja, postao je dinamičko okruženje u čijem stvaranju mogu sudjelovati i sami korisnici.

Razvoj web tehnologija omogućio je da web postane platforma za sofisticirane aplikacije visoke razine složenosti. U okviru web inženjerstva razvile su se metodologije koje propisuju kako razviti pouzdane aplikacije koje obećavaju njihov dug životni vijek. Jedna od preporuka web inženjerstva je korištenje oblikovnog obrasca model-izgled-kontroler. Aplikacija arhitekture koja slijedi navedeni obrazac posjeduje karakteristike lakog održavanja i proširivosti te stoga obećava njen dugi životni vijek.

Semantički web je zamišljen kao rezultat evolucije sadašnjeg weba – tehnološkog fenomena. U semantičkom webu u središtu su podaci i veze među njima. Njegov razvoj traje duže od očekivanog i, iako sve tehnologije koje bi omogućile da web zaživi na semantičkoj razini još nisu razvijene, temelj je postavljen. Semantički web je proširenje današnjeg weba u kojem je svakoj informaciji dano dobro definirano značenje što omogućava računalima da razumiju sadržaj, rade zaključke nad njim i na taj način pružaju veću potporu korisnicima.

1.1. Motivacija

Ideje i načela na kojima se temelji web i semantički web sasvim su različita. Samim time i realizacije u takvim okruženjima imaju međusobna odstupanja. Cilj je ovog istraživanja istražiti mogućnosti prilagodbe postojećih web aplikacija temeljenih na obrascu model-izgled-kontroler za okruženje semantičkog weba.

Kako bi se mogao predložiti način prilagodbe postojećih web aplikacija semantičkom webu, prethodno je potrebno obaviti pregled područja weba, kao i razvoja Java web aplikacija, da bi se utvrdili standardi kojima bi web aplikacije trebale udovoljavati kako bi bile konkurentne i zadovoljile potrebe današnjeg korisnika. Dalje, potrebno je napraviti i pregled područja semantičkog weba s ciljem

da se ocijeni stupanj dostignuća tehnologija na tom području i procijeni mogućnost njihovog korištenja pri izradi moderne web aplikacije.

Također, potrebno je provesti analizu postojećih semantičkih web rješenja, kako bi se identificirale osnovne karakteristike semantičkih web aplikacija, funkcionalnosti koje takve aplikacije implementiraju te kako bi se prepoznali oblikovni obrasci implementacija takvih funkcionalnosti. Na temelju provedene analize semantičkih web rješenja i rezultata pregleda područja weba i semantičkog weba mogu se predložiti arhitekture aplikacija semantičkog weba.

Prepoznate funkcionalnosti semantičkih web aplikacija čine motive za njihovu izgradnju, pa tako i za implementiranje semantičkih tehnologija u već postojeće web aplikacije. Kako bi se omogućila što lakša prilagodba postojećih aplikacija za okruženje semantičkog weba, potrebno je definirati i verificirati proces prilagodbe postojećih web aplikacija za semantički web.

1.2. Znanstveni doprinosi

U okviru ove disertacije predstavljaju se znanstveni doprinosi u području inženjerstva semantičkih web aplikacija: (1) na osnovu provedene analize semantičkih web rješenja obavit će se kategorizacija semantičkih web aplikacija te će se dati definicija za svaku od njih, (2) za prepoznate kategorije semantičkih web aplikacija predložit će se arhitektura realizirana Java programskim jezikom temeljena na MVC obrascu, (3) analizirat će se nedostaci postojećih semantičkih web rješenja te će se kao odgovor na uočen nedostatak predložiti sustav za upravljanje identitetom u okruženju semantičkog weba, (4) provest će se identifikacija karakteristika web aplikacija na temelju kojih će se predložiti metodologija procesa prilagodbe, (5) predstavit će se cjelokupni proces prilagodbe postojećih web aplikacija okruženju semantičkog weba uz detaljno objašnjenje svih aktivnosti koje propisuje metodologija prilagodbe i (6) primjenom procesa prilagodbe nad postojećim aplikacijama dat će se ocjena primjenjivosti predložene metodologije procesa prilagodbe.

1.3. Struktura disertacije

U 2. poglavlju dan je pregled područja razvoja Java web aplikacija koji pruža teorijsku osnovu aplikacija koje čine predmet ovog istraživanja, dok 3. poglavlje sadržava pregled razvoja osnovnih tehnologija nad kojima se zasniva semantički web. U 4. poglavlju razjašnjava se nastanak trenutno aktualnih pravaca istraživanja

semantičkog weba koja se slijede u ovom istraživanju. Dan je pregled tehnologija nastalih nad osnovnim tehnologijama semantičkog weba i metodologija njihovog korištenja u skladu s novim pravcima istraživanja. U 5. poglavlju iznose se rezultati analize semantičkih web rješenja provedene u svrhu otkrivanja i definiranja osnovnih kategorija i funkcionalnosti semantičkih web aplikacija. Poglavlje 6 sadrži prijedloge i razradu arhitekture prepoznatih kategorija semantičkih web aplikacija, kao i prijedlog arhitekture i realizacije sustava za upravljanje identitetom u okruženju semantičkog weba. U 7. poglavlju predložen je cjelokupni proces prilagodbe postojećih web aplikacija semantičkom webu te se daje i ocjena njegove primjenjivosti.

2.Razvojni put Java web aplikacija temeljenih na model-izgled-kontroler obrascu

2.1. Uvod u Java web aplikacije

Ideja je bila utipkati "login" na UCLA-u i vidjeti hoće li se riječ pojaviti na drugom računalu na sveučilištu u Stanfordu.

"Uspostavili smo telefonsku vezu između nas i momaka sa Stanforda,
utipkali smo L i pitali preko telefona:
"Vidite li L?"
"Da, vidimo L", došao je odgovor.
Utipkali smo O i pitali:
"Vidite li O?"
"Da, vidimo O."
Onda smo utipkali G i sustav je pao...

Ali revolucija je započela..."

Leonard Kleinrock

O webu se često priča kao o tehnološkom fenomenu. Brzina kojom je prihvaćen, kao i njegova rasprostranjenost predmet je brojnih rasprava i istraživanja. Da bi web postao ono što je danas - platforma za sofisticirane aplikacije, razvile su se brojne tehnologije koje su pomogle njegovom preoblikovanju. Aplikacije pisane programskim jezikom Java bile su među prvim rješenjima podrške kompleksnog poslovanja na webu.

2.1.1. Web

Tijekom posljednjeg desetljeća izgled weba drastično se promijenio iako su načela na kojima se temelji ostala nepromijenjena. Inicijalno je bio zamišljen kao sustav za distribuiranje statičkog sadržaja koji se sastojao od jednostavnog teksta i slika. Do danas se web prilično odmaknuo od svoje inicijalne svrhe, postao je platforma za sofisticirane aplikacije na najvišoj razini poslovanja. Izumi koji su najviše utjecali na razvoj weba i njegovu široku prihvaćenost od strane korisnika su redom: vizualni preglednik (Mosaic, 1993.), pretraživači (webCrawler i Lycos, 1994.), preglednik s podrškom za JavaScript (Netscape 2.0B3, 1995.), blog (1998.), Wikipedia (2001.), razvojna tehnologija Asynchronous JavaScript and XML (Ajax) i web aplikacije za društvenu interakciju kao što je Facebook (2005.). Kako bi navedeni izumi uopće mogli zaživjeti trebala je postojati odgovarajuća infrastruktura

koja bi ih podržala. Jedan od osnovnih preduvjeta široke prihvaćenosti weba svakako je njegova laka dostupnost. Konstantan tehnološki napredak same mrežne infrastrukture – interneta, kao i stalno snižavanje cijena usluge interneta, omogućilo je da danas gotovo svako kućanstvo ima stalnu internet vezu. Web je jedan od najbrže prihvaćenih tehnoloških izuma i utjecao je na ljudski način života više nego ijedna druga tehnologija u bližoj povijesti [Singh2002].

Razlozi prihvaćanja, odnosno odbijanja, novih informacijskih tehnologija dugo su bili predmet istraživanja. Davis 1989. identificira čimbenike prihvaćanja tehnološke inovacije [Abel2007, Davis1989]. Prema Davisu osnovni pokazatelji prihvaćanja nove tehnologije su ***percipirana korist*** i ***percipirana jednostavnost korištenja od strane korisnika***. Gotovo od svog samog početka web neupitno zadovoljava oba postavljena kriterija. Pronalaženje potrebnih informacija sigurno predstavlja najveću i najrašireniju korist weba [Lederer2000]. Pretraživači (npr. Google) danas su najvažniji alat za pronalaženje informacija na webu, te je njihova pojava imala najveći utjecaj na poboljšanje korisnosti weba. JavaScript tehnologija uvela je dinamičnost u web i na taj je način napravljen prvi korak prema poboljšanju jednostavnosti i većoj ugodnosti korištenja weba. Veliku razliku u percepciji korištenja čini Ajax koji je omogućio da se konačno premosti razlika u funkcionalnosti između klasičnih klijentskih (stolnih) aplikacija i web aplikacija, odnosno da se postigne interaktivnost web aplikacija koja prije nije bila moguća. Izumi kao što su Blog i Wikipedija otvorili su mogućnost aktivnog sudjelovanja korisnika, njegovog doprinosa u stvaranju sadržaja na webu, stoga korisnik više nije nužno samo njegov konzument, već je i autor.

S blogovima započinje nova generacija weba, poznatija pod nazivom *"društveni web"*. Tijekom prethodnih godina veliku popularnost postižu upravo društvene aplikacije, koje omogućuju sudjelovanje korisnika u društvenim mrežama. Ta nova generacija weba donosi sa sobom i novu dimenziju percepcije weba od strane korisnika. Više nisu u središtu dokumenti, njihov sadržaj i veze među njima, nego veze između ljudi i zajednica te njihova međusobna komunikacija. Neočekivana velika popularnost društvenih aplikacija (npr. Facebook) dovodi u pitanje metodologije koji su se do sad koristile pri evaluaciji upotrebljivosti aplikacija. Rasvjetljavanje razloga popularnosti društvenih aplikacija trenutno je predmet brojnih istraživanja [Hart2008].

2.1.2. Web aplikacije

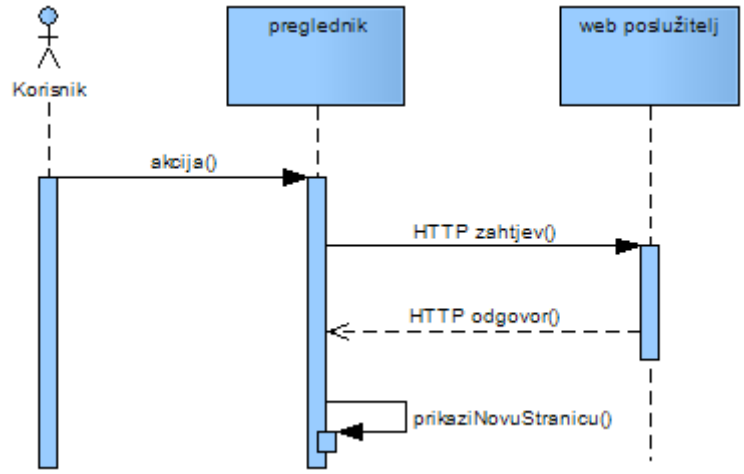
Postoje različita shvaćanja o tome kakva bi se sve programska rješenja mogla svrstati pod web aplikacije [Abel2007, Conallen1999]. Conallen [Conallen1999] definira web aplikaciju kao:

> ***Definicija 2.1:*** *„web aplikacija je web sustav koji implementira poslovna pravila, te u kojem akcija korisnika može promijeniti stanje poslovanja obuhvaćenog sustavom prema definiranim poslovnim pravilima.*“

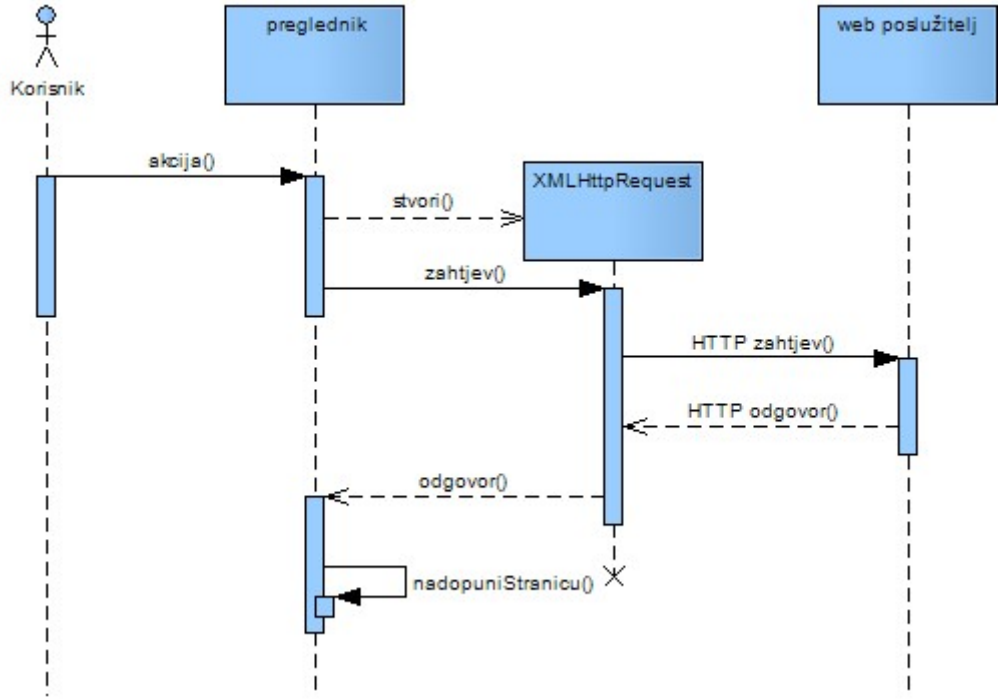
Važno je primijetiti da definicija izuzima web sjedišta (eng. web site), prethodnike današnjih web aplikacija, koje prvenstveno za cilj imaju pružanje informacija, ali ne i obavljanje funkcija. Web sjedište čini skup web stranica (hipertekstovnih dokumenata) na čiji sadržaj korisnik ne može utjecati [Baresi2000].

Današnje web aplikacije mogu se podijeliti u dvije skupine: ***klasične web aplikacije*** i ***bogate internet aplikacije***. Klasične web aplikacije su aplikacije u kojima se na svaku akciju korisnika iz korisničkog sučelja (preglednika) odvije HTTP (Hypertext Transfer Protocol) komunikacija između klijenta i web poslužitelja, te se na svaku takvu akciju korisnika ponovno učitava i prikazuje cijela web stranica. HTTP je samo jedan od protokola na internetu, a namjena mu je dohvaćanje i objavljivanje hipertekstovnih dokumenata. Na temelju obrađenog HTTP zahtjeva poslužitelj klijentu vrati odgovarajući hipermedijski sadržaj koji se prikaže korisniku na ekranu [Paulson2005]. Slika 1 prikazuje kako se slijedno odvija opisana komunikacija između klijenta i poslužitelja u klasičnim web aplikacijama. Iako s tehničke strane nema očitih zamjerki na ovakav scenarij, primjedbe korisnika na korištenje aplikacije su s vremenom postajale sve veće. Kao što je već istaknuto, korisnikova percepcija korištenja aplikacije jako je važna za njenu prihvaćenost. Potreba da se za svaku malu promjenu ponovno učita i prikaže stranica ograničava interaktivnost i dinamiku, te na taj način uskraćuje zadovoljstvo korištenja. Naravno, navedeno vrijedi samo za kompleksne aplikacije. Ukoliko je problematika pokrivena aplikacijom jednostavnijeg karaktera i može se realizirati linearnim slijedom aktivnosti, klasična web aplikacija je više nego zadovoljavajuća. Razvijanje aplikacija korištenjem tehnologije Ajax, koja omogućava odvijanje asinkrone komunikacije klijent-poslužitelj i osvježavanje samo dijela stranice prikazane na ekranu, rezultiralo je stvaranjem visoko interaktivnih web aplikacija koje se nazivaju: *bogate internet aplikacije* (eng. Rich Internet Applications - RIA). Slika 2 prikazuje kako se slijedno odvija asinkrona komunikacija u web aplikacijama. Zbog popularnosti koje su te aplikacije stekle kod korisnika, nastupio je zaokret u razvoju

web aplikacija, te se klasične web aplikacije razvijaju u sve manjoj mjeri [Mesbah2008].



Slika 1 Komunikacija klijent – poslužitelj (klasična web aplikacija)



Slika 2 Komunikacija klijent - poslužitelj (ajax zahtjev)

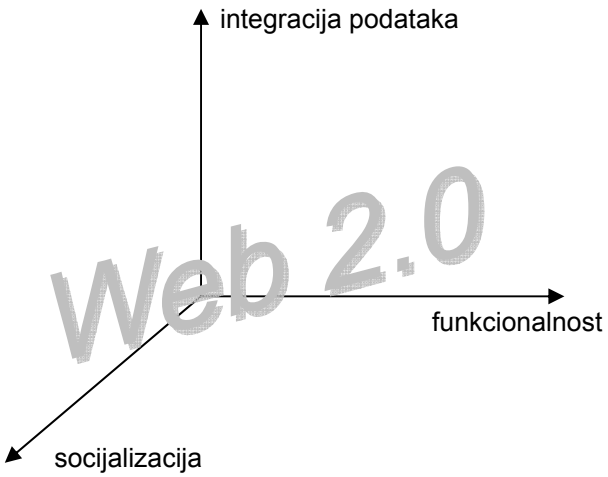
2.1.3. Web 2.0

Vrlo popularan i učestalo spominjan pojam Web 2.0 [Vossen2007] nema čvrste granice glede tog što on točno obuhvaća te se zbog tog veže dosta polemike uz njegovo značenje. Samo jedan od otvorenih problema, a sigurno najvažniji u

kontekstu ovog istraživanja, je taj što semantički web nije obuhvaćen pod pojmom Web 2.0, iako je njegov razvoj počeo davno prije. Kao rješenje predlagalo se uvođenje pojma Web 3.0 koji bi zapravo predstavljao fuziju semantičkih tehnologija i arhitekture današnjih modernih aplikacija [Vossen2007]. Budući da je pojam Web 2.0. jako zastupljen u svoj relevantnoj literaturi vezanoj uz web, od svakog rada koji nastoji pratiti trendove razvoja weba i aplikacija na webu očekuje se da se izjasni kako se uklapa u Web 2.0 koncept. Mogu se razlučiti tri osnovne dimenzije današnjeg weba:

- integracija podataka – razvile su se brojne metode za korištenje i kombiniranje podataka iz različitih izvora podataka u svrhu dobivanja novih informacija; aplikacije koje implementiraju te metode poznate su pod nazivom „mash-up“
- socijalizacija – korisnik sudjeluje u stvaranju sadržaja na webu; primjeri su društvene aplikacije, blogovi, označavanje sadržaja, itd…
- funkcionalnost – implementacija funkcionalnosti koje su prije bila karakteristika samo stolnih aplikacija, dakle realizacija bogatih internet aplikacija

Web 2.0 se upravo temelji na te tri dimenzije današnjeg weba, pa se može ilustrirati „Web 2.0 prostor“, kao što prikazuje slika 3. Web 2.0 aplikacija je ona aplikacija koja integrira činitelje svake od dimenzija.



Slika 3 Prostor Web 2.0

2.1.4. Java web aplikacije

Web aplikacije se razlikuju po složenosti poslovanja kojeg pokrivaju. Mogu se sastojati od svega nekoliko jednostavnih slučajeva uporabe ili mogu sadržavati brojna i pri tom kompleksna poslovna pravila. Prve klasične web aplikacije

realizirane su na temelju CGI (Common Gateway Interface) tehnologije koja je standardizirala način na koji web poslužitelj vrši interakciju s manjim programima (CGI skripta) zaduženim za procesiranje HTTP zahtjeva i davanje odgovarajućih HTTP odgovora [Doyle2005, Singh2002]. Ti programi (skripta) nalaze se na poslužitelju i mogu biti pisani u raznim programskim jezicima. Često su se za njihovu realizaciju koristili interpretativni jezici kao Perl. No, aplikacije razvijane prema CGI tehnologiji, rastom poslovanja kojeg se htjelo implementirati na webu, pokazale su probleme s *performansama*, *skalabilnošću* i ono najvažnije, s *održavanjem*. Da takve aplikacije nisu skalabilne već se pokazalo s rastom broja njihovih korisnika, pa tako i većim brojem istovremenih pristupa aplikaciji, jer se na svaki zahtjev korisnika stvara novi proces u operacijskom sustavu, što znatno troši memorijske i procesorske resurse [Doyle2005, Wu2000]. Važan čimbenik uspješnog održavanja aplikacija mogućnost je odvajanja zaduženja (npr. prezentacijski dio od poslovne logike) [Singh2002]. Budući da CGI program kao rezultat vraća potpun sadržaj web stranice, nužno je kombinirati kôd zadužen za sam izgled hipertekstovne stranice zajedno s kôdom za prikaz dinamičkog sadržaja. U skladu s navedenim, a kao što je i argumentirano u radu [Jazayeri2007], takav pristup otežava održavanje imalo kompliciranijih aplikacija. Ostali problemi, kao što su praćenje stanja aplikacije, sigurnost, provjera valjanosti unosa, pristup podacima (npr. bazi podataka), kao i rukovanje događajima, potpuno su prepušteni programerima. Istraživanje Wua [Wu2000] pokazalo je da CGI tehnologija može biti dobar odabir za aplikacije srednje veličine s malim brojem korisnika, ali da ipak složenije aplikacije s većim brojem korisnika performansama ne mogu parirati alternativnim tehnologijama koje su tada (2000. god.) već bile dostupne na tržištu.

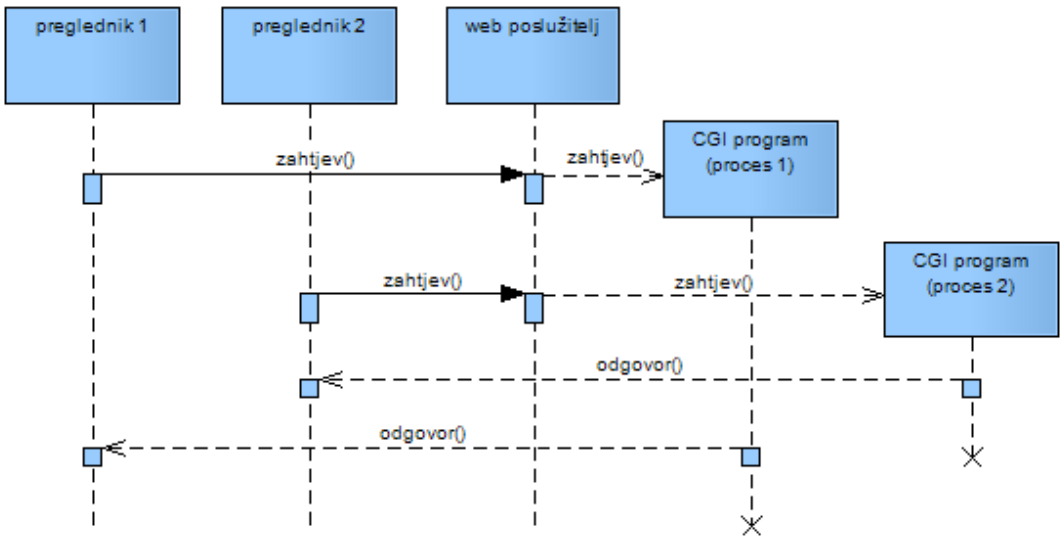
U okviru ovog istraživanja proučavat će se web aplikacije pisane u jeziku Java, koje su se razvile na temelju Java Servlet tehnologije, jedne od tehnologija koje su nastale u okviru pokušaja pronalaženja odgovora na probleme i ograničenja korištenja CGI tehnologije.

Java servleti i JavaServer Pages (JSP) komplementarne su tehnologije koje čine fundamentalan dio Java Enterprise Edition[1] (JEE) platforme [Brown2001]. JEE je namijenjen razvoju kompleksnih poslovnih aplikacija. Servlet je baš kao i CGI skripta zadužen za obradu HTTP zahtjeva i pružanje HTTP odgovora. Za razliku od CGI-a, prilikom pisanja kôda servleta osoba zadužena za razvoj programa nije opterećena

[1] s izlaskom verzije Jave 5 J2EE je promijenila naziv u JEE

svim detaljima implementacije, operacije niže razine su apstrahirane (putem servlet API-a) te se može više posvetiti implementaciji poslovnih pravila [Brown2001].

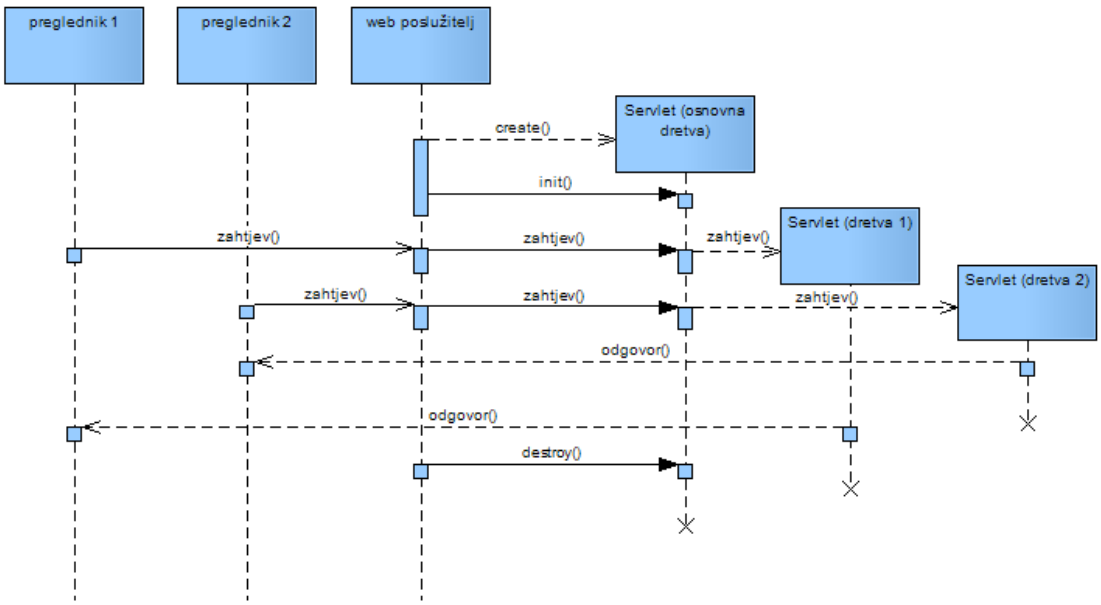
Kod izrade složenijih aplikacija vrijede sljedeći problemi primjene CGI tehnologije: *slabe performanse* i *skupo održavanje*. Upotrebom JEE rješavaju se oba navedena problema. Performanse JEE aplikacija su znatno poboljšane u odnosu na CGI aplikacije zbog činjenice da su servleti trajni (eng. persistent), što znači da je životni vijek servleta duži od jednog HTTP zahtjeva. Za razliku od CGI programa koji sa svakim HTTP zahtjevom stvara novi proces u memoriji, kao što prikazuje slika 4, HTTP zahtjevi upućeni servletu obrađuju se dretvama. U memoriji postoji samo jedna instanca servleta koja se kreira pri prvom HTTP zahtjevu koji pristigne određenom servletu ili ju se može instancirati i programski. Opisani proces prikazuje slika 5.



Slika 4 CGI - životni ciklus

Malo nakon JavaServleta na tržištu se pojavljuje i nova JEE tehnologija – JSP. Uvodi se kao odgovor na problem održavanja uslijed miješanja kôda potrebnog za prikaz i kôda za poslovna pravila, koji se do pojave JSP-a nalazio u cijelosti u servletu. Načelno, JSP obavlja istu zadaću kao i servlet, ali koristi drugačiju razvojnu paradigmu [Brown2001]. Dok su servleti zapravo Java klase pa, naravno, slijede i sintaksu klasa, JSP stranice čini sintaksa vrlo slična HTML-u (HyperText Markup Language) koja se automatski pretvara u servlete tijekom izvođenja. Sintaksa JSP-a idealna je za izradu dinamičkih web stranica, te dozvoljava razdvajanje prezentacijskog sloja od sloja poslovne logike budući se sastoji od čistog HTML-a

(kreira ga dizajner) i JSP oznaka kojima se umeće dinamički sadržaj (kreira ih Java programer). Servleti su se istakli između svih tehnologija koje su nastale kao odgovor na probleme CGI-a zahvaljujući *jednostavnosti*, *lakoj proširivosti*, *efikasnosti* i *dobrim performansama* [Brown2001]. Sam programski jezik Java ima veliku vrijednost za web budući da je neovisan o platformi, pa se može izvoditi na raznim konfiguracijama poslužitelja.



Slika 5 Servlet - životni ciklus

2.2. Održavanje i primjeri dobre prakse

> ***Čim smo počeli programirati, otkrili smo suprotno očekivanjima da nije tako lako napisati program točno iz prvog pokušaja. Bilo je neminovno da se (debugging) izdvoji kao posebna disciplina. Mogu se sjetiti točnog trenutka kad sam shvatio da ću od tog trenutka veći dio svog života provesti tražeći pogreške u vlastitim programima.***
>
> **Maurice Wilkes, 1949**

Želja za podržavanjem sve većeg opsega poslovanja na webu, potakla je razvoj sve složenijih aplikacija. Uskoro je postalo očito da metodologije razvoja web aplikacija ne udovoljavaju potrebama. Javila se nova disciplina, web inženjerstvo, koja je sa sobom donijela niz smjernica kako razviti aplikacije koje posluju na najvišoj razini i koje obećavaju dug životni vijek aplikacija.

2.2.1. Kriza weba

Programsko inženjerstvo (eng. software engineering) nastalo je 1968. kao odgovor na probleme u razvoju sustava koji su otkriveni 60-ih godina prošlog stoljeća [Jazayeri2007]. Naime, upravo tada prvi projekti koji su za cilj imali razvoj programske potpore (informacijskih sustava) kao podrške poslovanju na najvišoj razini počeli su nizati neuspjehe. Postalo je jasno da tadašnja tehnologija i metodologija za razvoj monolitnih sustava, jedina u to vrijeme dostupna, ima brojna ograničenja. Takvi sustavi bili su teško proširivi, jako teški za održavanje te su rijetko zadovoljavali uvjete naručitelja, ako bi uopće i bili isporučeni [Jazayeri2007].

> ***Programsko inženjerstvo*** *omogućava sistematičan pristup izradi programske potpore temeljen na modularnom dizajnu, standardiziranim komponentama programske potpore i definiranim procesima, a sve u cilju ostvarenja lako održavane, pouzdane, efikasne i dobro prihvaćene (upotrebljive) aplikacije* [Jazayeri2007].

Razvoj web aplikacija, od nestrukturiranih do strukturiranih, išao je sličnim povijesnim tijekom kao i razvoj klijentskih aplikacija, ali ipak, zahvaljujući prethodnom iskustvu, puno većom brzinom. Dio puta opisan je u prethodnom poglavlju, od CGI-a do JEE-a. Iako JEE pruža infrastrukturu za izradu pouzdanih aplikacija na najvišoj razini poslovanja, odgovornost o načinu korištenja te infrastrukture na osobi je zaduženoj za razvoj. Budući da su u samim počecima web

aplikacije bile dio promjenjivog okruženja, činilo se nepotrebnim specificirati zahtjeve i raditi razvoj na prethodno izgrađenom modelu, jer se evolucija aplikacije i onako očekivala pa je razvoj uglavnom bio "ad-hoc". Kako su web aplikacije postajale sve ozbiljnije, a nesistematičan način razvijanja aplikacija nosi visok rizik stvaranja pogrešaka, promjene u metodologiji razvoja bile su nužne [Pressman2000]. Može se uočiti sličnost s problemima koji su nastupili pri širenju poslovanja ranije spomenutih monolitnih sustava. Kao i u slučaju monolitnih aplikacija situacija se znatno poboljšala primjenom prakse iz programskog inženjerstva. Razlog zašto se dugo čekalo prije nego što se praksa iz programskog inženjerstva, već dokazana na klijentskim aplikacijama, primijenila i na razvoj web aplikacija je taj što se nije vjerovalo da je taj pristup pogodan za brzi razvoj web aplikacija koji je tada tržište nametalo [Pressman2000]. Kasnije, u sklopu nove discipline *web inženjerstva* nastale su metode i primjeri dobre prakse specifični za web aplikacije. Taj period između nestrukturiranog razvoja web aplikacija i masovnije primjene inženjerstva, koji je rezultirao nepouzdanim web aplikacijama, naziva se i *krizom weba* [Ginige2001].

2.2.2. Web inženjerstvo

Općenito, inženjerstvo se definira kao proces fizičke implementacije sustava polazeći od izrade apstrakcija na visokoj razini i logičkih modela neovisnih o samoj implementaciji [Demeyer2003]. Web inženjerstvo je inženjerstvo primijenjeno na web aplikacije. Iako se misli da je web inženjerstvo klonirano programsko inženjerstvo, ipak ono sadrži nove pristupe, načela, metode, alate, tehnike i smjernice kojima se mogu zadovoljiti specifičnosti web aplikacija [Ginige2001].

Važno je istaknuti da je razvoj web aplikacija znatno drugačiji od razvoja tradicionalnih (klijentskih) aplikacija, te njihov razvoj iziskuje brojne dodatne izazove. Tablica 1 prikazuje razlike između klijentskih i web aplikacija pa tako ukazuje na razloge za stvaranjem novog područja – web inženjerstva. Razvoj web aplikacija multidisciplinarno je područje [Ginige2001] te se potrebne discipline mogu podijeliti u tri osnovne dimenzije:

a) tehnološka dimenzija - web je heterogeno okruženje i zahtjeva se znanje brojnih tehnologija,
b) inženjerska dimenzija - znanje jezika modeliranja, obrazaca dizajna, programskih okvira i sl., i

c) estetska, etična, pravna, kognitivna, organizacijska i kulturna dimenzija - web aplikacije su više okrenute korisniku od tradicionalnih, važan je način prikaza sadržaja, široka dostupnost web aplikacija podiže brojna pravna pitanja, i sl.

Očigledno, žele li se zadovoljiti svi elementi web aplikacije, tijekom njene izrade trebalo bi konzultirati stručnjake svake od navedenih disciplina [Ginige2001]. U ovom istraživanju promatrat će se tehnološka i inženjerska dimenzija.

Tablica 1. Usporedba programskog i web inženjerstva

	klijentske aplikacije	**web aplikacije**
raspon korisnika	malen	velik
istovremen pristup	malen	velik
stručnost u modeliranju i razvoju	mali broj područja	velik raspon područja
ograničenja sklopovskog i programskog okruženja	specifična konfiguracija	mora podržati što više kombinacija
pridržavanje standardima	manje važno	jako važno
pravna i sigurnosna pitanja	manji opseg	veći opseg
korisničko sučelje	jednoobrazno	ovisi o kulturama, pristupnim uređajima i sl.

Većina zrelih metodologija web dizajna ističe važnost odvajanja dizajna podatkovnog modela, modela ponašanja i dizajna navigacije, odnosno sučelja. Jasno odvajanje zaduženja koja kôd mora ispuniti doprinosi kvaliteti dizajna, ključnoj za ostvarivanje ponovne iskoristivosti programskih modula, kao i za olakšavanje evolucije i održavanja aplikacije [Jacyntho2002].

Gledajući iz kuta web inženjerstva, tehnička pitanja s kojima se osobe zadužene za razvoj aplikacija susreću preklapaju se kod većine web aplikacija. Osnovni problemi mogu se podijeliti na: *izradu korisničkog sučelja*, *rukovanje podacima* te *kontroliranje akcija korisnika (navigacija)*. Ipak, nije se odmah prepoznalo da se

obrazac model-izgled-kontroler (eng. Model-View-Controller (MVC)), poznat iz razvoja klijentskih aplikacija, može uspješno primijeniti i na web aplikacije [Jazayeri2007]. Nakon što je prepoznat doprinos MVC-a web aplikacijama, razvijeni su brojni programski okviri za razvoj web aplikacija temeljeni upravo na tom obrascu (npr. Struts[2], tapestry[3], spring mvc[4]). Iako se od pojave MVC-a do danas izmijenio način modeliranja i razvijanja aplikacija, velik broj novijih trendova u razvoju aplikacija temelji se i dalje upravo na tom obrascu. Razliku čine programski okviri za razvoj bogatih internet aplikacija (ajax programski okviri). Od trenutno dostupnih, većina se ne temelji MVC obrascu. No, istraživanja pokazuju da je prepoznata potreba za takvim rješenjima pa se s pravom očekuje njihovo pojavljivanje u većem broju [Mycroft2008].

2.2.3. Programski okviri

Programski okviri (eng. frameworks) su ponovno iskoristive komponente i metode za njihovo povezivanje namijenjene korištenju u određenim slučajevima [Johnson1988]. Pružaju kostur sačinjen od programskih komponenti (općenitih) na temelju kojih se, njihovim prilagođavanjem specifičnom slučaju, mogu brzo i strukturirano izgraditi aplikacije. Usko su povezani s oblikovnim obrascima (eng. design pattern), odnosno sastoje se od uhodanih rješenja na najčešće probleme iz prakse. Dakle, nastoje implementirati najbolja rješenja koja su se pojavila u dosadašnjem radu na način da se omogući njihova laka primjena u različitim situacijama.

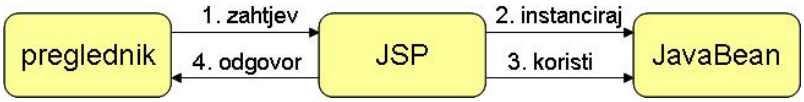
Razvoj programskih okvira izrazito je popularan u svijetu weba te postoji jako velik broj rješenja od kojih se ipak samo neka posebno ističu po svojoj zastupljenosti u praksi. Važno je primijetiti da se baš ti istaknuti programski okviri većinom temelje na MVC obrascu. Programski okviri za JEE aplikacije koji se često mogu susresti u praksi su: Spring, JSF (JavaServer Faces), Struts, Tapestry, Cocoon i Hibernate. U nedostatku istraživanja o java web programskim okvirima zaključak o trenutnoj zastupljenosti temelji se na opaženom broju korisnika interesnih grupa na webu. Uobičajeno je i korištenje kombinacije programskih okvira, obično po slojevima, npr. Struts za prezentacijski sloj, Spring za sloj poslovne logike i Hibernate za podatkovni sloj.

[2] http://struts.apache.org/
[3] http://tapestry.apache.org/
[4] http://www.springsource.org/

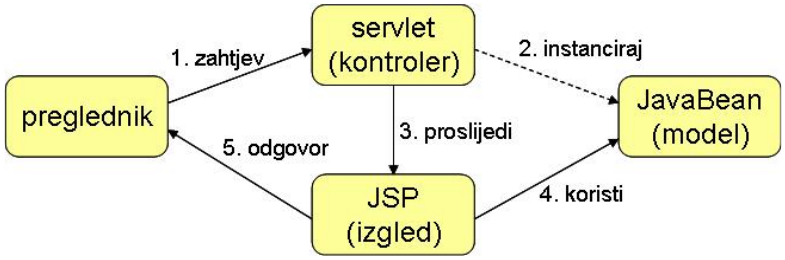
2.2.4. Arhitektura Java web aplikacija

Dvije arhitekture koje propisuju logičku razdvojenost, Model 1 i Model 2, bile su posebno dominantne u JSP zajednici (predložene u JSP specifikacijama). Slika 6 prikazuje arhitekturu Modela 1, dok slika 7 prikazuje arhitekturu Modela 2. Elementi arhitekture Servlet i JSP su prethodno objašnjeni te se ovdje daje detaljnije objašnjenje JavaBeana. JavaBean je Java klasa pisana po zadanoj konvenciji. Mora ispuniti sljedeće uvjete: sadržavati konstruktor vidljivosti *public*, za svako svojstvo klase mora postojati članska funkcija za čitanje vrijednosti takvog svojstva, kao i za promjenu njegove vrijednosti te se mora moći serijalizirati. Model 2 arhitektura temelji se na MVC obrascu, te je ona preporučena arhitektura uslijed lakog održavanja i proširivosti tako izgrađenih aplikacija [Singh2002]. Osnovna razlika između Modela 1 i Modela 2 u mjestu je gdje se odvija obrada zahtjeva klijenta. Sve ostale razlike proizlaze iz karakteristika koje propisuje MVC obrazac.

Iako se danas većina postojećih aplikacija ipak temelji na Modelu 2, kad se novi programeri tek susretnu s JEE platformom s velikom vjerojatnošću će proizvesti aplikaciju Model 1 arhitekture. Razvoj takvih aplikacija nema strmu krivulju učenja, brzo se dolazi do vidljivih rezultata koji pružaju programerima određenu motivaciju tijekom prihvaćanja nove tehnologije. Nedostaci ovakve arhitekture brojni su i vrlo brzo su se pokazali u praksi. Očit je nedostatak neiskorištavanje servleta kao zasebne komponente, što dovodi do isprepletenosti poslovne logike s kôdom potrebnim za prezentaciju, JSP stranice postaju prevelike, nema modularnosti i ponovne iskoristivosti te je nemoguće osigurati odvajanje uloga, odnosno paralelni rad više osoba.



Slika 6 Model 1 arhitektura



Slika 7 Model 2 arhitektura

2.2.5. Obrazac model-izgled-kontroler u JEE aplikacijama

Primjenom MVC arhitekture jasno su odijeljeni slojevi: model, izgled i kontroler. Kod JEE aplikacija svaki od tih slojeva realizira se primjenom međusobno različitih (specijaliziranih) tehnologija te je od posebnog značaja umijeće modeliranja odnosno pravilnog uočavanja pripadnosti slojevima određene logike. Da bi se što uspješnije mogla obavljati ta zadaća poželjno je biti upoznat s oblikovnim obrascima koji propisuju jasno odvajanje uloga i zaduženja koja kôd mora ispuniti.

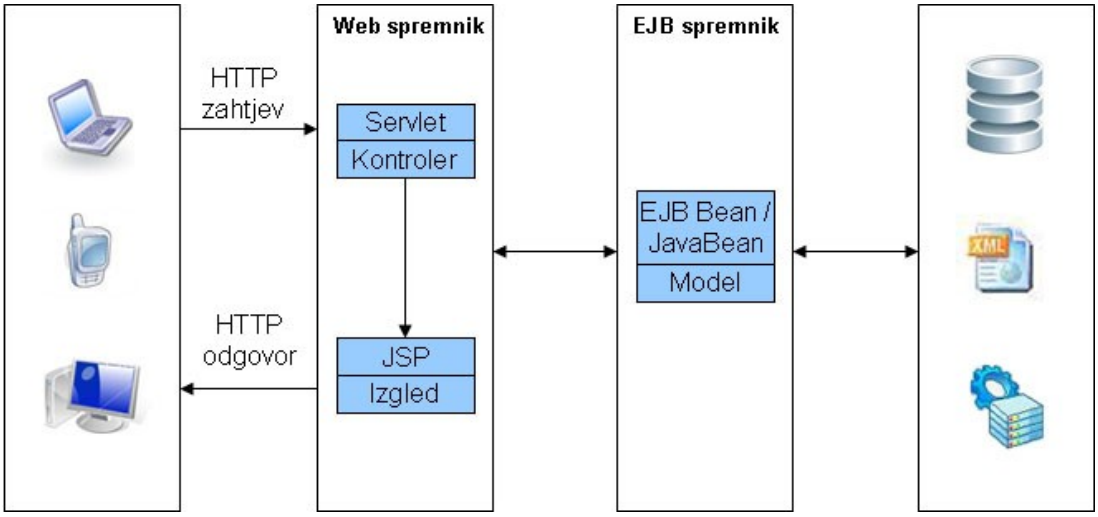
Bilo da se koristi neki od programskih okvira, neke dodatne tehnologije ili samo osnovni elementi JEE platforme realizacija se svodi na: [Falkner2004]

Izgled – realizira se upotrebom JSP-a

Model – skup JavaBeanova kojima je lako rukovati iz JSP-a (prikazivati dinamičke podatke)

Kontroler – servlet dizajniran tako da može primiti zahtjev klijenta i proslijediti odgovor; mjesto na kojem je zgodno implementirati sigurnost, praćenje rada aplikacija (eng. logging) i druge funkcionalnosti koje se odnose na cijelu aplikaciju

MVC je samo oblikovni obrazac koji ne propisuje kako fizički razdvojiti aplikaciju na slojeve, već samo logički. Budući da JEE platforma omogućava izgradnju višeslojnih (fizički) aplikacija, daje se i kraće objašnjenje uobičajene raspodjele MVC objekata po slojevima višeslojne arhitekture kao što prikazuje slika 8.



Slika 8 Višeslojna JEE arhitektura

Enterprise JavaBeans (EJB) je sloj koji nije posve neophodan. On se uvodi ovisno o opterećenju i složenosti koja se očekuje od aplikacije. U današnje vrijeme, postoje brojne tehnologije koje se mogu koristiti za realizaciju trajnog (eng. persistence)

sloja, od kojih su svakako najpopularniji alati za preslikavanje relacijskog modela u objektni model (eng. Object – Relational Mapping - ORM). U arhitekturama koje ne koriste EJB JavaBeanove, model se može nalaziti i u web spremniku. Najnovija specifikacija EJB-a (EJB 3.0) također dozvoljava pohranu EJB objekata u web spremniku.

Važno je istaknuti da primjer dobre prakse ne predstavlja uvođenje svih slojeva samo zato što to odabrana tehnologija dopušta, nego samo onih komponenti koje su nužne za tip problema koji se rješava aplikacijom.

Izrada aplikacija korištenjem različitih programskih okvira, koji pružaju gotova najnovija rješenja, danas je nužnost da bi se moglo konkurirati na tržištu. Zadaća je arhitekta sustava da odabere i poveže odabrane okvire na način da zadovoljavaju višeslojnu Model 2 arhitekturu, koja je uslijed svih promjena web tehnologija i dalje ostala preporučen primjer dobre prakse. Na taj način lakše je zamijeniti bilo koju ugrađenu komponentu s ciljem danas neophodne stalne modernizacije sustava.

3.Semantički web

Pitanje može li računalo misliti nije ništa manje zanimljivo od onog može li podmornica plivati.

Edsger W. Dijkstra

3.1. Od weba do semantičkog weba

Semantički web je vizija evolucije sadašnjeg weba. Ta je vizija započela ambicioznim primjerima mogućeg boljeg korištenja weba u kojoj računala još više pomažu korisniku. Osnovna je ideja sadržaj weba zapisati u takvom formatu koji bi omogućio njegovu jednostavnu integraciju, a koji bi pri tom bio razumljiv i računalima kako bi nad njim mogli izvoditi zaključke baš kao i čovjek. Semantički agenti bi procesirali sadržaj weba, međusobno ga povezivali i donesene zaključke iznosili korisniku. Da bi web mogao zaživjeti na semantičkoj razini potrebno je stvoriti tehnološko okruženje koje to omogućava. Razvijene su brojne tehnologije, svaka prilagođena posebnom tehničkom problemu s ciljem omogućavanja semantičkog weba te je svaka sposobna raditi u zajedništvu s ostalima. Unatoč dugotrajnom razvoju sve potrebne tehnologije još nisu razvijene, ali temelj je postavljen.

3.1.1. Početna vizija semantičkog weba

Ideju semantičkog weba, kao evolucije sadašnjeg weba, iznio je 2001. u članku za "Scientific American" [Berners-Lee2001] autor samog weba Tim Berners Lee. On opisuje semantički web kao proširenje sadašnjeg weba, u kojem je svakoj informaciji dano dobro definirano značenje što omogućava da računala 'razumiju' sadržaj i na taj način pruže veću potporu korisnicima. Iznosi motivirajuće primjere uporabe i predviđa da će se semantički web brzo realizirati. Primjeri se većinom temelje na agentima semantičkog weba koji bi imali za cilj sakupiti podatke iz različitih izvora podataka, procesirati ih i razmjenjivati s ostalim agentima [Berners-Lee2001]. Semantički web inicijalno je zamišljen kao sredstvo koje je u stanju podržati dnevne zadatke korisnika. Zamislimo sljedeću situaciju – korisnik sjeda za računalo i želi pronaći trgovinu sportske opreme koja nudi tenisice veličine 36. Dodatno, korisnik želi da se trgovina nalazi najviše dva kilometra od njegova mjesta stanovanja. Agentu (koji je zapravo računalni program) prosljeđuje se takav zahtjev te agent skuplja podatke i vraća dohvaćene rezultate. U rezultatima se nalazi nekoliko

trgovina, no ni u jednoj ponuđenoj nema tenisica po cijeni koja korisniku odgovara. Korisnik dodaje uvjet cjenovnog ranga koji mu je prihvatljiv te se opet konzultira agenta koji kreće u novo prikupljanje informacija. Scenarij do sad, čini se, nije tako teško realizirati, no sama ideja semantičkog weba zamišljena je još ambicioznije. Pretpostavimo sljedeći nastavak slijeda događaja - agent ne nalazi prihvatljive rezultate, no nalazi zanimljive rezultate koji malo odskaču od zadanih podataka i to samo u pogledu udaljenosti od mjesta stanovanja. Korisniku ne odgovara dulje putovanje budući da ima dogovoren sastanak, međutim agent može prepoznati kako je taj sastanak niskog prioriteta i korisniku nudi odgađanje sastanka za količinu vremena koja je potrebna kako bi se mogla obaviti kupnja tenisica. Iz primjera je vidljivo da se od semantičkog weba nije očekivalo da samo omogući interakciju računala bez ljudske intervencije, nego i dozu umjetne inteligencije. Zamišljeno je da se semantički web postavi nad osnovnim zaključcima, alatima i tehnikama proizašlim iz pedesetogodišnjeg istraživanja područja umjetne inteligencije [Shadbolt2006]. Pri iznošenju vizije semantičkog weba, njegov se autor nije zadržao na primjerima eksplicitnog korištenja weba, već je zamišljeno i komuniciranje samih uređaja – "nije teško zamisliti kako vaša mikrovalna pećnica prilagođena za web obavlja komunikaciju s web sjedištem proizvođača smrznute hrane u svrhu podešavanja parametara za spremanje hrane" [Berners-Lee2001].

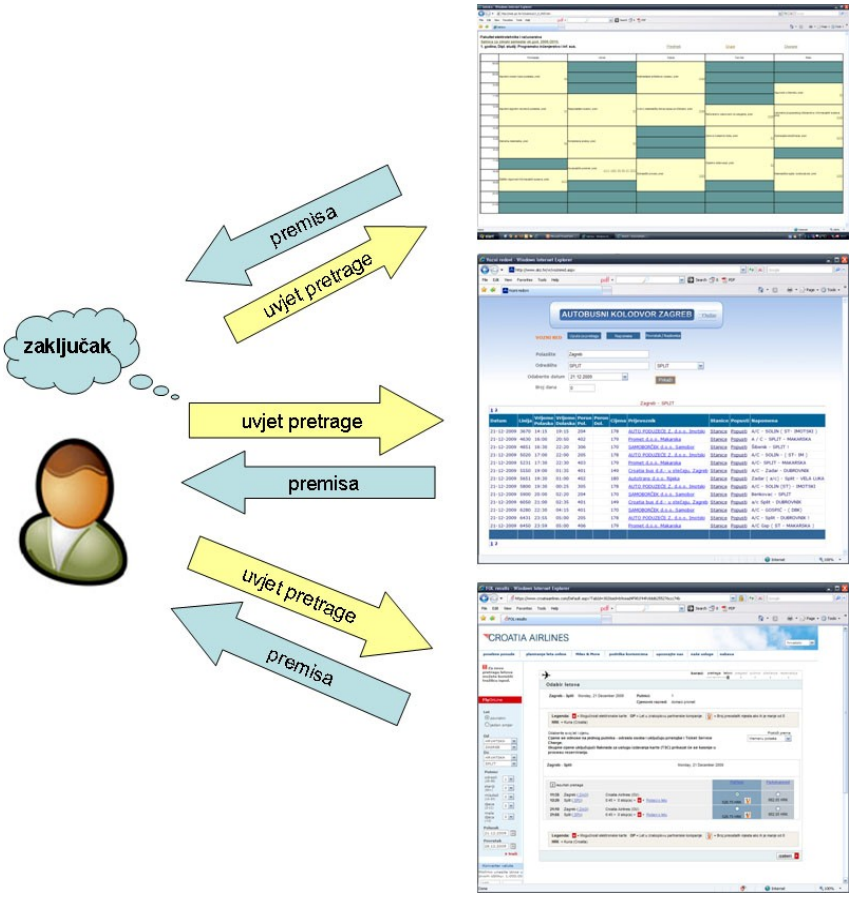
Osam godina nakon iznesene ideje, semantički web većinom se i dalje temelji na motivirajućim primjerima, još uvijek bez njihove primjene. Razlozi zbog kojih semantički web nije slijedio brzinu prihvaćanja samog weba, treba tražiti u onim istim razlozima koji su web učinili popularnim. Vrlo je malo scenarija na kojima se temelji osnovna vizija semantičkog weba u nekom obliku uistinu i zaživjelo, a ti rijetki primjeri koji su realizirani *ne mogu se pohvaliti jednostavnošću korištenja*. Da bi se opisani scenariji mogli i tehnički podržati treba ostvariti i odgovarajući infrastrukturu. Kako je za ostvarenje takve infrastrukture potrebno osmisliti i međusobno povezati brojne tehnologije od kojih svaka rješava jedan specifičan problem (npr. opisivanje podataka, logika, sigurnost, itd.), razvoj je potrajao. Dodatan problem predstavlja to što u tom periodu razvoj klasičnog weba nije stao, pa je kompatibilnost svih tehnologija gotovo nemoguće održati i nužne su stalne preinake. Takve okolnosti, utjecale su na to da realizirani primjeri nisu usmjereni prema korisniku, već su se više temeljili na realizaciji nad dijelovima infrastrukture, tako da se osim u znanstvenim krugovima, korisnost semantike na webu nije osjetila. Realizirane aplikacije, koje slijede početnu viziju semantičkog weba, a koje su ipak usmjerene

na krajnjeg korisnika, nisu mogle odgovoriti na njegove potrebe uslijed nedostatka sadržaja opisanih semantičkim web tehnologijama. Dostupan skup semantički opisanih podataka je ograničen te se zaključci doneseni nad tim skupom ne mogu nikako mjeriti s onima koje danas korisnik donosi sam koristeći podatke cijelog weba, što znači da semantička web rješenja još uvijek *ne pokazuju značajnu korist u usporedbi s tradicionalnim manualnim procesima donošenja odluka*.

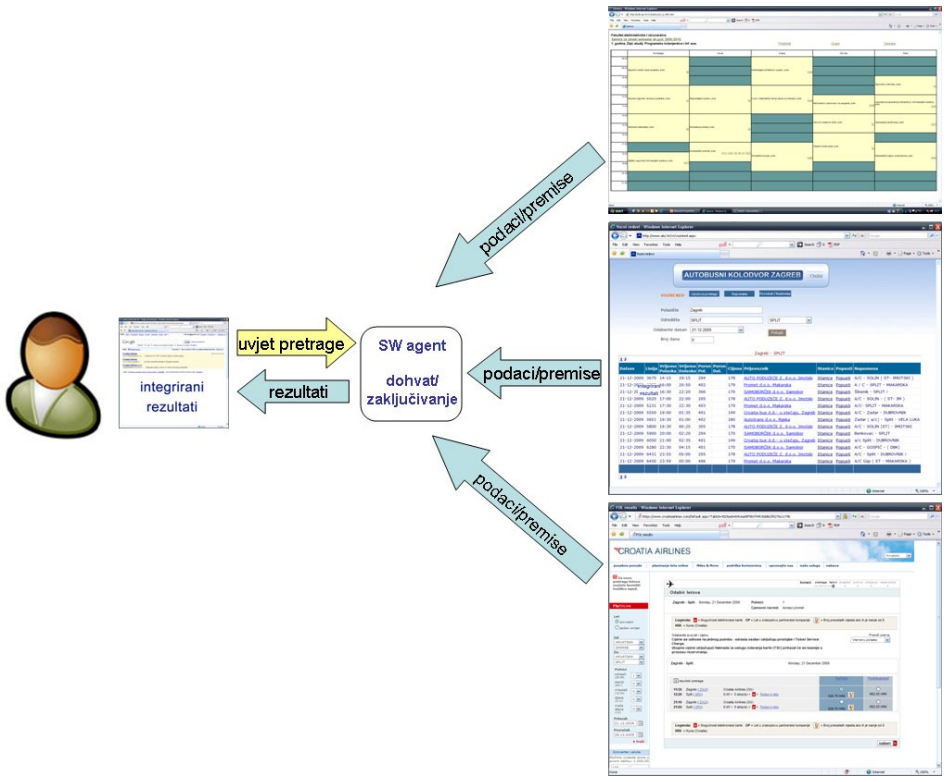
Od 2007. godine, stvaranjem novih pravaca istraživanja unutar područja semantičkog weba ipak se osjetio pomak u broju realizacija budući da se okrenulo krajnjem korisniku i većem stvaranju rješenja koja za svoje slučajeve uporabe dozvoljavaju nepotpunost rješenja (npr. nedovoljna sigurnost, manju izražajnost logike, itd.).

3.1.2. Temelj semantičkog weba

Da bi semantički web mogao zaživjeti, računala prvenstveno moraju imati pristup strukturiranim izvorima informacija te trebaju postojati pravila za zaključivanje uz pomoć kojih bi se mogli automatski izvoditi zaključci. Dakle, potrebne su tehnologije područja koje se proučava još od prije nastanka weba. Radi se o području umjetne inteligencije - predstavljanju znanja [Berners-Lee2001]. Umjetna inteligencija, upravo kao i semantički web, trebala je promijeniti svijet, no usprkos obećavajućim rezultatima to se nije ostvarilo [Antoniou2008]. Ta činjenica ipak ne bi trebala obeshrabriti i utjecati na interes za razvoj semantičkog weba, iako je umjetna inteligencija jedan od njegovih stupova. Naime, i u slučaju da se ne može doseći ljudska razina inteligencije, pa tako ni agenti ne mogu izvesti konačne zaključke bez ljudske interakcije, bilo kakvo njihovo zaključivanje koje se temelji na računalno dostižnoj razini inteligencije bio bi već velik napredak u odnosu na web danas [Antoniou2008]. Slika 9 i slika 10 ilustriraju razliku korištenja weba i semantičkog weba.



Slika 9 Korištenje weba



Slika 10 Korištenje semantičkog weba

Realizacija *zaključivanja* u umjetnoj inteligenciji, pa i u semantičkom webu, podrazumijeva primjenu znanja iz logike. Općenito, logika definira: [Vossen2007]

- formalan jezik za opis znanja
- formalnu semantiku za definiranje prirodnog jezika
- mehanizam za zaključivanje (eng. reasoner)

Mehanizam za zaključivanje iz danog opisa znanja donosi zaključke i na taj način čini implicitno znanje eksplicitnim. Slika 11 prikazuje primjer zaključivanja. Također, što je u nekim slučajevima upotrebe i važnije, kao na primjer kod opisa domenskih modela aplikacija, zaključivanjem se može otkriti nekonzistentnost modela.

Slika 11 Primjer zaključivanja - predikatna logika

Jedan od osnovnih problema klasičnih sustava za predstavljanje znanja je centralizacija. ***Pravila*** i ***podaci*** koji postoje u jednom sustavu teško su prenosivi u drugi sustav. Semantički web usmjeren je na decentralizaciju, odnosno omogućavanje prenosivosti (zajedničkog korištenja) podataka i pravila. S jedne se strane, od semantičkog weba očekuje standardiziranje značenja podataka, kao što je XML (Extensible Markup Language) standardizirao strukturu podataka, kako bi podatke mogle koristiti različite aplikacije [Jazayeri2007]. S druge strane, dodavanje logike webu, odnosno decentralizaciju samih pravila, koju je znatno teže ostvariti. Prvo pitanje koje se nameće jest koliku izražajnost logike uopće treba zahtijevati, mora se uspostaviti kompromis između opsega kojeg pravila obuhvaćaju i računalne efikasnosti. Što je logika izražajnija, treba više vremena da se postigne zaključak, a onda naravno uvijek prijeti mogućnost da se do rezultata u realnom vremenu uopće ne može doći [Antoniou2008]. Da bi se logika uopće mogla koristiti na webu, podaci i pravila moraju biti izraženi web jezicima. U poglavlju 3.1.4 predstavljene su tehnologije koje omogućavaju decentralizaciju podataka i primjenu logike na webu.

3.1.3. W3C – World Wide Web Consortium

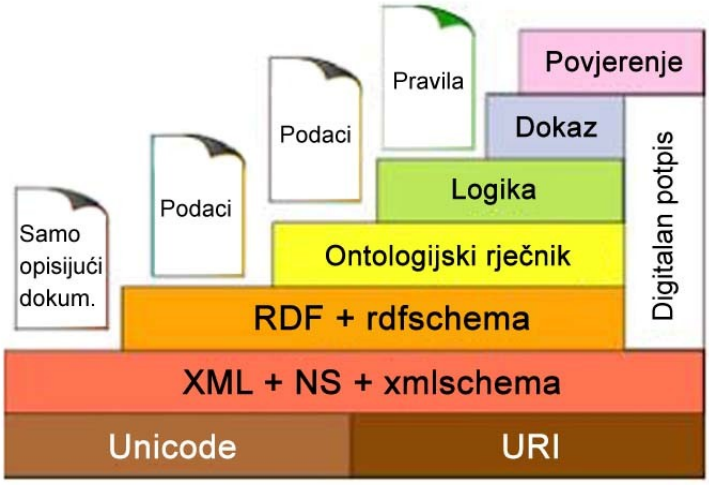
Namjena W3C[5] konzorcija je provođenje standardizacije web tehnologija u svrhu podržavanja interoperabilnosti na webu. Dakle, osnovni cilj je kontrolirati evoluciju weba kako bi se zadržala kompatibilnost tehnologija, spriječila njegova

[5] http://www.w3.org/

fragmentacija, odnosno kako bi se između web aplikacija omogućila interoperabilnost. W3C konzorcij čine znanstvenici i partneri iz privrede iz cijelog svijeta te se na taj način nastoji izbjeći protekcionizam pri donošenju odluka. Radne grupe za pojedina područja predlažu specifikacije koje nakon niza rasprava, analiza i promjena bivaju prihvaćene ili odbijene kao W3C preporuke (eng. W3C recommendation). Neke od poznatijih tehnologija sa statusom W3C preporuke su HTML 4.01, XML, CSS (Cascading Style Sheets), WSDL (Web Services Description Language), itd. Tehnologije koje su se razvile, te one koje se još moraju razviti da bi se vizija semantičkog weba mogla i ostvariti, trebale bi također biti odobrene kao W3C standardi. Na taj se način osigurava dugoročna egzistencija aplikacija u promjenjivom web okruženju.

3.1.4. Semantičke web tehnologije

Kako bi se definirala i izgradila infrastruktura potrebna za stvaranje aplikacija koje bi činile semantički web, predložen je slojevit pristup razvoja. Slika 12 prikazuje prvi prijedlog arhitekture. Prevelik je problem definirati sve tehnologije odjednom, pa je predložena ideja standardizacije slojeva, počevši od najnižih. Od svakog sljedećeg sloja zahtijeva se da mora biti kompatibilan s onim prethodnim. Do trenutka provođenja ovog istraživanja svi slojevi još uvijek nisu u potpunosti definirani te će se u nastavku opisati samo one tehnologije koje su postale W3C standardi. Prijedlog arhitekture slojeva doživio je brojne promjene od svog prvotnog izgleda kojeg prikazuje slika 12, do posljednjeg prijedloga arhitekture kojeg prikazuje slika 13.

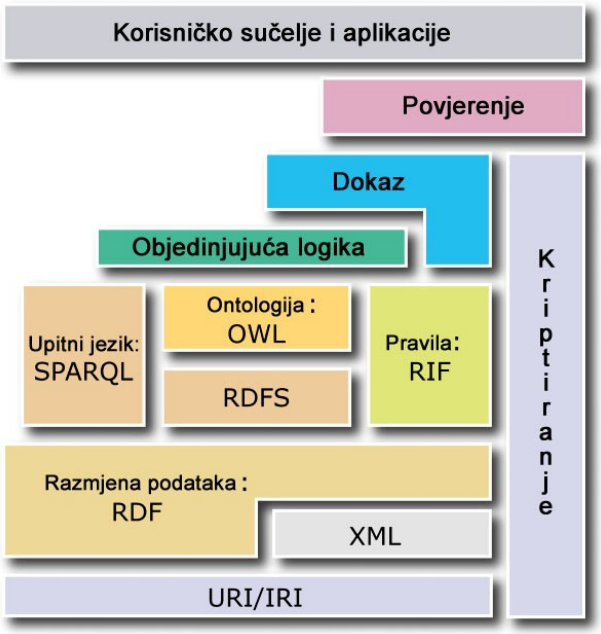


Slika 12 Slojevi semantičkog weba - prvotni izgled (2001.)

Postoje brojna istraživanja u kojima se diskutira kakav bi trebao biti izgled stôga tehnologija. Problemi su nastupili kada su se identificirali primjeri koji dovode u pitanje adekvatnost tehnologija sa samog dna piramide semantičkog weba, što je dodatno obeshrabrivalo ikakvu inicijativu razvijanja semantičkih web aplikacija. U [Kifer2005] se predlaže pristup u kojem bi se definiralo više stogova tehnologija među kojima bi bila moguća interoperabilnost, jer se smatra da je jedan stog tehnologija nerealističan i neodrživ. Svoju ideju temelje na mišljenju da je nerealno, pogotovo u današnjem vremenu, očekivati da će tehnologije koje su definirane na dnu piramide biti nezamjenjive. Da zamišljen način kompatibilnosti slojeva nosi probleme potvrđuje se i u radu [Horrocks2003] u kojem autori iznose probleme koji su nastupili zbog tog što se OWL nije stvarao u izolaciji, već se tražilo da se naslanja na XML i RDF slojeve definirane i već odobrene od strane W3C-a. Smatra se da je kao rezultat razvijanja pod tim uvjetima OWL postao kompliciraniji jezik nego što je to trebao biti. Jedna od većih i značajnih izmjena u arhitekturi semantičkog weba je mjesto za pravila. RDF i OWL nisu dovoljni za realizaciju nekih prepoznatih primjera slučajeva korištenja semantičkog weba navedenih u [Boley2007]. Ukazivanje na otkrivene primjere potaklo je na razvoj tehnologije za razmjenu pravila na webu (eg. RIF – Rule Interchange Format) koja bi podržavala znatno izražajnija pravila.

Pravila su važan dio semantičkog weba, pa treba razjasniti razliku između različitih tipova pravila budući da stručnjaci iz područja umjetne inteligencije, semantičkog weba i baza podataka imaju različito shvaćanje njihove definicije. Sva pravila semantičkog weba dijele se u sljedeće kategorije: [Boley2007]

- deduktivna pravila – opisuju statičku ovisnost između entiteta (u semantičkom webu opisana u RDFS-u, OWL-u, te se mogu izvoditi SPARQL-om)
- normativna pravila – postavljaju ograničenja na podatke ili na logiku aplikacije (u semantičkom webu dijelom opisana u RDFS-u i OWL-u)
- okidajuća pravila - koriste se za opisivanje ponašanja koje se odvija nakon nekog događaja (trenutno ne postoji odobrena tehnologija semantičkog weba za njihov opis)



Slika 13 Slojevi semantičkog weba – stog tehnologija (2006.)

Budući da su tema ovog istraživanja semantičke web aplikacije, važno je uočiti kako je infrastruktura na kojoj su se one morale temeljiti mijenjala kroz vrijeme. Kako bi se izbjegao zastoj u razvoju semantičkog weba, bila je preporuka da se aplikacije razvijaju koristeći one tehnologije koje su definirane, a ne da se čeka definiranje svih slojeva budući da bi taj zadatak mogao iziskivati jako puno vremena [Antoniou2008]. Naravno, osim u znanstvene svrhe, upitna je vrijednost takvih aplikacija kao i njihova dugoročna isplativost što je rezultiralo jako malim brojem semantičkih web aplikacija, koje se primjenjuju u praksi. Infrastruktura još nije standardizirana do kraja, gledajući posljednji prijedlog koji prikazuje slika 13, standardizirani su redom URI/IRI, XML, RDF, SPARQL (radi se na njegovom proširenju), RDFS i OWL (u tijeku je razvoj OWL 2).

3.1.4.1. URI i Unicode (univerzalni skup znakova)

U semantičkom webu svaki pojam, bilo da se radi o apstraktnim neopipljivim pojmovima (npr. sreća) ili opipljivim stvarima iz svakodnevnog života (npr. kuća) ima jedinstven identifikator URI (Uniform Resource Identifier). URI je identifikator i za sve dokumente koji čine web te u tom slučaju je URI zapravo jednak URL-u (Uniform Resource Locator). Ostale se sheme, poput URN (Uniform Resource Name) ili DOI (Digital Object Identifier) shema, nikada ne koriste na semantičkom webu. Na semantičkom se webu URI identifikatori ne dijele na web lokacije i imena, kako je to dosad bila neslužbena praksa na webu dokumenata (URL, URN), stoga

što oni uvijek imaju obje uloge. Ono što je posebno važno za korisnike s jezičnih područja čiji jezici za ispravno prikazivanje svih znakova koriste Unicode je uvođenje IRI-a (International Resource Identifier), kao i njegovo nedavno prihvaćanje od strane W3C-a [Boley2007]. IRI ima isto značenje kao i URI, s razlikom da može sadržavati i znakove izvan engleske abecede.

3.1.4.2. RDF i RDF Shema

Jezici RDF, RDFS i OWL nazivaju se ontologijskim jezicima. Prije no što se krene u opise tih jezika, potrebno je uvesti pojam ontologije onako kako se ona koristi u okviru semantičkog weba. ***Ontologija*** je formalna, eksplicitna specifikacija zajedničke konceptualizacije. *Konceptualizacija* se odnosi na apstraktan model nekog dijela svijeta, na način da se identificiraju relevantni pojmovi za njegov opis. *Eksplicitno* podrazumijeva da je vrsta pojmova koji se koriste i njihova uporaba eksplicitno definirana [Studer1998]. Uspoređujući ontologiju s ostalim shemama klasifikacije kao što su taksonomije i tezaurus, prema [Huhns1997] ontologije pružaju potpuniji i precizniji model podataka.

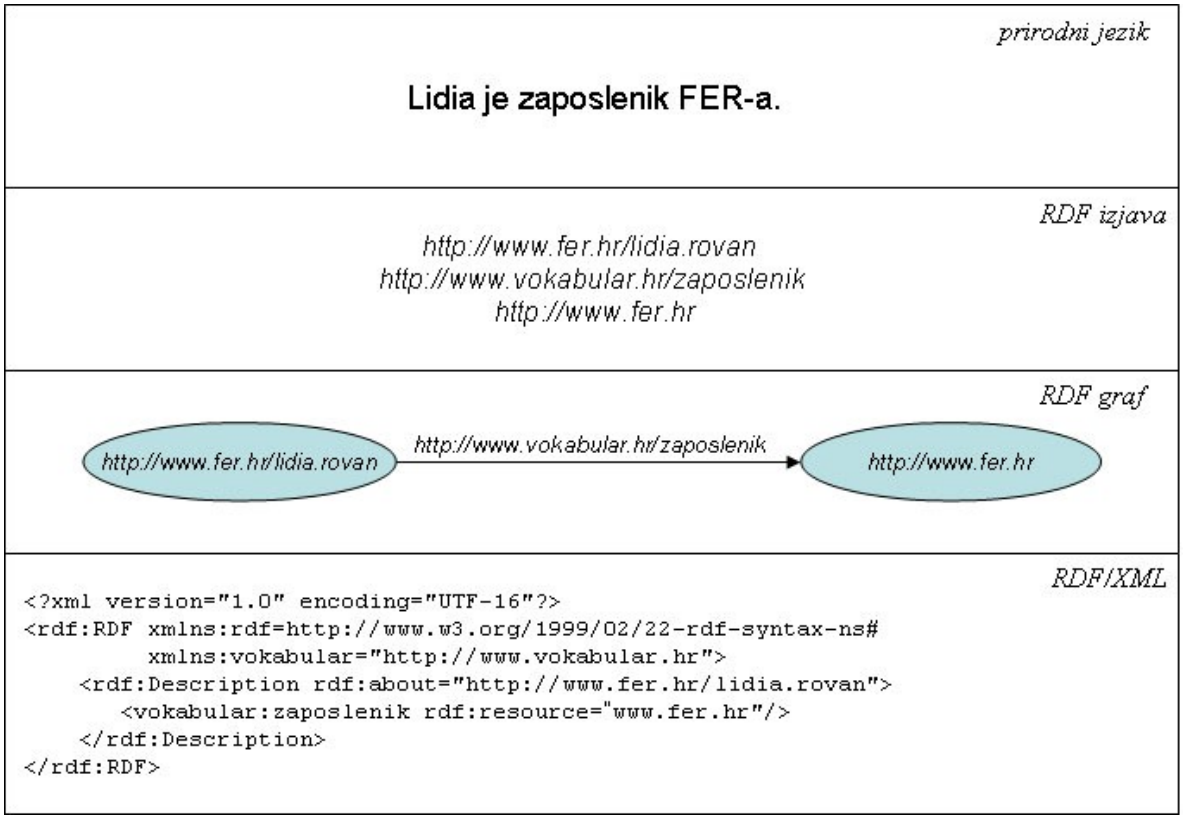
Sve se tehnologije semantičkog weba zasnivaju na XML-u, koji je opće prihvaćen meta-jezik za označavanje podataka. XML je dosad uspješno korišten kao sredstvo za razmjenu podataka, što znači da već postoje alati, tehnologije i metode kako koristiti i pohranjivati sadržaj XML dokumenata. Uz XML vežu se i drugi standardi [Boley2007]:

- XPath (XML Path Language) i XQuery za pretraživanje XML dokumenata
- XSLT (Extensible Stylesheet Language Transformations) za transformiranje XML dokumenata u druge dokumente
- XML Schema za definiranje strukture XML-a

Budući da je XML postao standard za strukturiranje podataka namijenjenih međusobnoj razmjeni, bio je prirodan izbor temeljne tehnologije semantičkog weba. XML se koristi samo kao format zapisa, jer semantiku ne može ni obuhvatiti te je ona pohranjena u višim slojevima arhitekture semantičkog weba.

Izazov je napraviti model podataka za semantički web, odnosno uobličiti prirodni jezik (prikazan na web stranicama) da bude razumljiv računalima, što je uostalom i sama okosnica semantičkog weba. Proučavajući gramatiku prirodnih jezika može se primijetiti da se svaka potpuna izjava sastoji od trojke: *subjekt*, *predikat* i *objekt* [Powers2003]. ***Trojke*** koje čine ***izjave*** su osnovni gradivni element RDF modela podataka. Postoje različiti načini predstavljanja trojki, a najčešće upotrebljavan način

i jedini odobren od strane W3C-a je XML zbog prethodno navedenih razloga. Slika 14 prikazuje različite načine predstavljanja izjave.



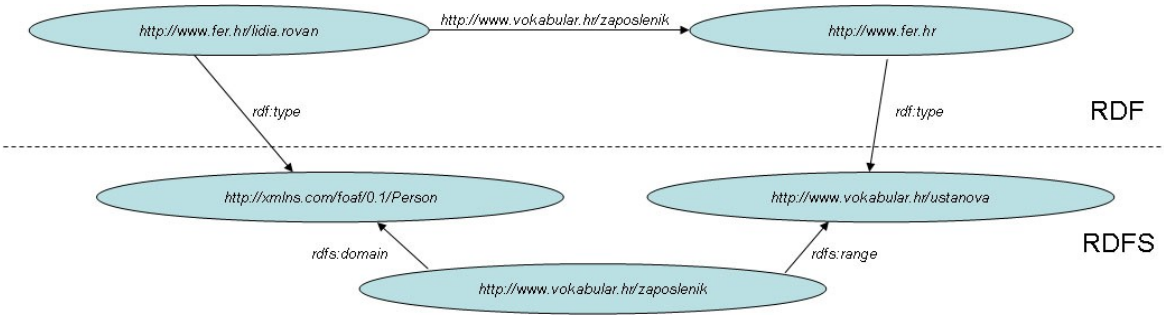
Slika 14 Vizualizacija RDF izjave

U RDF-u subjekt je uvijek *resurs*, predikat označava neko *svojstvo*, a *objekt* je vrijednost tog svojstva. Slijedi opis osnovnih termina:

- **resursi** – svi pojmovi opisani RDF izrazima zovu se resursima. Resurs može biti bilo što što je dio weba: cijela web stranica, dio neke web stranice, ali i kolekcija web stranica. Resursi su identificirani s URI-em, ali resursi mogu biti i neidentificirani (prazni čvorovi).
- **svojstva** - specifični aspekti, karakteristike, atributi ili relacije korišteni za opis resursa. Svako *svojstvo* ima svoje specifično značenje i može definirati svoju dopustivu vrijednost, tip resursa koji opisuje, te vezu prema ostalim opisima.
- **izjave** - pojedini resurs zajedno sa svojstvom, te vrijednosti tog svojstva za taj resurs čine RDF izjavu. Objekt izjave može biti drugi resurs ili podatkovna vrijednost (eng. literal).

RDFS [Brickley2009] je vrlo jednostavan ontologijski jezik, sadržava skup standardnih resursa koji definiraju značenja, karakteristike i odnose između dozvoljenih termina i omogućuje kreiranje vlastitog RDF rječnika, kao što su javno dostupni rječnici FOAF (Friend-of-a-friend) ili SKOS (Simple-knowledge-organisation-system-vocabulary). RDFS-om se može definirati hijerarhija klasa, odnosno klase i podklase, pripadnost svojstava klasi, svojstva, njihova domena i raspon, kao i hijerarhija svojstava. Slika 16 prikazuje primjer RDFS definicija.

Sam RDF je univerzalan jezik, neovisan o domeni, a dopuštena terminologija (rječnik ili vokabular) definira se RDFS-om. Slika 15 ilustrira odnos između RDF-a i RDFS-a.

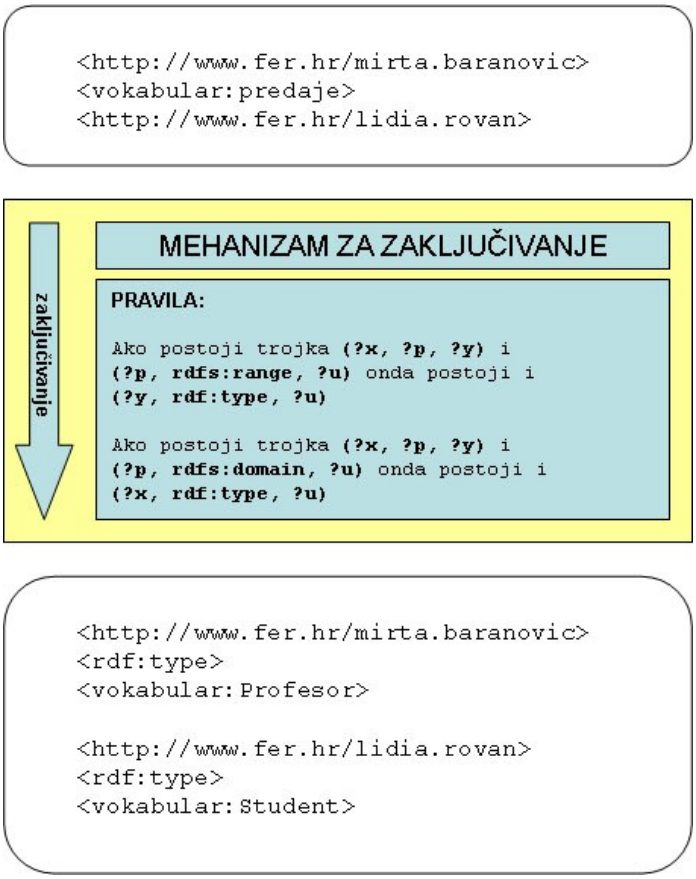


Slika 15 RDF i RDFS slojevi

Logika semantičkog weba već se jednim dijelom realizira zaključivanjem nad definicijama RDFS-a. Slika 17 pokazuje kako se na temelju definicija opisanih u RDFS-u, koje prikazuje slika 16, i samo jedne eksplicitne izjave, može izvesti dodatno znanje (implicitne izjave).

```
<rdfs:Class rdf:ID="vokabular:Student">
   <rdfs:subClassOf rdf:resource="foaf:Person"/>
</rdfs:Class>
<rdfs:Class rdf:ID="vokabular:Profesor">
   <rdfs:subClassOf rdf:resource="foaf:Person"/>
</rdfs:Class>
<rdf:Property rdf:ID="vokabular:predaje">
   <rdfs:domain rdf:resource="vokabular:Profesor"/>
   <rdfs:range rdf:resource="vokabular:Student"/>
</rdf:Property>
```

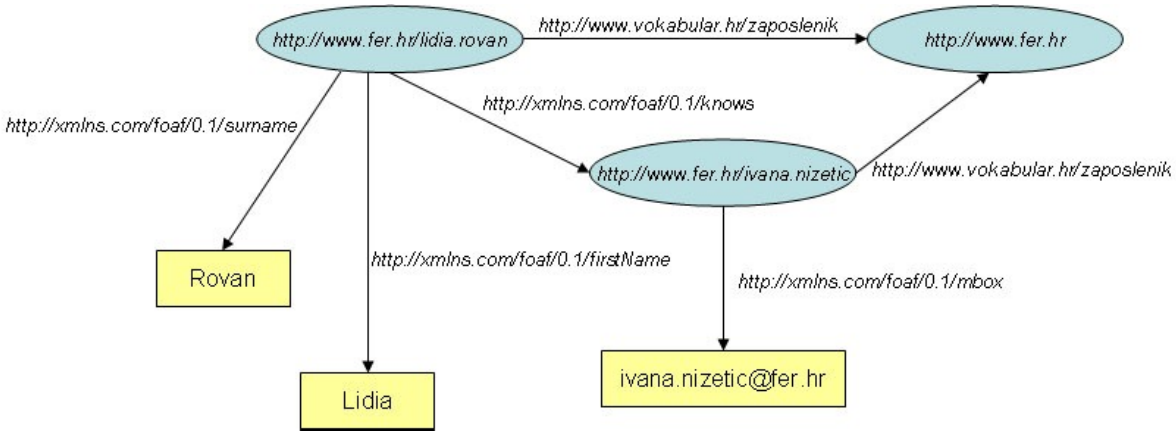
Slika 16 RDFS definicije



Slika 17 Zaključivanje nad RDFS-om

Važno je primijetiti da ne vrijedi analogija odnosa RDF-a i RDFS-a sa XML-om i XML Shemom. Dakle, XML Shema definira ograničenja u strukturi XML-a, dok RDFS definira pojmove koji se mogu koristiti u RDF izjavama i pridružuje im određeno značenje.

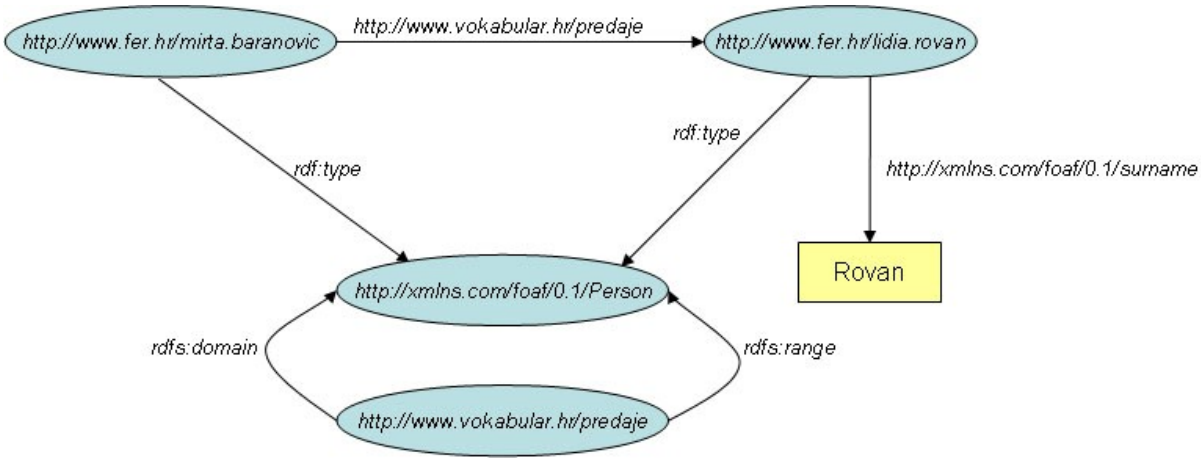
Budući da su u fokusu ovog istraživanja semantičke web aplikacije, potrebno je uvesti pojam ***RDF grafa***, jer podaci kojima aplikacija upravlja upravo čine RDF grafove. Sve akcije aplikacija nad podacima (npr. čitanje i mijenjanje podataka) zapravo su operacije nad RDF grafovima (npr. presjek ili unija). Općenito, RDF graf je skup RDF izjava, čvorove čine subjekti i objekti, dok su njegovi bridovi predikati. Slika 18 sadrži primjer RDF grafa. No, definiciju RDF grafa ipak treba detaljnije razraditi.



Slika 18 RDF graf

Pojam RDF grafa prvi put je uveden u službenoj W3C dokumentaciji [Klyne2009]. Na razvoj RDF-a utjecao je prethodni rad na polustrukturiranim modelima podataka [Abel2007, Klyne2009], od kojih treba izdvojiti OEM (Object Exchange Model) model podataka koji se također predstavlja grafom, pa je kao takav najsličniji RDF modelu podataka. Da bi se eliminiralo različitost shvaćanja grafa kao vizualnog modela, kao i grafa kao matematičkog pojma nad kojim se odvijaju spomenute dobro utvrđene operacije (presjek, unija, i sl.), u daljnjem će se tekstu definirati značenje RDF grafa koje će se koristiti tijekom ovog istraživanja.

Kako je definirano u [Klyne2009], RDF graf se tvori tako da se svaka trojka (s, p, o) predstavlja kao $s \xrightarrow{p} o$. Na taj se način tvori usmjereni označeni graf. Budući da je i sam predikat (p) URI kojemu se RDFS-om mogu dodjeljivati karakteristike, predikat se teoretski može pojaviti i kao brid, a ujedno i kao čvor. Višestruko pojavljivanje resursa narušava svojstva matematičkog modela grafa, točnije iz takve definicije proizlazi da se radi o hipergrafu [Hayes2004]. Slika 19 ilustrira kako se isti resurs (http://www.vokabular.hr/predaje) može javiti i kao brid i čvor. Ideja grafa je da podaci budu što čitljiviji, no hipergraf je gotovo nemoguće vizualizirati.



Slika 19 Primjer hipergrafa

Sve navedeno vrijedi samo u slučaju kada se uzima obzir RDF semantika (definirana RDFS-om). Ako se uzimaju u obzir samo eksplicitne izjave, odnosno izjave koje su rezultat zaključivanja nad RDF podacima i semantikom, tada se one uistinu mogu modelirati kao RDF usmjeren označen graf, koji može biti i nepovezan. Upravo nad takvim modelima izgrađuju se semantičke web aplikacije. U skladu s navedenim, uvodi se formalna definicija RDF grafova koji se koriste u semantičkim web aplikacijama:

Definicija 3.1: *RDF graf* G *je definiran kao* $G = (V, E, L, l)$ *gdje je* V *skup čvorova (subjekti i objekti),* E *je skup bridova (predikati),* L *je skup oznaka,* $l:E \rightarrow oznaka$ *je funkcija označavanja predikata (bridova).* V *i* E *su disjunktni. Projekcije* $ulaz:E \rightarrow V$ *i* $izlaz: E \rightarrow V$ *vraćaju početne i završne čvorove bridova.*

3.1.4.3. Upitni jezik – SPARQL

SPARQL (SPARQL Query Language for RDF) [Prud'Hommeaux2009b] je W3C standardiziran upitni jezik za RDF skupove podataka. Predloženi su i brojni drugi RDF upitni jezici s različitim svojstvima od kojih su značajniji: RQL (RDF query language) [Karvounarakis2002], SeRQL (Sesame rdf query language) [Boley2007], TRIPLE [Sintek2002] i RDQL (RDF Data Query Language) [Seaborne2009a]. Potpuna usporedba karakteristika navedenih upitnih jezika navedena je u radu [Haase2004].

RDF graf može imati različite oblike zapisa, na primjer trojna notacija, kao graf ili XML. Otkriveno je da njegov zapis korištenjem XML-a ima brojne nedostatke glede procesiranja upita [Broekstraw2002] te je istraživanje krenulo u smjeru obavljanja

upita na višoj razini apstrakcije, nad grafovima. Suštinski, SPARQL je upitni jezik koji se zasniva na usporedbi grafova (traženju uzoraka u grafovima) [Perez2006].

***Definicija 3.2*:** Svaki SPARQL upit Q je definiran kao četvorka Q = (GP, DS, SM, R), gdje je:

- ***GP*** – uzorak grafa (može biti samo jedan uzorak, uzorci mogu biti rekurzivno zadani, može se navesti unija uzoraka, uzorak može biti opcionalan, itd…)
- ***DS*** – RDF skup podataka nad kojim se obavlja upit (skup RDF grafova)
- ***SM*** – skup parametara koji modificiraju rješenje (utječu na poredak, eliminiraju duplikate, ograničavaju broj trojki u rezultatu)
- ***R*** – oblik u kojem se prikazuju podaci (SELECT – skup varijabli, ASK – logička vrijednost, CONSTRUCT – RDF graf i DESCRIBE – RDF graf)

U SPARQL-u su definirane četiri vrste operacija ovisno o prikazu rezultata podataka:

- ***SELECT*** *(Q = (GP, DS, SM, SELECT VS); VS – skup varijabli)* - iz izvora podataka izdvaja one elemente koji se podudaraju sa zadanim uzorcima
- ***ASK*** *(Q = (GP, DS, SM, ASK))*- vraća logičku vrijednost istine ili laži (*true* ili *false*) u ovisnosti postoje li elementi koji se podudaraju sa zadanim uzorcima u upitu
- ***CONSTRUCT*** *(Q = (GP, DS, SM, CONSTRUCT T); T – skup izjava)*- stvara RDF graf prema predlošku zadanom u upitu, *where* dio upita koristi za definiranje načina zamjene varijabli u predlošku s konkretnim vrijednostima
- ***DESCRIBE*** *(Q = (GP, DS, SM, DESCRIBE V); V – skup varijabli i IRI-a)*- vraća jedan RDF graf s podacima o URI-u. URI može biti konstanta ili varijabla čija se vrijednost dobije iz *where* dijela upita.

Gramatika SPARQL upita je kompleksna, pa slika 20 pojednostavljeno prikazuje strukturu SPARQL upita.

Prolog – opcionalan	BASE <*iri*>
Prolog – opcionalan; može se ponavljati	PREFIX *prefix*: <*iri*>
Vrste operacija (R)	SELECT (DISTINCT) … ili DESCRIBE … ili CONSTRUCT { *uzorak grafa* } ili ASK
Definicija izvora podataka (DS) – opcionalno; može se ponavljati	FROM <*iri*> ili FROM NAMED <*iri*>
WHERE izraz - (opcionalno, osim za ASK operaciju)	WHERE { *uzorak grafa* [FILTER *izraz*]}
Modifikatori rezultata (SM) – opcionalno	ORDER BY … LIMIT *n*, OFFSET *m*

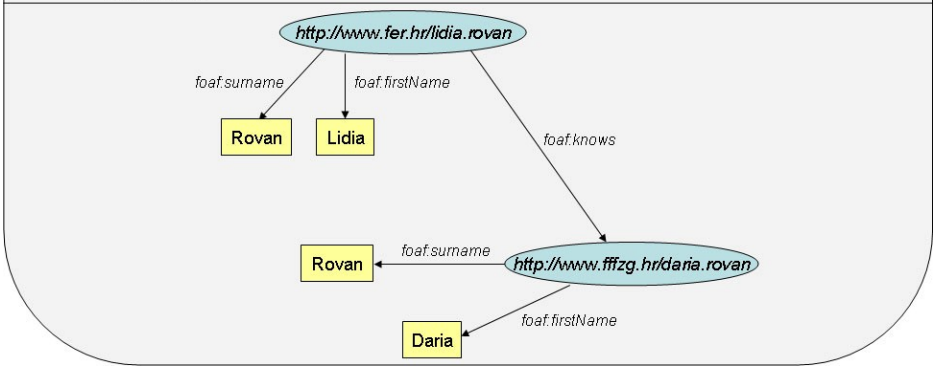
Slika 20 Struktura SPARQL upita

Slika 22 sadržava primjere svih vrsta operacija SPARQL upita i rezultat njihovog izvođenja na primjeru skupa podataka kojeg prikazuje slika 21.

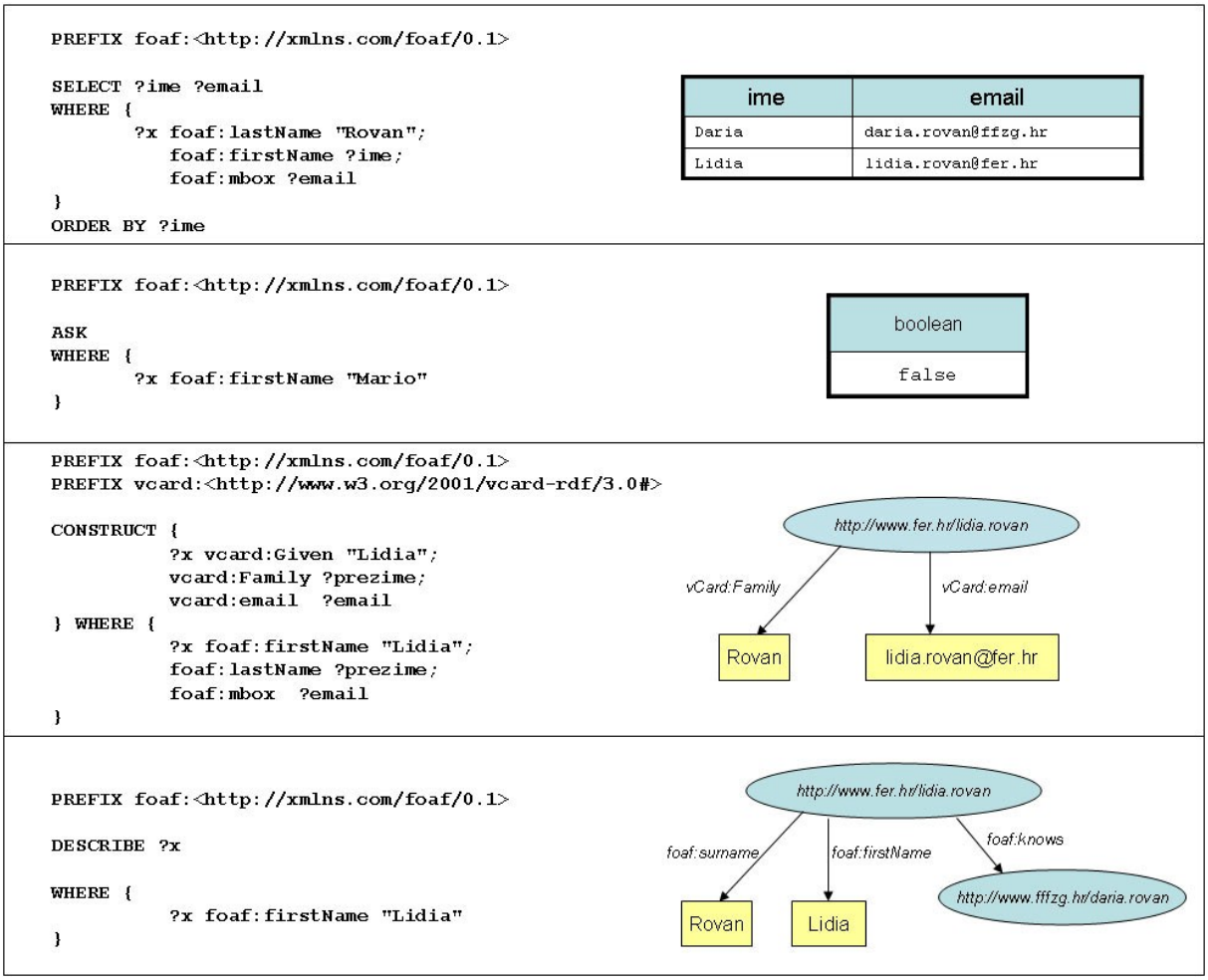
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <rdf:Description rdf:about="http://www.fer.hr/lidia.rovan">
     <foaf:firstName>Lidia</foaf:firstName>
     <foaf:surname>Rovan</foaf:surname>
     <foaf:mbox>lidia.rovan@fer.hr</foaf:mbox>
     <foaf:knows>
         <rdf:Description rdf:about="http://www.ffzg.hr/daria.rovan"/>
     </foaf:knows>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.ffzg.hr/daria.rovan">
     <foaf:firstName>Daria</foaf:firstName>
     <foaf:surname>Rovan</foaf:surname>
     <foaf:mbox>daria.rovan@fer.hr</foaf:mbox>
  </rdf:Description>
</rdf:RDF>
```



Slika 21 Skup podataka - RDF graf



Slika 22 SPARQL upiti

Predložena su brojna proširenja SPARQL-a, od kojih je svakako najvažniji SPARUL [Seaborne2009b] koji definira dodatne operacije za rukovanje RDF grafovima i njihovim sadržajem.

U SPARUL-u su definirane sljedeće operacije:

- ***INSERT*** – dodavanje izjava u graf
- ***DELETE*** – brisanje izjava iz grafa
- ***MODIFY*** – definira se skup za brisanje i dodavanje izjava iz modela; koristi se za izmjenu izjava
- ***CLEAR*** – briše sve izjave iz odabranog grafa
- ***LOAD*** – kopira sve izjave u odabrani graf
- ***CREATE GRAPH*** – kreira novi graf
- ***DROP GRAPH*** – briše zadani graf (sa cjelokupnim njegovim sadržajem)

Teško je ostvariti ikakvu funkcionalnost web aplikacija ako nije moguće mijenjanje podataka. Iako sada postoji način, odnosno tehnologija za mijenjanje grafova, problem je što još uvijek nema status W3C preporuke. Sve izmjene podataka grafa, npr. dodavanje ili brisanje izjava, mogu se obavljati i bez korištenja upitnog jezika. Procedura se svodi na učitavanje grafa, obavljanje promjena u memoriji, te ponovno pohranjivanje. Koliko će biti efikasna ta operacija, ovisi o dovitljivosti programera i mogućnostima korištenih programskih okvira. Standardizirano i efikasno rješenje koje bi se zasnivalo na algoritmima iz teorije grafova bi svakako doprinijelo efikasnosti semantičkih web aplikacija, naravno u slučaju kada se govori o RDF grafovima koji su tako i pohranjeni.

3.1.4.4. RDF trajna spremišta

Trenutno postoje različite implementacije semantičkih spremišta podataka, veliki dio njih koristi relacijsku bazu podataka kao trajno spremište, tako da su u njima implementirane različite nestandardne (posebno prilagođene) metode za pristup i upravljanje podacima. U početku je za svako spremište bio razvijen i poseban upitni jezik, ali nakon standardizacije SPARQL-a većina implementacija je prilagođena njegovoj primjeni. U literaturi se može naći podjela vrsta implementacija spremišta podataka u tri skupine [Liu2005]:

- ***sadržana u memoriji (eng. in-memory) –*** Jena[6], Sesame[7], itd…
- ***prirodna spremišta (karakteristične za RDF modele) –*** Sesame Native, Virtuoso[8], AllegroGraph[9], Oracle 11g[10], itd…
- ***temeljena na relacijskoj bazi podataka***. – Jena + (SQL Server, MySQL, PostgreSQL, itd...)

Iako je korištenje memorije kao spremišta daleko najefikasnije [Guo2005, Stocker2008], nedostatak je što se ponovno stvaraju svaki put kad se aplikacija pokrene i sve promjene se gube kad aplikacija završi s radom. Postoji mogućnost da se objekt koji sadrži model serijalizira na datotečni sustav, no opet postoje brojni nedostaci (npr. obnova, transakcije, kontrola pristupa, i sl.) koji su davno riješeni u sustavima za upravljanje relacijskim bazama podataka.

Prirodna spremišta i *spremišta temeljena na relacijskoj bazi podataka* trajna su spremišta, pri čemu prirodna spremišta pokazuju bolje performanse [Bizer2009b,

[6] http://jena.sourceforge.net/
[7] http://www.openrdf.org
[8] http://virtuoso.openlinksw.com/
[9] http://www.franz.com/agraph/
[10] http://www.oracle.com/technology/tech/semantic_technologies/index.html

Stocker2008]. Grafovima se rukuje na specifičan, njima primjeren način. Virtuoso je popularna implementacija prirodnog spremišta, radi se o proširenoj verziji MySQL poslužitelja baza podataka RDF grafovima - SPASQL. Ta promijenjena verzija MySQL poslužitelja ima mogućnost direktnog izvođenja SPARQL upita, bez potrebe za pretvaranjem u SQL [Prud'Hommeaux2009b]. Kod spremišta koja se temelje na relacijskim bazama podataka, SPARQL upiti se pretvaraju u SQL upite pa se sva optimizacija vrši na već uhodane načine specifične za relacijske baze. Dakle, odvija se korištenjem heuristike i statističkih podataka o bazi nad kojoj se upiti izvode [Stocker2008].

Što se tiče primjenjivosti RDF modela u aplikacijama, svakako bi se trebalo poraditi na optimizaciji upita – pronaći plan izvođenja koji bi najbrže vratio rezultate. Performanse izvođenja upita ovise o statičkim tehnikama optimizacije (npr. određivanjem redoslijeda spajanja), ali i o indeksnoj strukturi [Harth2005] i točnosti statističkih informacija [Stocker2008]. Dostupni su razni rezultati mjernih testova SPARQL upita [Bizer2009b, Liu2005, Schmidt2008]. Iako je brzina jako važna, barem za sada ne bi trebalo tražiti da SPARQL upiti i vremenom obavljanja odgovaraju SQL upitima. Ipak, vremenska manjkavost se kompenzira točnijim (semantički) i izražajnijim odgovorima. Mora se naglasiti da SPARQL upiti imaju semantički veću preciznost od SQL upita te, ovisno o kontekstu u kojem se upotrebljavaju, mogu poprilično skratiti vrijeme utrošeno u obavljanje neke zadaće.

Što se više podataka opisuje RDF-om, osim problema optimizacije upita, javlja se i problem skalabilnosti. S velikim se brojem podataka memorijski modeli ne mogu mjeriti zbog memorijskog ograničenja, tako da postaje upitna svrha daljnjeg usavršavanja takvih modela. Ono što je trenutno veliki izazov istraživanja je kako obavljati efikasne upite nad velikom količinom trojki, odnosno kako pohranjivati podatke u stalno spremište. Postoji više istraživanja koje koriste zanimljive nove tehnike rješavanja problema optimizacije, kao što su npr. genetski algoritmi [Gueret2008, Hogenboom2008] ili objektno-orijentiran pristup [Corno2008], a i ostali pristupi [Cheng2008, Erling2008].

Neka od spremišta imaju implementiran mehanizam za zaključivanje pa zaključivanje nije potrebno izvoditi u višim slojevima aplikacije. U RDF spremištima zaključivanje se izvodi pri dodavanju izjava u spremište (stvaraju se nove izjave) i za vrijeme izvođenja upita. Kada bi se sve izjave izračunavale apriori, spremište bi moglo drastično narasti, dok bi zaključivanje trajalo predugo kad bi se obavljalo u trenutku izvođenja upita, tako da se u primjeni teži kompromisu [Staab2004].

Funkcije SPARQL-a

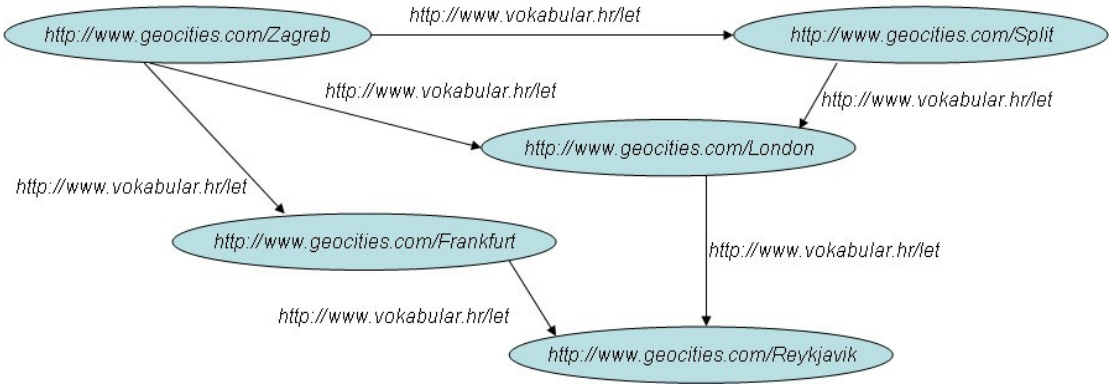
Nedostatak funkcija u SPARQL-u (samo ih je nekoliko u W3C službenoj dokumentaciji) trenutno jako utječe na percipiranu korist primjene tehnologija semantičkog weba u obavljanju složenijih zadaća od samog pretraživanja skupa podataka. Postoje ugrađene funkcije u pojedinim alatima za rukovanje RDF grafovima, tako npr. Jenin[11] modul ARQ ima niz funkcija za operacije s nizovima. Iako postoje primjene RDF-a (pa tako i SPARQL-a) u poslovnim aplikacijama [Servant2008], realizacija poslovnih izvještaja čini se nemoguća (bespotrebno kompliciranа pomicanjem u viši sloj) bez agregatnih funkcija (npr. SUM, AVG, COUNT, …) i grupiranja. U prihvaćenoj specifikaciji SPARQL-a kao rezultat upita ne može se vratiti vrijednost koja nije eksplicitni dio trojke. Izrazi su dozvoljeni samo u FILTER dijelu SPARQL upita. Slika 23 prikazuje primjer korištenja izraza u SPARQL-u. Ipak, implementacija nekih agregatnih funkcija postoji u Virtuosu i u ARQ-u (modul Jene) uslijed prepoznate potrebe za primjenom semantičkih tehnologija u poslovnim aplikacijama (npr. skladište podataka).

```
PREFIX  dc:  <http://purl.org/dc/elements/1.1/>
PREFIX  ns:  <http://example.org/ns#>
PREFIX  xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT  ?cijena ?naslov
WHERE   { ?x ns:cijena ?cijena .
          ?x dc:naslov ?naslov
          FILTER ( (?cijena * xsd:double(0.9) ) < 38)
}
```

Slika 23 Primjer korištenja izraza u SPARQL upitu

Ono što bi trebala biti najveća prednost modela koji čini graf pronalaženje je putova u tom grafu. Postoje brojni primjeri iz prakse koji su pokazatelj potrebe za modelima organiziranim na takav način. Na primjer, pronalaženje letova od grada A do grada B. Slika 24 ilustrira primjer takvog modela nad kojim bi učestali problem bilo traženje najkraćeg puta od čvora do čvora, što upravo predstavlja problem iz teorije grafova. Relativno velik broj istraživanja bavio se problematikom identificiranja putova u grafu. Istraživanja su rezultirala raznim predloženim proširenjima SPARQL-a koji su specijalizirani upravo za takve upite [Alkhateeb2008, [Kochut2007, Anyanwu2007]. Ipak još nema konkretizacije u smislu proširenja standardnog SPARQL-a.

[11] Jena je programski okvir za razvoj semantičkih Java aplikacija - http://jena.sourceforge.net/



Slika 24 Primjer RDF grafa

3.1.4.5. SPARQL protokol

Trend servisno orijentirane arhitekture te općenito trend otvaranja sustava u cilju razmjene i integracije podataka dao je na važnosti REST (Representational State Transfer) web servisa. REST je stil arhitekture koji se temelji na tehnologijama i protokolima weba i može se primijeniti za izradu web servisa. REST servisi su puno jednostavniji za implementaciju i korištenje od SOAP (Simple Object Access Protocol) servisa. Dizajn temeljen na REST-u fundamentalan je za Web 2.0. aplikacije (utječe na spoj egzaktnih dimenzija ***integracije podataka*** i ***funkcionalnost***), a i izvrsno se uklapa u novije vizije semantičkog weba koje više uključuju korisnika [Battle2006]. Zbog toga je važno spomenuti još jednu W3C preporučenu tehnologiju koja je odgovor semantičkog weba na imperativ uklapanja u nove web standarde. - SPARQL protokol (SPROT). SPROT [Clark2009] je jednostavan način za postavljanje SPARQL upita i primanje rezultata preko HTTP-a ili SOAP protokola. U ovom istraživanju koristi se samo primjena SPROT-a korištenjem HTTP-a, pa će se primjena SPROT-a korištenjem SOAP-a zanemariti.

Izvođenje SPARQL upita preko HTTP protokola prilično je jednostavno, SPARQL URL sastoji se od tri dijela:

1. URL SPARQL pristupne točke (eng. endpoint) - (npr. *http://dbpedia.org/sparql*)
2. Imena grafova nad kojima se postavlja upit – (ovaj dio je opcionalan; npr. *default-graph-uri=http://dbpedia.org***)**
3. SPARQL upit - (npr. *Query=SELECT distinct ?x WHERE {?x a <http://umbel.org/umbel/sc/Artist> }***)**

Primjer cjelokupnog URL-a je sljedeći:

```
http://dbpedia.org/sparql?default-graph-
uri=http://dbpedia.org&query= SELECT DISTINCT ?winner WHERE {
  <http://dbpedia.org/resource/World_Snooker_Championship_2009>
  <http://dbpedia.org/property/winner>
  ?winner
}
```

Rezultati upita postavljenih preko SPARQL protokola najčešće se vraćaju u obliku XML-a. Iako SPARQL proširenje za rukovanje izmjenama grafova još nije standardizirano, svakako je potrebno i proširenje SPARQL protokola koje bi podržalo udaljenu izmjenu i rukovanje grafovima je neophodno.
Slika 25 prikazuje primjer HTTP zahtjeva i odgovora prilikom izvođenja SPARQL upita korištenjem SPROT protokola putem HTTP-a.

```
GET /sparql?default-graph-uri=http://dbpedia.org&query=
SELECT DISTINCT ?winner WHERE {
  <http://dbpedia.org/resource/World_Snooker_Championship_2009>
  <http://dbpedia.org/property/winner>
  ?winner
}
Host: dbpedia.org


HTTP/1.1 200 OK
Server: Virtuoso/06.00.3124 (Solaris) x86_64-sun-solaris2.10-64  VDB
Connection: close
Date: Mon, 09 Nov 2009 08:03:10 GMT
Accept-Ranges: bytes
X-SPARQL-default-graph: http://dbpedia.org
Content-Type: application/sparql-results+xml; charset=UTF-8

<sparql xmlns="http://www.w3.org/2005/sparql-results#">
 <head>
  <variable name="winner"/>
 </head>
 <results distinct="false" ordered="true">
  <result>
   <binding name="winner">
   <uri>http://dbpedia.org/resource/John_Higgins_%28snooker_player%29</uri>
   </binding>
  </result>
 </results>
</sparql>
```

Slika 25 Primjer izvođenja SPARQL upita putem HTTP protokola

3.1.4.6. OWL

Ontologije imaju najvažniju ulogu pri ostvarenju scenarija opisanih na početku ovog poglavlja u obliku potpore agentima u procesu dohvata informacija. Ontologije su zapravo strukturirani rječnici koji sadrže eksplicitne veze između različitih pojmova te na taj način omogućavaju inteligentnim agentima da jednoznačno interpretiraju njihovo značenje [Horrocks2003].

OWL [McGuinness2009] je jezik ontologija za semantički web, koji se temelji na deskriptivnoj logici i koji je postao W3C standard 2004. godine. OWL nije prvi jezik koji je bio prilagođen za web, tako da su na razvoj imali utjecaj njegovi prethodnici od kojih ipak najviše DAML+OIL [Horrocks2003]. Uzevši u obzir arhitekturu semantičkog weba, OWL proširuje mogućnosti RDF-a i RDFS-a (prethodnih slojeva u stogu tehnologija). Temelji se na načinu izražavanja činjenica kojeg propisuje RDF te na definiranju hijerarhije svojstava i klasa kojeg omogućava RDFS [Horrocks2003]. Ono što OWL dodatno podržava je ***definiranje logičkih operacija nad OWL klasama (presjeci, unije i komplement), definiranje kardinalnosti pojedinih objekata te definiranja karakteristika svojstava***. Na primjer, svojstvo može biti tranzitivno, simetrično ili inverzno u odnosu na neko drugo svojstvo. Također, može se definirati koji objekti (jedinke) pripadaju kojoj klasi te koje vrijednosti svojstava imaju pojedini objekti. Ekvivalencija (izjednačavanje) može se definirati nad klasama i svojstvima, dok se disjunktnost može definirati samo nad klasama, a jednakost i nejednakost samo na objektima. Kao najveći dodatak izdvaja se mogućnost restrikcije na korištenje svojstava pojedine klase [Horrocks2003].

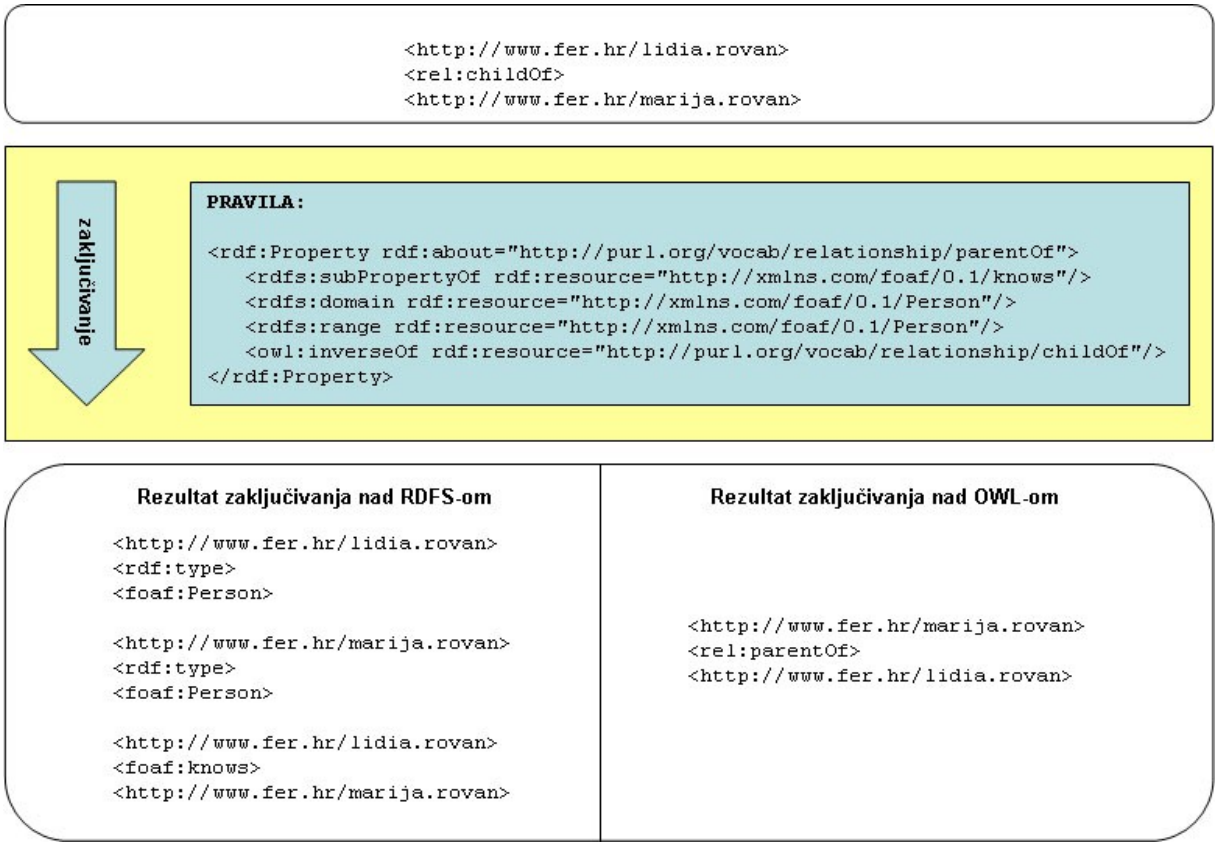
Slika 26 sadrži primjer OWL zaključivanja. Gledajući primjer, ukoliko bi se koristilo zaključivanje samo nad RDFS pravilima na pitanje "je li Marija roditelj Lidije", dobili bi negativan odgovor, jer takva izjava eksplicitno ne postoji (nije izvedena). No međutim, korištenjem OWL inverznog pravila (karakteristike svojstva), te zaključivanjem nad tim pravilom, dobijemo eksplicitno i izjavu da je Marija Lidijin roditelj.

OWL ima tri varijante jezika koje se mogu koristiti ovisno o traženom stupnju zaključivanja [Horrocks2003]:

- **OWL-Lite** – moguća klasifikacija hijerarhija i jednostavnih ograničenja; koristi se gdje je dovoljna minimalna izražajnost ontologija
- **OWL-DL** – maksimalna izražajnost koja garantira izračunavanje; daje minimalna ograničenja na jezik uz maksimalnu izražajnost

- **OWL-Full** – maksimalna izražajnost i sintaktična sloboda RDF-a bez garancije za izračunavanjem

Budući da OWL nudi napredne metode zaključivanja, ujedno radi znatan pomak u onom što je primarni cilj semantičkog weba, povećava izražajnost pitanja na koja bi agenti mogli korisniku pružiti traženi odgovor, odnosno dozvoljava korisnicima određenu slobodu nepreciznosti u svojim pitanjima baš kao što je ona prisutna i u realnom životu. U okviru web aplikacija, OWL omogućava bolju iskoristivost baze znanja nad kojom se aplikacija temelji u smislu veće količine informacija koje može ponuditi korisniku.



Slika 26 Zaključivanje nad RDFS i OWL pravilima

U tijeku je razvoj nove verzije OWL-a, OWL 2 Web Ontology Language [Motik2009] koji u odnosu na sadašnju verziju na traženje korisnika dodatno obuhvaća skup funkcija koji omogućava korištenje efikasnijih algoritama zaključivanja.

4.Semantički web danas

Budućnost je već tu. Samo još nije široko rasprostranjena.
William Gibson

Prijedlog i razrada arhitekture semantičkih web aplikacija jedan je od osnovnih rezultata ovog istraživanja. Da bi se do takvog rezultata moglo doći, bilo je nužno provesti analizu i evaluaciju dostupnih tehnologija i metodologija njihovog korištenja. Te tehnologije u konačnici predstavljaju elemente predložene arhitekture koji se povezuju slijedeći zadana pravila (metodologije) pa kao takve imaju direktan utjecaj na njen izgled. U ovom poglavlju prikazan je rezultat provedene analize i evaluacije. Napravljen je pregled trenutnog stanja razvoja semantičkog weba. Razjašnjava se nastanak novih, trenutno aktualnih pravaca istraživanja područja semantičkog weba koji se dalje slijede tijekom ovog istraživanja. Ti novi pravci istraživanja za cilj imaju stvaranje *mreže podataka* i *mreže oznaka*. Također, dan je pregled istaknutih tehnologija i preporuka koje su usko vezane uz izgradnju semantičkih rješenja koja udovoljavaju kriterijima novih pravaca istraživanja.

4.1. Udaljavanje od osnovne vizije semantičkog weba

Iako je razvoj semantičkog weba krenuo od ideje semantičkih web agenata, da bi ta ideja zaživjela, potrebno je oformiti cijelu tehnološku infrastrukturu koja bi podržala njihov automatiziran rad. Kao rezultat dosadašnjeg istraživanja, izrodile su se brojne semantičke web tehnologije od kojih se većina pokazala korisna za rješavanje određenih, specifičnih problema. Uzevši u obzir takve dostupne primjere iz prakse [SWCSUC2009], može se reći da je semantički web u svojoj osnovnoj viziji parcijalno zaživio. Odnosno, postoje semantička rješenja koja za sad funkcioniraju kao izolirani otoci. Da bi semantički web zaživio na globalnoj razini, potrebno je ustrajati na povezivanju tih rješenja, znači prvenstveno nastojati povezati različite izvore podataka na kojima se takva rješenja zasnivaju. Kako bi se na temelju dosad ostvarenih rezultata u istraživanju semantičkog weba što prije dobio opipljiviji oblik na globalnoj razini, napravljen je zaokret u istraživanjima te su se istraživači gotovo u potpunosti posvetili povezivanju semantičkih podataka (instanci, a ne meta-podataka) weba.

Pod utjecajem uspjeha rješenja Weba 2.0, od 2007. godine javljaju se novi pravci istraživanja u području semantičkog weba koji za cilj imaju ostvarenje *mreže*

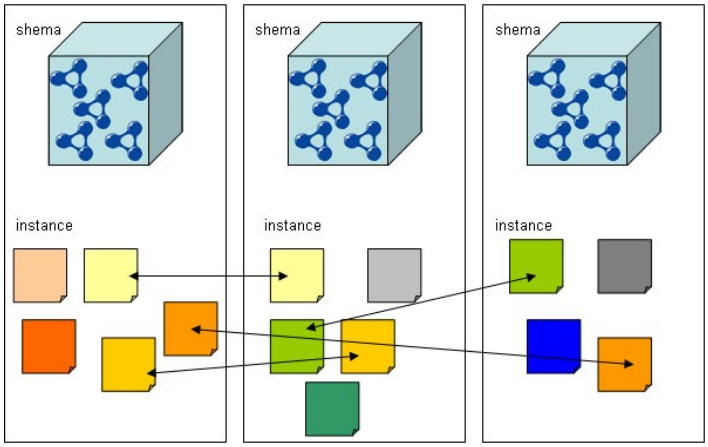
podataka i *mreže oznaka*. Predložen način ostvarenja „mreže podataka“ bliži je osnovnoj, početnoj ideji semantičkog weba. Dakle, podacima se daju značenja na njihovom izvoru, u raznim bazama podataka. Zbog takvog načina uvođenja semantike, taj se pristup naziva i *pristup od dna prema vrhu*. S druge strane, *pristup od vrha prema dnu* odgovara *mreži oznaka*. Mreža oznaka još je jedna reakcija na sporu realizaciju semantičkog weba. Temelji se na ideji da su korisnici ti koji daju oznake na sadržaje prikazane na web stranicama. Budući da su razni načini označavanja sadržaja jako zastupljeni u okviru Weba 2.0, sasvim je logično da se krenulo s ugradnjom semantike u proces označavanja kakav danas postoji. Ovaj pristup obećava privlačenje jako velikog broja korisnika što upravo predstavlja okidač koji semantičkom webu treba. No, dok je vizija *mreže oznaka* lako ostvariva, točnost tako unesene semantike je upitna, a također ima i mali stupanj izražajnosti. Vizija *mreže podataka* obećava točnije podatke, može se uvesti veća izražajnost, ali je, naravno, i teška za realizaciju. Da bi se ostvarila globalna realizacija, ipak je potrebna sinergija oba navedena pristupa.

4.2. Mreža podataka – povezani podaci

Razvoj semantičkog weba trenutno je usmjeren na razvoj mehanizama za definiranje veza između podataka na webu, odnosno na njihovo automatsko detektiranje. Taj novi pravac istraživanja pripada području „mreže podataka“ (eng. Web of data) ili „povezanih podataka“ (eng. Linked data) [Berners-Lee2009b]. Kao što je već objašnjeno u prethodnom poglavlju, podaci u semantičkom webu pripadaju ontologijama. Ako se gleda iz kuta deskriptivne logike, baza znanja se sastoji od Abox-a i Tbox-a, gdje su u Tbox-u sadržana pravila, a u Abox-u same instance [Wang2006]. Sve do 2006. godine istraživanja su uglavnom bila usmjerena na preslikavanje shema ontologije (Tbox), dok se u posljednje tri godine intenzivno radi na povezivanju samih instanci (slika 27). Integracija na razini instanci znatno je važnija za izgradnju web aplikacija i čini osnovni problem njihove izrade.

Povezivanje je ono što je i sam web učinilo uspješnim. Semantički web uvodi još finije veze. Za razliku od weba gdje su uspostavljene veze između dokumenata, ovdje se veze uvode između podataka sadržanim u različitim izvorima podataka. „Povezani podaci“ su koncept koji je osmislio Tim Berners-Lee i iznio u svojoj bilješci o arhitekturi weba [Berners-Lee2009b]. Cilj poveznih podataka je omogućavanje jednostavnog objavljivanja i dijeljenja podataka na webu. Temeljna pretpostavka na kojoj počiva ideja povezanih podataka jest da je vrijednost i korisnost proizvoljnog

podatka na webu tim veća što je snažnija i bolja njegova povezanost s drugim podacima na webu. U spomenutoj je bilješci Tim Berners-Lee naveo načela kojima bi se korisnici trebali voditi prilikom dodavanja sadržaja na web kako bi semantički web mogao dosegnuti što veću funkcionalnost i izražajnost:

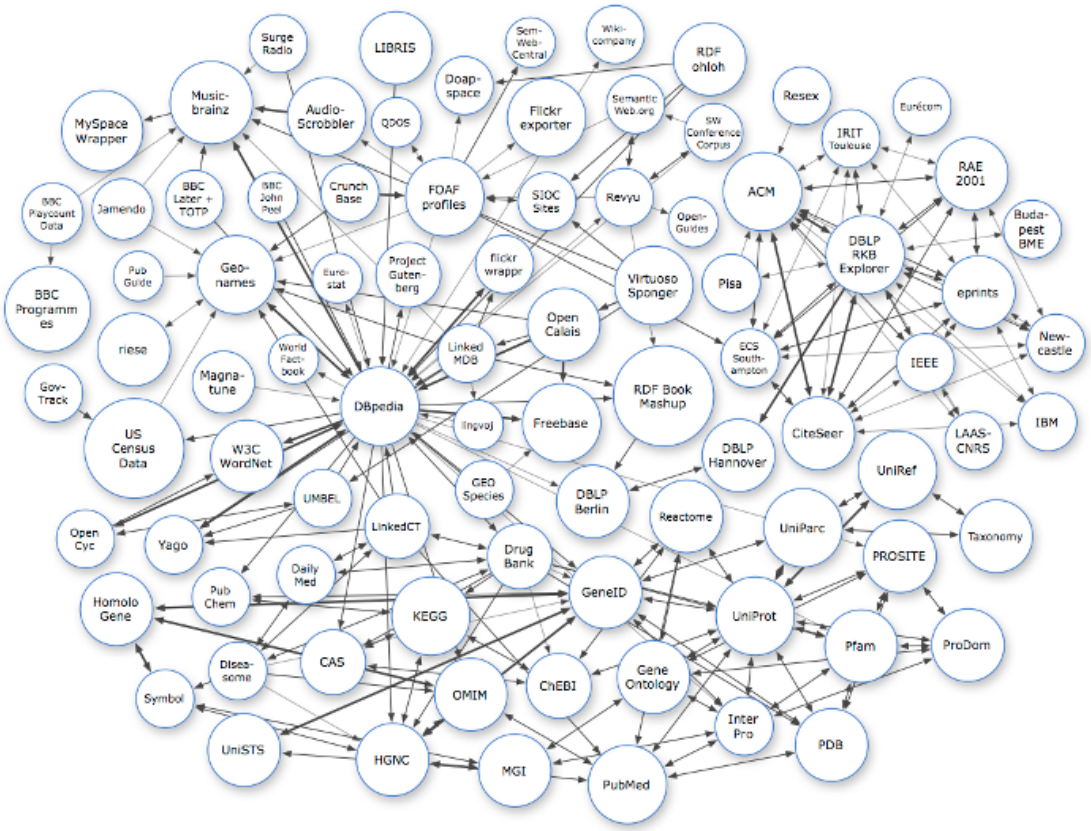


Slika 27 Povezivanje instanci ontologija – "povezani podaci"

Načela povezanih podataka:

1. *svi elementi moraju biti identificirani URI-em (nema praznih čvorova)*
2. *svi URI-i moraju se moći razriješiti (eng. dereference) – pronaći HTTP URL koji daje koristan opis elementa identificiranim zadanim URI-em*
3. *moraju se postaviti veze prema drugim URI-ima kako bi se omogućila daljnja laka potraga za podacima*

Dokument [Bizer2009a] još detaljnije opisuje kako objaviti povezane podatke te objašnjava dva osnovna pristupa za kreiranje veza prema drugim skupovima podataka. U osnovi, postoje dva načina postavljanja veza: *ručno definiranje veza* i *korištenje algoritama sličnosti*. U ovom istraživanju iznesena su opažanja o primjenjivosti ta dva načina ovisno o kontekstu u kojem se žele primijeniti. Detalji o primjenjivosti su objašnjeni kasnije u poglavlju 6.3 - Semantički web portal.



Slika 28 Povezani podaci (ožujak 2009.) [Linked2009]

Kao rezultat istraživanja na području povezanih podataka, a pod okriljem projekta Linking Open Data (LOD) nastaju izvori podataka koji objavljuju podatke u skladu s navedenim načelima. Trenutno je obuhvaćeno tridesetak različitih izvora podataka (slika 28), koji zajedno sadrže preko 2 bilijuna trojki što je prilično impresivno, ali i dalje čine mali postotak svih podataka weba.

4.2.1. Postupak objavljivanja podataka – načela povezanih podataka

4.2.1.1. Odabir URI-a

URI je vrsta identifikatora koju je kao standard za web propisao W3C. Ključna svojstva URI identifikatora su ***jednoznačnost*** i ***globalnost***. *Jednoznačna identifikacija* znači da ne mogu postojati dva resursa na webu koja će imati isti URI identifikator, dok *globalna identifikacija* znači da se URI oznaka jednako tumači u svakoj njenoj primjeni na webu. Navedena svojstva čine osnovne uvjete koje je potrebno ispuniti da bi se ostvarila globalna mreža podataka. URI identifikatori definiraju se unutar shema (eng. URI scheme) koje određuju zapis, odnosno sintaksu identifikatora te najčešće i protokol vezan uz taj identifikator. Za potrebe

semantičkog weba i povezivanja podataka koriste se identifikatori koji pripadaju HTTP URI shemi, takozvani HTTP URI identifikatori.
Dodatna svojstva koja bi URI trebao zadovoljiti su: ***jednostavnost***, ***stabilnost*** *i* ***upravljivost*** [Berners-Lee2009b].

URI-i koji zadovoljavaju navedene kriterije nazivaju se "Cool URIs" [Berners-Lee2009a], a tehnike za objavljivanje takvih URI-a obuhvaćene su W3C dokumentom "Cool URIs for the Semantic Web" [Sauermann2009].

Svojstvo ***jednostavnosti*** (eng. simplicity) očituje se u duljini i simbolici URI-a. Jednostavni URI-i su robusni i otporni na pogreške u sintaksi te se uz to lako i pamte. Primjerice, lokalni put do resursa na poslužitelju ne bi trebao biti dio identifikatora. Tehnički, takvo nešto nije teško ostvariti budući da svi današnji HTTP poslužitelji imaju mogućnost preslikavanja resursa u proizvoljne identifikatore na domeni.

Svojstvo ***stabilnosti*** (eng. stability) posjeduje identifikator koji se nikad ili gotovo nikad ne mijenja. To znači da bi se iz identifikatora trebali izbaciti svi dijelovi koji jesu ili bi mogli biti podložni izmjenama. Dijelovi koji ovise o tehnologiji posluživanja ili prikazivanja, poput nastavka imena datoteke, nepoželjni su jer nisu prilagodljivi promjenama tehnologije.

Svojstvo ***upravljivosti*** (eng. manageability) poželjno je ukoliko je resurs određen URI-em podložan promjenama. Dobra je navika stavljanje vremenskih oznaka u URI, najčešće godine objavljivanja. Time je omogućeno objavljivanje novih inačica resursa bez gubitka starih.

4.2.1.2. Vokabular – opis podataka

Globalnu mrežu podataka moguće je ostvariti uz uvođenje opravdanih ograničenja na točnost i izražajnost semantičkih modela nad kojima se podaci temelje. Znači, potrebno je uvesti ograničenja u proces kreiranja vokabulara. Osnovni napuci za objavljivanje pojmova koji bi bili dio mreže podataka su sljedeći: [Bizer2009a]

1. **Izbjeći definiranje potpuno novih vokabulara** – uvijek pokušati pronaći postojeće vokabulare te nad njima dodati nove pojmove
2. **Pripremati ih i za računala, ali i za ljude** – uvijek uz pojam dodati i tekstualna objašnjenja za ljude koristeći svojstva *rdfs:comment* i *rdfs:label*
3. **Svaki URI mora biti moguće razriješiti** – objašnjenje URI pojma učiniti dostupnim različitim klijentima

4. **Koristiti pojmove definirane od strane drugih autora** – korištenjem takvih pojmova ili samo pružanjem veza prema njima ostvaruje se mreža podataka; obično se u tu svrhu koriste svojstva rdfs:subClassOf ili rdfs:subPropertyOf
5. **Sve važne informacije izreći eksplicitno** – domene, raspon ili sl. treba eksplicitno definirati, budući da računala ne mogu pogađati kao ljudi
6. **Ne stvarati modele sa suviše ograničenja, ostaviti prostora za fleksibilnost i mogući rast –** preporuka je koristiti samo RDFS za definiranje logičkih pravila, budući da se u mreži podataka očekuje da se i drugi referenciraju na objavljen podatak; ukoliko se koriste OWL svojstva može doći do nekonzistentnosti kada se netko drugi referencira na taj isti pojam

4.2.1.3. Razrješavanje URI-a

Pojam razrješavanja URI-a (eng. dereferencing) predstavlja proces dohvaćanja prikaza URI-a s ciljem da se dohvate informacije o referenciranom resursu. Taj prikaz resursa (eng. representation) je oblikovana poruka čiji sadržaj nosi srž informacije koju taj resurs sadrži. Primjenski programi poput preglednika weba ostvaruju prikaz resursa kako bi ih korisnik mogao osjetiti, tj. vidjeti ili čuti.

Resursi semantičkog weba dijele se na *informacijske resurse* i *ostale resurse*. Ta je podjela jako važna u kontekstu koncepta povezanih podataka, jer se razrješavanje URI-a odvija različito za navedene grupe.

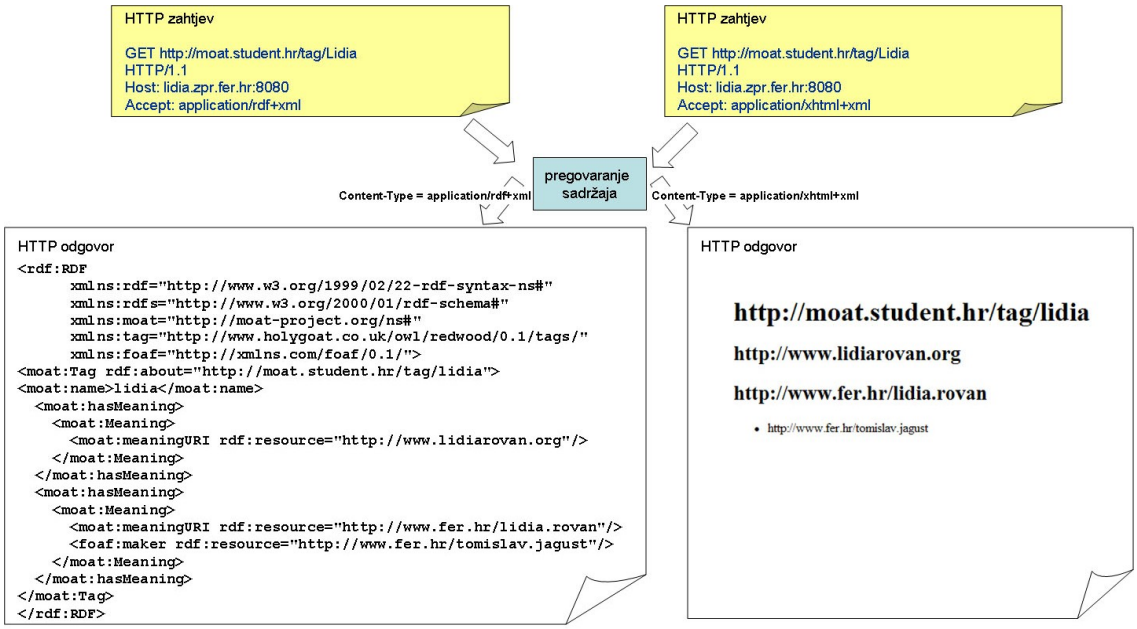
- ***Informacijski resursi*** (eng. information resources) su dokumenti kojima se rukuje na webu poput tekstualnih dokumenata, slika i drugih multimedijskih zapisa, čije se osnovne osobine mogu prenositi webom putem poruka. Informacijski resursi obično imaju jedan ili više prikaza (npr. prikaz na različitim jezicima) kojima se može pristupiti korištenjem HTTP protokola.
- ***Ostali resursi*** su oni koji su označeni URI-ima, a nalaze se izvan prostora weba. Ti se resursi nazivaju neinformacijski ili ostali resursi (eng. non-information resources, other resources) i oni obuhvaćaju osobe, grupe, organizacije, prirodne pojave te apstraktne i druge pojmove. Kako neinformacijski resursi nisu prisutni na webu, oni ne mogu posjedovati prikaz.

Razrješavanje URI-a realizira se na način da se HTTP poslužitelj prilagodi tako da vraća odgovore na GET zahtjeve za određenim resursom predstavljenim URI identifikatorom na sljedeći način:

- Odgovor s HTTP statusnim kodom *200 (OK)* označava da URI identifikator iz zahtjeva predstavlja informacijski resurs te da je njegov odgovarajući prikaz sadržan u tom odgovoru.
- Odgovor s HTTP statusnim kodom *303 (See Other)* označava da URI iz zahtjeva predstavlja resurs koji nema prikaz te u sebi sadrži URI identifikator njemu povezanog resursa koji može, ali ne mora posjedovati prikaz.
- Odgovor s HTTP statusnim kodom iz područja *4xx i 5xx* označava da je došlo do pogreške na strani klijenta, odnosno na strani poslužitelja. Iz tog se odgovora ne može razlučiti na koju se vrstu resursa odnosi traženi URI, ako je URI uopće ispravan.

Za usmjeravanje klijenta na informacijski resurs koji posjeduje odgovarajući prikaz, primjerice HTML dokument za preglednike weba, a RDF dokument za semantički preglednik weba, koristi se još jedan mehanizam HTTP protokola koji se zove pregovaranje sadržaja.

Pregovaranje sadržaja (eng. content negotiation) [Bizer2009a] predstavlja mehanizam definiran HTTP specifikacijom koji omogućava posluživanje različitih verzija dokumenta, odnosno prikaza resursa predstavljenog jednom URI oznakom kako bi korisnički agenti poput preglednika weba i preglednika semantičkog weba mogli u zahtjevu (u zaglavlju (eng. header) HTTP zahtjeva) navesti koji im oblik prikaza odgovara. Slika 29 prikazuje primjer pregovaranja sadržaja.



Slika 29 Primjer pregovaranja sadržaja

4.2.1.4. Postavljanje veza prema drugim izvorima podataka

Veze su, kao što i samo ime kaže, okosnica ideje povezanih podataka. Jedno od načela mreže podataka kaže da vlasnici objavljenog resursa *moraju* taj resurs povezati s unaprijed poznatim resursima na semantičkom webu. Udovoljavanje tom načelu bitno otežava postupak povezivanja podataka te je jedan od razloga usporenog rasta broja povezanih podataka na webu. Sama je implementacija veze trivijalna, svodi se na postavljanje svojstava koji definiraju odnose između dvije instance. Tablica 2 sadrži primjere veza između podataka.

Tablica 2. Primjer veza između instanci

`owl:sameAs`	najčešće korišteno svojstvo pri postavljanju veza, označava da su dva svojstva jednaka
`foaf:knows`	svojstvo iz FOAF vokabulara, označava da osoba poznaje neku drugu osobu
`foaf:based_near`	svojstvo iz FOAF vokabulara, označava pojam koji je fizički blizu (obično za vezivanje na instance geonames vokabulara)
`contact: nearestAirport`	svojstvo iz PIM (personal information management) Contacts ontologije za definiranje najbližeg aerodroma
…	…

Motiv za pružanje veza prema drugim izvorima podataka je integracija i omogućavanje preglednicima i sličnim alatima kretanje po mreži podataka i skupljanje mogućih korisnih informacija koje pobliže označavaju traženi resurs. Proces postavljanja veza zahtijeva puno vremena, a moguće su i pogreške u povezivanju. Naime, često se neće povezati baš iste instance, nego slične. Proces se odvija u dva koraka:

1) pronalaženje izvora podataka na koje se treba povezati
2) definiranje veza između podataka

Pronalaženje izvora podataka na koje se treba povezati

Dok nije postojalo toliko semantičkih izvora podataka (slika 28), pronalaženje izvora na koje se žele pružiti poveznice bilo je jednostavno. Sada se za lakše pronalaženje mogu koristiti preglednici povezanih podataka (npr. Tabulator[12], Disco[13] ili Zitgist

[12] http://www.w3.org/2005/ajar/tab
[13] http://www4.wiwiss.fu-berlin.de/bizer/ng4j/disco/

DataViewer[14]) ili pretraživači (npr. Sindice[15]). No, iako postoje alati koji pomažu pri pronalaženju izvora, sam odabir onog podatka koji je najprikladniji opet obavlja čovjek vodeći se svojom intuicijom.

Definiranje veza između podataka

Budući da je manualan pristup definiranja veza između podataka nemoguć za velike izvore podataka, pojavila su se razna rješenja koja automatski traže veze ili na temelju uzoraka URI-a ili na temelju mjera sličnosti. Pristup ne podrazumijeva automatizaciju oba navedena koraka postavljanja veza. Odgovarajuće instance automatski će se identificirati, ali se svejedno mora ručno odrediti izvor podataka nad kojim će se ta identifikacija i obavljati.

U nekim domenama postoje jedinstveni identifikatori, npr. ISBN (International Standard Book Number) za knjige, JMBG/OIB za građane Hrvatske, međunarodne oznake država, i sl. Kada je u URI-u resursa sadržana neka od jedinstvenih oznaka, tada se može primijeniti algoritam traženja uzorka i mogu se izjednačiti instance dvaju izvora kao u sljedećem primjeru:

<http://www.fer.hr/knjiznica/07632201> owl:sameAs < http://www.nsk.hr/knjige/07632201>

Prethodno treba provjeriti da uistinu druga strana sadrži podatak koji se pokušava izjednačiti da se ne bi referencirali nepostojeći podaci.

U slučaju kada ne postoje upotrebljivi identifikatori u URI-u, tada se primjenjuju algoritmi sličnosti koji se provode nad svojstvima instanci. Dobar primjer uspješnosti ovakvih metoda je izjednačavanje gradova po vrijednosti geografske lokacije, budući da ta svojstva moraju imati istu vrijednost. Kako bi se smanjilo vrijeme potrebno za pretraživanje i pronalaženje srodnih resursa, trenutno je područje automatskog traženja sličnih instanci predmet brojnih istraživanja [Hausenblas2008, Hassanzadeh2009, Raimond2008, Volz2009, Wang2006].

Među rezultatima dosadašnjih istraživanja izdvaja se alat koji predstavlja općenito rješenje pa se može primijeniti nad svim povezanim podacima - **Silk** (Silk Link Discovery Framework*)* [Volz2009]. Daje se njegovo detaljnije objašnjenje, budući da se tehnike traženja sličnosti resursa koriste dalje u istraživanju. Silk je primjer alata koji pomaže u automatizaciji procesa traženja srodnih resursa. Omogućuje pretraživanje RDF grafa u potrazi za podacima koji odgovaraju uvjetima

[14] http://dataviewer.zitgist.com/
[15] http://sindice.com/

koje zadaje korisnik te stvaranje dotad nepostojećih veza među njima. Uvjeti se definiraju pomoću Silk – Link Specification Language (Silk-LSL) jezika, koji je dijalekt XML jezika. Slika 30 prikazuje primjer dokumenta pisanog u Silk-LSL jeziku. U *Prefix* oznaci navode se vokabulari koji se koriste, *DataSource* predstavlja izvore podataka nad kojima se obavlja povezivanje, *Metric* oznaka služi za određivanje mjera sličnosti između podataka, dok *Interlink* sadrži definiciju veza.

```
<?xml version="1.0" encoding="utf-8" ?>
<Silk>
      <Prefix ... />
        ...
      <DataSource ... />
        ...
      <Metric ... />
        ...
      <Interlink ... />
        ...
</Silk>
```

Slika 30 Primjer strukture dokumenta pisanog u Silk-LSL jeziku

Korisnici mogu navesti kakve se RDF veze mogu otkrivati među resursima te koje uvjete resursi moraju ispunjavati kako bi se povezali. Uvjeti uključuju različite mjere sličnosti, koje se mogu proizvoljno podesiti kako bi se u obzir uzimali i rezultati koji se ne podudaraju u potpunosti s uvjetima. Silk pretražuje RDF grafove koristeći SPARQL pristupne točke te postavljajući SPARQL upite na osnovi navedenih uvjeta. Rezultat izvođenja pretrage skup je trojki koji predstavlja pronađene veze u zadanom podatkovnom skupu, koji se može upotrijebiti za dodavanje novih resursa u mrežu povezanih podataka.

4.3. Mreža oznaka

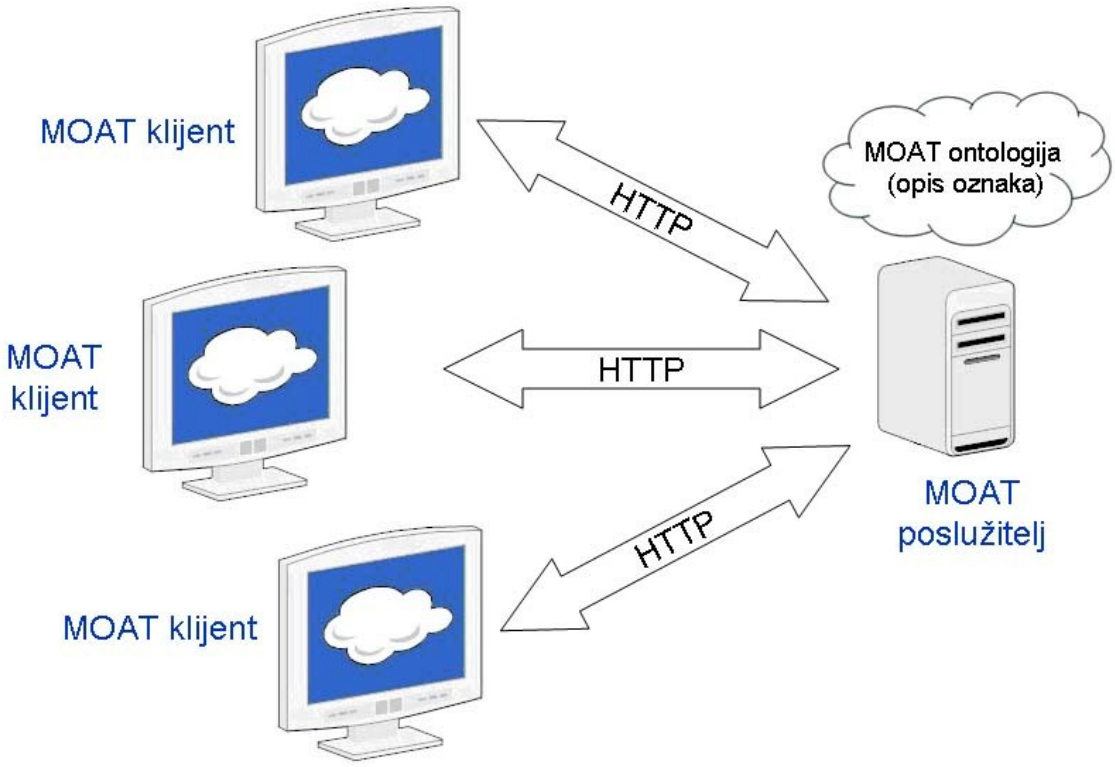
Označavanje sadržaja (eng. tagging, content tagging) razvilo se s popularizacijom Weba 2.0. Omogućilo je korisnicima dodavanje semantičkih oznaka dokumentima koje su stvorili. Oznaka (eng. tag) je ključna riječ ili izraz dodijeljen dijelu informacijskog resursa koji pruža dodatan opis resursa i omogućava lakše pronalaženje pretraživanjem weba.

Načelo slobodnog označavanja (eng. free tagging) sadržaja, koje omogućava stvaranje oznaka autorima, vlasnicima ili korisnicima resursa, rezultira neformalizmom i neuređenosti sustava oznaka. U nedostatku tijela za formalizaciju i uređenje neizbježno dolazi do nejednoznačnosti unutar skupa oznaka.

Nejednoznačnost se može očitovati kao istoznačnost dviju oznaka te kao višeznačnost jedne oznake. Problem predstavlja i činjenica da oznake izvorno ne posjeduju mehanizam koji bi omogućio njihovo međusobno povezivanje.

Projekt MOAT [Passant2008b] (Meaning of a Tag) oslovljava navedene probleme omogućujući korisnicima određivanje značenja oznaka povezujući ih s URI-ima koji predstavljaju njihovo značenje. Tako se omogućuje dodavanje semantičkih podataka slobodno označenom sadržaju, povezujući ga u mrežu podataka semantičkog weba. MOAT također omogućuje dijeljenje značenja oznaka među korisnicima i zajednicama te koristi FOAF vokabular za povezivanje moguće višeznačnih oznaka s osobama koje su definirale pojedina značenja.

MOAT se oslanja na arhitekturu koja obuhvaća MOAT vokabular, MOAT poslužitelj, kao i klijente drugih proizvođača (eng. third-party clients) te koja se može implementirati u bilo kojoj organizaciji ili zajednici. Slika 31 vizualizira komunikaciju između dijelova MOAT arhitekture.



Slika 31 MOAT arhitektura

4.4. Tehnologije i alati

Sve tehnologije opisane u poglavlju 3.1.4 čine slojeve arhitekture semantičkog weba ili se direktno na nju naslanjaju. Tehnologije i pripadajući alati opisani u ovom poglavlju ne čine same stupove semantičkog weba, već njihovo proširenje koje ima jasnu primjenu u postojećim uvjetima weba. U poglavlju 4.2 navedene su tehnologije koje su usko vezane uz ostvarenje "mreže podataka", dok su u poglavlju 4.3 navedene tehnologije usko vezane uz ostvarenje "mreže oznaka". Tehnologije koje se dalje opisuju, omogućuju prezentaciju semantike, odnosno kontakt korisnika i semantičkog weba i nisu usko vezane ni uz jedan od navedenih pravaca razvoja semantičkog weba, već predstavljaju općenita rješenja. Također, dan je i kratak osvrt na programski okvir Jena, koji je korišten pri izradi svih semantičkih web rješenja u ovom istraživanju.

4.4.1. Mikroformati

Tehnologija mikroformati (eng. microformats) [Microformat2009] omogućava da se u postojeće aplikacije na jednostavan način može unijeti semantika. Radi se o tehnologiji koju je vrlo lako naučiti i koristiti, pa i podaci iz 2007. godine [Adida2008] ukazuju na njihovu veliku popularnost. Na žalost, iako su jednostavni za implementaciju (ali ne i za održavanje!) količina semantike koju oni mogu podržati u većini slučajeva nije dovoljna. Koncept mikroformata temelji se na karakteristici da neki HTML atributi sadrže i semantiku. Na primjer HTML atribut <title> predstavlja naslov dokumenta. Dakle, kod mikroformata sva semantika se uvodi korištenjem sljedećih HTML atributa: *class, rel i title* [Microformat2009]. Slika 32 prikazuje na odsječku (X)HTML stranice kako se ugradi semantika korištenjem jedne od definiranih sintaksa mikroformata (hCal) na događaj u kalendaru.

```
<div id="hcalendar-predavanja" class="vevent">
   <a href="http://www.fer.hr/predmet/bazepod" class="url">
   <abbr title="2009-05-04T10:00" class="dtstart">May 4, 2009 10</abbr> -
   <abbr title="2009-05-04T12:00" class="dtend">12pm</abbr> :
   <span class="summary">Predavanja</span> u
   <span class="location">D2</span></a>
</div>
```

Slika 32 Mikroformat – kalendar

Mikroformatima se može pripisati problem nepružanja mogućnosti za razvoj općenitih rješenja, a koji je posebno važan u kontekstu održavanja aplikacija. Kao što je istaknuto u radu [Adida2008], mikroformati ne podržavaju *neovisnost* i *proširivost*. Naime, korištenje više mikroformata istovremeno može rezultirati konfliktom. Dodatno, jedno od većih ograničenja ove tehnologije je što se ne preporuča korištenje mikroformata koji nisu unaprijed definirani, pa tako primjerice parseri koji dohvaćaju semantiku iz (X)HTML dokumenata podržavaju samo određen skup vokabulara.

4.4.2. GRDDL

GRDDL (Gleaning Resource Descriptions from Dialects of Languages) [Connolly2009] tehnika je za generiranje RDF podataka iz XML dokumenata s posebnim naglaskom na XHTML kao dijalektu XML-a. GRDDL pruža skup mehanizama pomoću kojih je moguće transformirati sadržaj zapisan XML sintaksom u RDF sadržaj. Autori mogu eksplicitno definirati da dokument sadrži GRDDL metapodatke (dodati *profile* atribut u head element) i povezati dokumente s transformacijskim algoritmima, obično napisanim u XSLT (XSL Transformations) tehnologiji. Iako se teoretski mogu podržati i algoritmi pisani u drugim programskim jezicima, XSLT trenutno podržavaju GRDDL svjesni agenti (npr. koristeći Jena GRDDL reader[16]). Programski klijenti koji čitaju dokument mogu slijediti hiperveze na webu koje vode na odgovarajuće transformacije opisane u GRDDL specifikaciji.

```
<head profile='http://www.w3.org/2003/g/data-view'>
 <link type="text/css" rel="stylesheet"
href="http://www.w3.org/StyleSheets/base"/>
 <link rel='transformation' href='transformRaspored.xsl' />
</head>
```

Slika 33 Veza na transformacijski algoritam po GRDDL specifikaciji

Osim algoritama koje može napisati bilo tko zadužen za razvoj aplikacija, postoji i standardna biblioteka transformacija pomoću koje se ekstrahiraju RDF podaci zapisani u XML ili XHTML dokumentu. GRDDL omogućuje da se transformacije vezuju na sadržaj na više načina, ovisno o potrebi i pripadnom kontekstu. Najjednostavnija je metoda za autore (X)HTML sadržaja uključiti referencu na transformaciju direktno koristeći *link* element u zaglavlju dokumenta. GRDDL

[16] http://jena.sourceforge.net/grddl/

transformacije mogu se načiniti za skoro bilo koji web dijalekt – uključujući i već prije navedene mikroformate. Slika 33 prikazuje primjer veze iz (X)HTML dokumenta na transformacijski algoritam za sadržaj te stranice.

Mora se napomenuti da je velika prednost GRDDL-a jednostavnost korištenja, brzo dobivanje rezultata kao i postojanje niza testnih primjera koji znatno pomažu na putu učenja korištenja spomenute tehnologije. Kroz primjenu ove tehnologije možda se najbolje može i ocrtati trenutno stanje u primjeni tehnologija semantičkog weba. GRDDL je jednostavan za korištenje, ali značajno težak za održavanje, budući da se ipak radi o brojnim skriptama koje se trebaju mijenjati s gotovo svakom promjenom prezentacijskog sloja. Količina semantike koja se može ugraditi na ovaj način je ograničena, a pri tom je i njena uporabna vrijednost upitna.

4.4.3. eRDF

eRDF [Davis2009] je sintaksa koja, baš kao i mikroformati, koristi HTML atribute *class*, *rel* i *title* kako bi se omogućilo umetanje jednostavnih RDF podataka u HTML prikaz. Korištenjem atributa *profile* elementa *head* daje se na znanje da ta stranica ima ugrađenu eRDF sintaksu. Sljedeći primjer prikazuje korištenje eRDF-a:

```
<html>
  <head profile="http://purl.org/NET/erdf/profile">
    <meta name="dc.creator" content="Lidia" />
    <meta name="dc.title" content="eRDF primjer" />
    <link rel="schema.dc" href="http://purl.org/dc/elements/1.1/" />
    <link rel="schema.foaf" href="http://xmlns.com/foaf/0.1/" />
    <link href="#lidia" rev="foaf-homepage foaf-made" rel="foaf-maker" />
  </head>
  <body>
      Autor:
      <span class="foaf-name">
         <span class="foaf-firstName">Lidia</span>
         <span class="foaf-surname">Rovan</span>
       </span>
      e-mail
      <span class="foaf-mbox_sha1sum"
            title="69e31bbcf58d432950127593e292a55975bc66fd">
      lidia.rovan {at} fer.hr</span>.
  </body>
</html>
```

```
<http://primjer.org/about> dc:creator "Lidia" .
<http://primjer.org/about> dc:title "Lidia's Homepage" .
<http://primjer.org/about> foaf:maker <http://primjer.org/about#lidia> .
<http://primjer.org/about#lidia> foaf:name "Lidia Rovan" .
<http://primjer.org/about#lidia> foaf:firstName "Lidia" .
<http://primjer.org/about#lidia> foaf:surname "Rovan" .
<http://primjer.org/about#lidia> foaf:mbox_sha1sum
"69e31bbcf58d432950127593e292a55975bc66fd" .
```

Osnovne zamjerke korištenja eRDFa su što ne podržava tipove podataka, prazne čvorove te može definirati samo svojstva za stranicu na kojoj se implementira. Sve navedeno učinilo ga je potpuno marginalnom tehnologijom od pojave RDFa tehnologije koja odgovara na sve navedene probleme. Razvoj eRDF-a odvijao se 2005. i 2006. godine te je nakon toga potpuno zanemaren.

4.4.4. RDFa

RDFa [Adida2009] je tehnologija preporučena od strane W3C-a. Radi se o dijalektu XHTML-a, tehnologiji koja omogućava pripremu i objavljivanje sadržaja koji je ujedno razumljiv ljudima i računalima (slika 34). Ugradnja RDF izjava u XHTML omogućena je korištenjem proširenja XHTML-a – sintakse RDFa. U odnosu na mikroformate pruža fleksibilno i skalabilno rješenje, jer dozvoljava korištenje proizvoljnih vokabulara, a ne samo unaprijed definiranih. RDFa je kompatibilan s RDF-om, podržava sve XML imenike (eng. XML namespace), kao i CURIE (Compact URI).

RDFa atributi (postojeći XHTML atributi i specifični za RDFa) su sljedeći [Adida2009b]:

Atributi iz XHTML sintakse:

- **rel** i **rev** – CURIE ili lista ključnih riječi koji označavaju vezu i reverznu vezu između resursa (i predikata)
- **href** – URI resursa koji predstavlja objekt RDF trojke (pri tom ima ulogu i hiperveze)
- **src** – URI resursa koji predstavlja subjekt RDF trojke (ograničen na img XHTML element)
- **content** – atribut koji kad je naveden nadjačava sadržaj elementa XHTML-a

Atributi iz RDFa sintakse:

- **about** –URI ili CURIE koji označava subjekt na kojeg se odnose uvedeni meta podaci
- **property** – lista CURIE-a koji označavaju svojstvo, ili više njih, kojem odgovara sadržaj elementa XHTML-a
- **resource** – URI ili CURIE koji predstavlja subjekt RDF izjave
- **datatype** – CURIE koji označava tip podatka sadržaja
- **typeof** – lista CURIE-a koji označavaju RDF tip podatka subjekta

RDFa je jedina od navedenih tehnologija za ugradnju semantike u prezentacijski sloj koja omogućava unošenje semantike u obliku trojki. Obično se za postavljanje subjekta koristi atribut @about, dok se predikati postavljaju jednim od atributa: @rel, @rev i @property. Postavljanje objekata razlikuje dva slučaja. Kada se radi o objektima označenim URI-ima tada se mogu koristiti atributi: @resource, @href i @src., a za objekte koji su podatkovna vrijednost: @content ili sam sadržaj XHTML elementa za kojeg se uvodi semantika.

XHTML + RDFa (kod)

```
<table>
  <tr>
      <th>Rbr</th>
      <th>JMBAG</th>
      <th>Ime</th>
      <th>Prezime</th>
  </tr>
  <tr about="#0036390867" typeof="foaf:Person">
      <td>1</td>
      <td>0036390867</td>
      <td><span property="foaf:firstName">Jasmina</span></td>
      <td><span property="foaf:surname">Drmić</span></td>
  </tr>
  <tr about="#0036405741" typeof="foaf:Person">
      <td>2</td>
      <td>0036405741</td>
      <td><span property="foaf:firstName">Igor</span></td>
      <td><span property="foaf:surname">Gotovac</span></td>
  </tr>
</table>
```

XHTML prikaz (preglednik)

Rbr.	JMBAG	Ime	Prezime
1	0036390867	Jasmina	Drmić
2	0036405741	Igor	Gotovac

RDF izjave (nakon parsiranja)

```
#0036390867 foaf:firstName Jasmina.
#0036390867 foaf:surname Drmić.
#0036390867 rdf:type foaf:Person.

#0036405741 foaf:firstName Igor.
#0036405741 foaf:surname Gotovac.
#0036405741 rdf:type foaf:Person.
```

Slika 34 XHTML + RDFa - različiti prikazi

Budući da je RDFa brzo stekao popularnost, s njegovom većom primjenom razvijeni su dodaci sadašnjih internet preglednika, kao i sami RDFa preglednici. Na taj način omogućena je vizualizacija semantike pohranjene u prezentacijskom sloju. Prednost RDFa nad ostalim spomenutim tehnologijama prikazuje tablica 3, koja sadržava usporedbu karakteristika svih navedenih tehnologija.

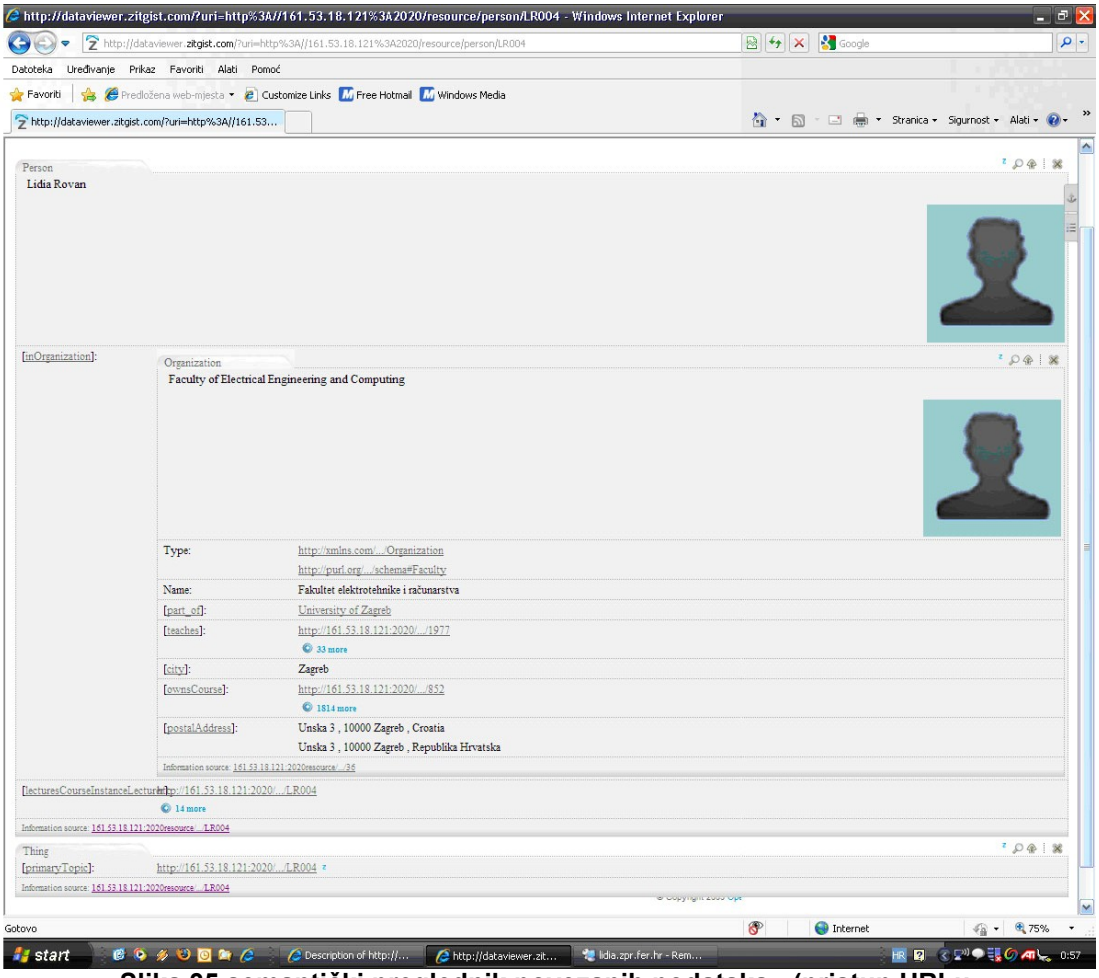
Tablica 3. Usporedba tehnologija za ugradnju semantike u prezentacijski sloj

	mikroformati	GRDDL	eRDF	RDFa
mogućnost korištenja proizvoljnih vokabulara	ne	da	da	da
stupanj izražajnosti	nizak	nizak	nizak	visok
semantika vezana uz prikazan podatak	da	ne	da	da
održavanje	teško	teško	lako	lako
vizualizacija	dodaci preglednika	dodaci preglednika	dodaci preglednika	preglednik

4.4.5. Preglednici semantičkog weba

GRDDL i RDFa stekli su veliku popularnost kod osoba uključenih u razvoj aplikacija. Da bi se mogla pokazati iskoristivost i primjenjivost na taj način semantički obogaćenih stranica potrebno je stvoriti alate koji bi bili u stanju konzumirati takav sadržaj i, ono što je najvažnije, uključiti čovjeka u proces korištenja takvih podataka. Iako se počelo s manjim rješenjima u obliku raznih modula za preglednike (npr. Operator modul za Firefox) pa i samih preglednika povezanih podataka (npr. Tabulator, Disco ili Zitgist DataViewer), ipak još nijedno rješenje ne zadovoljava ugodnost korištenja.

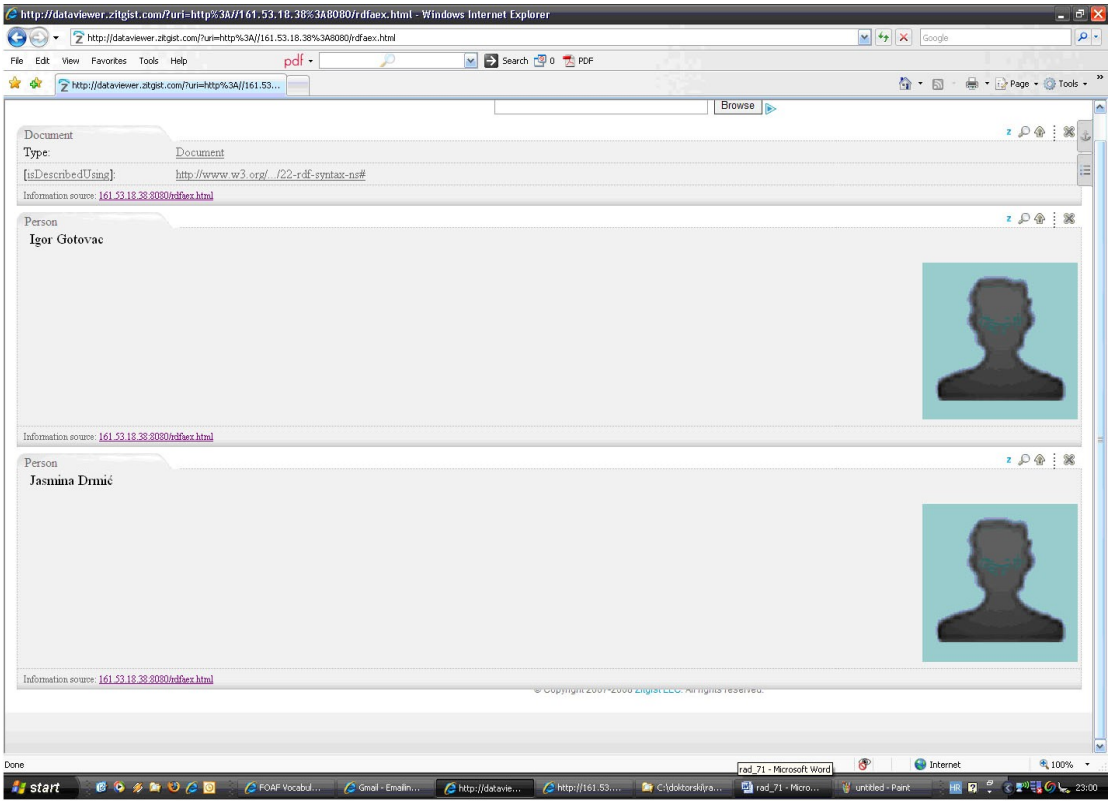
Glavna namjena semantičkog preglednika weba (eng. *semantic web browser*) je nehotična i nasumična ponovna upotreba podataka i informacija (eng. *data re-use*), koja se očituje usputnim otkrivanjem novih, nepoznatih ili na prvi pogled nevezanih informacija o predmetu zanimanja tijekom pretraživanja weba. Korištenjem takvih preglednika semantički web dobiva novu, korisnicima opipljivu dimenziju. Upravo je vidljivost na web postavljenog dokumenta bila jedan od ključnih razloga uspjeha klasičnog weba, a upravo ta vidljivost nije izvorno prisutna na semantičkom webu. Semantički preglednici weba sada se uglavnom koriste za pronalaženje objavljenih resursa na semantičkom webu, kako bi se utvrdilo jesu li dostupni te jesu li sve RDF veze ispravno zadane, što ih čini korisnim alatima u postupku objavljivanja resursa. Slika 35 prikazuje semantički web preglednik povezanih podataka.



Slika 35 semantički preglednik povezanih podataka - (pristup URI-u http://lidia.zpr.fer.hr:2020/resource/person/LR004)

RDFa preglednici logičan su nastavak sadašnjih preglednika, budući da je RDFa dijalekt XHTML-a te objedinjuje oba prikaza. No, jedan od problema RDFa preglednika je omogućavanje čitanja RDFa oznaka iz dijela stranice koji se osvježava Ajaxom. Naime, klijent ne može pristupiti tom dijelu DOM (Document Object Model) stabla, pa se tako ne može doći ni do oznaka. Rješavanje ovog problema trebalo bi biti vrlo visoko na listi prioriteta uzimajući u obzir zastupljenost Ajaxa u modernim aplikacijama. RDFa preglednici sigurno zadovoljavaju potrebe korisnika, ali ostaje problem što s podacima koji su rezultat GRDDL mehanizama, a koje RDFa preglednici opet zaobilaze. Istraživanje [Adida2008] se upravo bavi tom problematikom. Autori u tom radu predlažu način kako umjesto u RDF grafove taj sadržaj pretvarati u RDFa sadržaj. Na taj bi se način omogućilo da se sav istraživački trud usmjeri na RDFa preglednike koji su ključni za vizualizaciju i

kretanje korisnika u mreži podataka. Slika 36 prikazuje prikaz web dokumenta obogaćenog RDFa oznakama (slika 34) u RDFa pregledniku.



Slika 36 Semantički preglednik - RDFa prikaz

4.4.6. Programski okvir Jena

Jena [Jena2010] je programski okvir za izradu semantičkih web aplikacija programskim jezikom Java. Pruža programsko okruženje za rukovanje osnovnim semantičkim tehnologijama: RDF, RDFS, OWL, SPARQL, GRDDL i dr. Također, Jena se može koristiti i kao trajno spremište RDF podataka te sadrži mehanizam za zaključivanje koji se može koristiti nad tako trajno pohranjenim podacima.

5.Definicija i kategorije semantičkih web aplikacija

Prvo prikupi činjenice, a tek ih onda možeš izvrtati po želji.
Mark Twain

Osnovni je cilj ovog istraživanja predlaganje procesa prilagodbe postojećih web aplikacija za semantički web. Kako bi se mogao definirati cilj takve prilagodbe bilo je potrebno istražiti što karakterizira semantičku web aplikaciju i kako se ona definira. Poseban je naglasak stavljen na utvrđivanje razlike između web aplikacija i semantičkih web aplikacija. Na taj način dobila se jasna razlika između početnog stanja i konačnog cilja procesa prilagodbe.

Semantički web je od svog početka promjenjivo okruženje, pa rezultati rada u području inženjerstva semantičkih web aplikacija kao posljedica intenzivnih promjena u njegovom razvoju, brzo zastarijevaju. S ciljem da se utvrdi trenutno stanje inženjerstva semantičkih web aplikacija, u okviru je ovog istraživanja provedena analiza trenutnog stanja područja semantičkog weba, kao i tehnologija koje ga čine. Ustanovljeno je da su osnovni elementi potrebni za razvoj semantičkih rješenja standardizirani pa se može postaviti temelj inženjerstva aplikacija semantičkog weba. Također, provedena je analiza dostupnih semantičkih web aplikacija u cilju identifikacije učestalih slučajeva uporabe semantičkih tehnologija. Prepoznate su osnovne kategorije semantičkih web aplikacija ovisno o njihovoj namjeni, pa tako i funkcionalnostima koje sadržavaju. Na temelju rezultata analize, predložena je i razrađena arhitektura svake od prepoznatih kategorija. Arhitektura je realizirana u skladu s otkrivenim funkcionalnostima koje kategorija obično sadrži i tehnologijama koje su dio standarda povezanih u skladu s metodologijama novih pravaca istraživanja.

U ovom se poglavlju daje definicija semantičke web aplikacije te se ističe što je razlikuje od web aplikacije. Analiziraju se rezultati provedene analize postojećih semantičkih web aplikacija. Uvode se definicije osnovnih prepoznatih kategorija semantičkih web aplikacija te se elaboriraju učestali slučajevi uporabe, odnosno funkcionalnosti aplikacija.

5.1. Semantička web aplikacija

Da bi se mogla provesti analiza semantičkih web aplikacija neophodna je jasna definicija pojma semantičke web aplikacije. U prethodnim istraživanjima drugih

autora više puta je uvedena definicija semantičke web aplikacije [Hausenblas2007, Oren2008, Reif2005]. Nedostatak je tih definicija u tome što se ne grade na dovoljno precizno definiranim pojmom web aplikacije. U literaturi se često nailazi na razna shvaćanja o tome što je web aplikacija. Kako bi se izbjegla dvosmislenost, u poglavlju 2.1.2. uvedena je precizna definicija web aplikacije te se za svako pojavljivanje tog pojma u daljnjem tekstu smatra da podliježe uvedenoj definiciji. Uvodi se definicija koja razjašnjava što web aplikaciju čini ujedno i semantičkom web aplikacijom:

Definicija 5.1: *Semantička web aplikacija je web aplikacija čije se uspješno izvođenje zasniva na standardima semantičkog weba.*

U nastavku, iznose se razlike između web aplikacija i semantičkih web aplikacija kako bi se utvrdilo koji se primjeri iz prakse web inženjerstva mogu koristiti i pri inženjerstvu semantičkih web aplikacija.

5.1.1. Razlika web aplikacija i semantičkih web aplikacija

Da bi se utvrdila razlika web i semantičkih web aplikacija prvenstveno je potrebno napraviti usporedbu okruženja u kojima djeluju web aplikacije, odnosno semantičke web aplikacije. Ideje i načela weba i semantičkog weba sasvim su različita. Samim tim i realizacije nad takvim okruženjima imaju međusobna odstupanja. Tablica 4 prikazuje osnovne razlike spomenutih okruženja.

Tablica 4. Razlike u konceptima weba i semantičkog weba

	web	**semantički web**
jezgra	nestrukturirani sadržaj	formalne izjave
primarna ciljna skupina	ljudi	strojevi
veze	veze na dokumente (lokacija)	veze na podatke
logika	neformalna	deskriptivna

Uzevši u obzir osnovna načela semantičkog weba, semantičke web aplikacije u svom punom smislu trebale bi imati sljedeće karakteristike:

- **jezgru čine podaci** - centralna komponenta su podaci i čine osnovnu jezgru aplikacije

- **podaci imaju jasnu semantiku** – značenje se direktno daje podacima, a ne ugrađuje se u poslovnu logiku ili prepušta korisnicima da ga sami interpretiraju
- **integracija/dijeljenje podataka** – aplikacija bi trebala biti prilagođena za korištenje različitih informacijskih izvora s weba, te također omogućiti da druge aplikacije mogu koristiti njene podatke
- **dinamika podataka** – podaci se mogu mijenjati, na primjer donositi novi zaključci za vrijeme izvođenja aplikacije pa aplikacija mora biti u mogućnosti podržati fluktuirajući model podataka

Na osnovu navedenih karakteristika okruženja vidljivo je da su razlike suštinske. Očito je da se razvojni proces aplikacija koje djeluju u takvim okruženjima mora razlikovati. Važno je spomenuti da se razvojni proces web aplikacija već počeo mijenjati s Webom 2.0. Kako su Web 2.0 aplikacije prihvatile načelo dijeljenja i integracije podataka, odnosno već su okrenute prema podacima, za takve aplikacije razlika razvojnog procesa i korištenih tehnologija manja je nego za klasične web aplikacije.

Oren i suradnici [Oren2007b] iznose osnovne razlike sadašnjih web aplikacija i semantičkih web aplikacija. Kao razlike ističu karakteristiku centraliziranosti sadašnjih web aplikacija, korištenje jednog modela podataka, fiksne sheme i vokabulara za razliku od decentraliziranih semantičkih web aplikacija koje se temelje na različitim izvorima podataka, različitim shemama i proizvoljnim vokabularima. Tablica 5 prikazuje razlike koje su identificirali Oren i suradnici.

Tablica 5. Usporedba web aplikacija i semantičkih web aplikacija [Oren2007b]

web aplikacija	**semantička web aplikacija**
centralizirana	decentralizirana
jedna fiksna shema	polustrukturirani podaci
jedan fiksan vokabular	proizvoljan vokabular
centralizirano objavljivanje podataka	slobodno objavljivanje podataka
jedan izvor podataka	više distribuiranih izvora podataka
zatvoren sustav	otvoren sustav

Semantička web aplikacija ne mora zadovoljiti sve navedene kriterije niti je nužno da se razlikuje od sadašnjih web aplikacija po svim svojstvima koje prikazuje tablica 5. ***Da bi se neka aplikacija klasificirala kao semantička web aplikacija dovoljno je da upotrebljava semantičke tehnologije za rješavanje bilo kojeg dijela funkcionalnosti aplikacije.*** Naime, načela semantičkog weba u svom izvornom obliku nisu u skladu s realnim slučajevima uporabe aplikacija na webu. Za svaku se implementaciju ipak mora uvesti određena specijalizacija izvornih načela i prilagoditi je uvjetima u kojima aplikacija mora uspješno djelovati. Naime, ni osnovna postavka semantičkog weba - decentraliziranost - u nekim slučajevima ne mora biti zadovoljena. Na primjer, aplikacija rađena za potrebe suda u Španjolskoj [Casanovas2005] *zatvorena je aplikacija*, temeljena nad samo *jednim izvorom podataka* i semantičke tehnologije koristi u svrhu pretraživanja prirodnim jezikom, a ipak se nedvojbeno radi o semantičkoj web aplikaciji. U nekim slučajevima slobodno objavljivanje podataka također nije ostvarivo, odnosno ne smije se ostvariti, ali se semantičke tehnologije mogu iskoristiti u svrhu boljeg zaključivanja, pretraživanja, itd… Kao primjer aplikacija koje ne bi trebale olako objavljivati podatke su: aplikacije elektronskog bankarstva, aplikacije u području zdravstva koje posjeduju podatke pacijenta, aplikacije u području obrazovanja koje sadrže informacije o učenicima, i sl.

5.2. Dosadašnja istraživanja razvojnog procesa semantičkih web aplikacija

Uvriježeno je mišljenje da je implementacija semantike jako teška, pa i neostvariva u realnim uvjetima. Stoga je važno utvrditi razvojni proces semantičkih web aplikacija da bi se na taj način utjecalo na smanjenje krivulje učenja izgradnje semantičkih web aplikacija, što je posredno doprinos ovog istraživanja. Odstupanje od jedinstvenog pravca istraživanja posebno je utjecalo na područje definiranja inženjeringa semantičke web aplikacije. Prethodna istraživanja rezultirala su predloženim razvojnim metodologijama, ali je relevantnost takvih istraživanja brzo opala ponajviše zbog promjena same piramide semantičkih web tehnologija, a onda dodatno i zbog promjena u razvoju aplikacija općenito.

U literaturi se može pronaći više predloženih metodologija razvoja semantičkih web aplikacija [Corcho2006, Houben2003, Reif2005]. Na primjer, Hera metodologija [Houben2003] propisuje da se aplikacija dijeli na semantički, aplikacijski i prezentacijski sloj. U svom radu ne daju prijedlog za cjelokupni razvoj aplikacija,

nego se koncentriraju samo na generiranje prezentacijskog sloja. No, osim što ne koriste recentne tehnologije za označavanje u prezentacijskom dijelu, kao što je RDFa, koje znatno mijenjaju način izrade stranica, dodatan problem predstavlja što je navigacija ograničena na prethodno definirane putove. Iz tog se razloga metodologija može primijeniti samo na manji skup aplikacija, koje podržavaju određen skup scenarija.

Ovo se istraživanje temelji na postavci da je za identifikaciju koraka razvojnog procesa semantičkih web aplikacija potrebno prethodno napraviti kategorizaciju takvih aplikacija. Naime, smatra se da je generičko rješenje neostvarivo te je potrebno uvesti specijalizaciju, ovisno o namjeni aplikacije. Tako Corcho i ostali [Corcho2006] u svom radu predlažu proširenje postojećeg programskog okvira ODESeW za razvoj jedne od kategorija semantičkih web aplikacija - semantičkih web portala. Programski okvir se temelji na MVC obrascu te se nudi proširenje tog obrasca. Ali kako se radi o specifičnom programskom okviru, a i koriste tehnologije koje nisu dio standarda, kao što je upitni jezik RQL, ne može se uzeti kao dovoljno dobro rješenje za semantičke web portale općenito.

U sklopu ovog istraživanja napravljena je analiza semantičkih web aplikacija u svrhu definiranja osnovnih kategorija semantičkih web aplikacija. Slično istraživanje, analizu semantičkih web aplikacija s ciljem otkrivanja uobičajenih problema njihovog dizajna proveo je 2007. godine Heitmann [Heitmann2007a]. U svom istraživanju, Heitmann uzima u obzir i aplikacije koje ne zadovoljavaju definiciju web aplikacija. Dodatno, razmatrane aplikacije rađene su isključivo u znanstvene svrhe i razvijene u periodu do 2007. godine. Znači, prije pojave novih pravaca istraživanja pa uzorak tog istraživanja više nije reprezentativan. Također, Heitmann se koncentrirao samo na identifikaciju komponenti koje se često javljaju, neovisno u kojoj se kategoriji aplikacija one javljaju i kako se u tom slučaju međusobno kombiniraju. U ovom istraživanju pokazalo se da svaka kategorija aplikacija posjeduje specifičan skup funkcionalnosti i način primjene tehnologija za ostvarenje takvih funkcionalnosti.

5.3. Analiza postojećih web aplikacija

U sklopu godišnje europske i svjetske konferencije područja semantičkog weba održavaju se natjecanja programskih rješenja za semantički web. Također, W3C održava listu semantičkih web rješenja[17] koja se aktivno koriste u privredi, kao i onih koja su nastala za potrebe rješenja u privredi, ali trenutno nisu u produkciji. Kao

[17] http://www.w3.org/2001/sw/sweo/public/UseCases/

uzorak, odnosno empirijska osnova na temelju koje je napravljena analiza semantičkih web aplikacija uzeta su sljedeća rješenja:

- projekti s W3C liste koji se aktivno koriste u privredi
- projekti s natjecanja "Scripting for the Semantic Web"[18] održanih u sklopu europske konferencije u periodu od 2005. do 2009.
- projekti s natjecanja "Semantic web challenge"[19] održanih u sklopu svjetske konferencije u periodu od 2003. do 2008.

Kao osnovni cilj analize postavljena je kategorizacija semantičkih web aplikacija. Uz kategorizaciju cilj je bio i prepoznati najčešće slučajeve uporabe semantičkih tehnologija, odnosno funkcionalnosti aplikacija koje se realiziraju semantičkim tehnologijama. Također, potrebno je bilo prepoznati koje su od tih funkcionalnosti karakteristične za pojedine kategorije. Implementacije istih funkcionalnosti po različitim aplikacijama odstupaju, odnosno sadrže posebnosti koje uvjetuje problematika određene aplikacije. Dodatnom analizom nastojali su se prepoznati učestali uzorci dizajna te je tako za svaku funkcionalnost dano i njeno poopćenje.

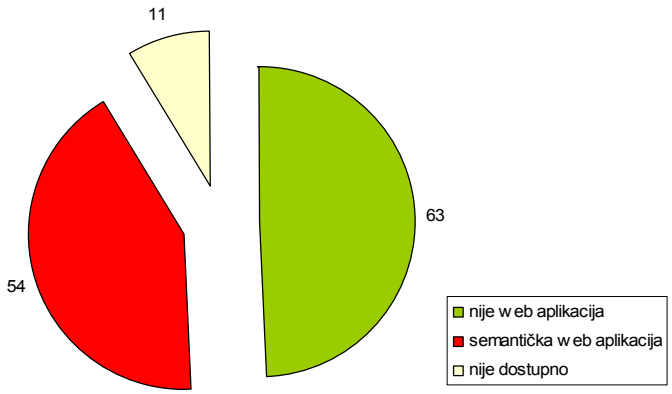
Osim ispunjenja osnovnih ciljeva istraživanja, rezultati analize potvrdili su i teoretsko objašnjenje razvoja semantičkog weba, odnosno pokazalo se da se ipak priklonilo novim pravcima istraživanja koji streme ostvarenju mreže podataka i mreže oznaka.

5.4. Rezultati analize

5.4.1. Uzorak

Uzorak je ukupno brojao 128 rješenja. Slika 37 prikazuje strukturu uzorka. Od ukupnog broja rješenja u uzorku, za njih 11 nije dostupan nikakav pisani materijal, niti su sama rješenja dostupna za uvid. Tako da je analizi podvrgnuto 117 rješenja, od kojih se ustanovilo da 54 zadovoljava uvjete definicije semantičke web aplikacije. Ostali dio aplikacija čine programska rješenja koja su ili samostojeće aplikacije, ili programski okviri, različita infrastruktura, itd… Detalji o samim aplikacijama sadržani su u dodatku - tablica 18.

[18] www.semanticscripting.org
[19] challenge.semanticweb.org/



Slika 37 Struktura uzorka

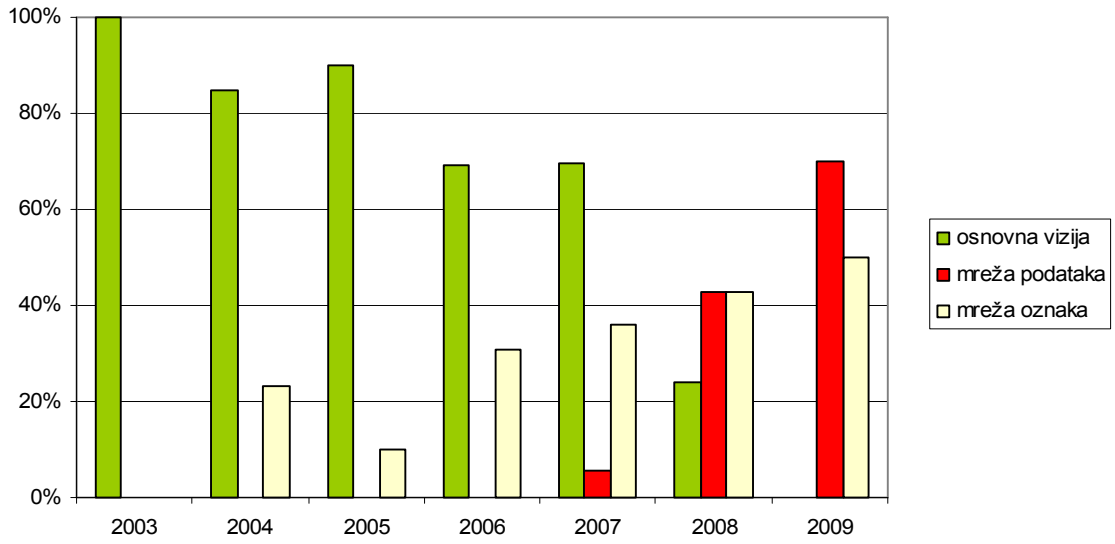
5.4.2. Vremenski tijek razvoja web aplikacija

Prethodno je već objašnjen razvoj semantičkog weba, njegov put od osnovne vizije do povezanih podataka (mreže podataka) i semantičkog označavanja (mreže oznaka). Tijekom analize rješenja, za svako rješenje za koje je to bilo moguće, odnosno za koje je bilo dovoljno informacija, utvrdilo se kojem pravcu istraživanja pripada. Vremenski prikaz nastanka rješenja uz pripadnost pravcu istraživanja potpuno potvrđuje prethodno iznesenu teoriju o zaokretu u području interesa pri razvoju semantičkog weba. Slika 38 prikazuje kako se pripadnost rješenja pojedinom pravcu istraživanja mijenjala kroz vrijeme. Stupci predstavljaju postotak aplikacija koji odgovara pojedinom pravcu istraživanja za svaku godinu. Za aplikacije koje su rezultati dugogodišnjih projekata pa su bile prijavljene na više natjecanja tijekom njihovog razvoja, uzeta je samo posljednja godina pojavljivanja, jer su se kroz to vrijeme prilagodili trendovima. Također, moguće je da je programsko rješenje rađeno u skladu s više pravaca istraživanja, npr. mreža oznaka i mreža podataka.

Pojednostavljeno, rješenje pripada određenom pravcu istraživanja ako ima sljedeće karakteristike:

- **osnovna vizija** – programska rješenja koja kao model podataka koriste ontologije pisane s visokom razinom ugrađene logike i koja su usmjerena na dobivanje dodatne koristi iz takvih modela korištenjem zaključivanja; programska potpora koja na neki način potpomaže razvoju osnovne vizije semantičkog weba (npr. Swoogle – pretraživač RDF i OWL podataka)

- **mreža podataka** – programsko rješenje je realizirano nad podacima koji su objavljeni u skladu s pravilima koje propisuje koncept "povezanih podataka"
- **mreža oznaka** – programsko rješenje se ne temelji na modelu podataka opisanog ontologijom ili povezanim podacima, već je usmjereno na označavanje podataka uglavnom korištenjem niske razine logike; označavanje obično obavljaju korisnici kroz prilagođeno sučelje



Slika 38 Pripadnost rješenja pravcima istraživanja

5.4.3. Kategorije semantičkih web aplikacija

Analizom zadanog uzorka semantičkih web aplikacija obavljena je njihova kategorizacija. Prije nego što se pristupilo samom analiziranju dostupnih materijala i probnih verzija aplikacija, određene su preliminarne kategorije. Kategorija aplikacije određuje se prema njenoj namjeni, odnosno funkcionalnostima koje pruža. Kako bi se mogle postaviti preliminarne kategorije napravljen je pregled dosadašnjih istraživanja koja su se bavila problematikom kategorizacija aplikacija.

Objavljeno je više radova koji su iznosili kategorizaciju klasičnih web aplikacija. Pregled je iznesen u radu Ziemera [Ziemer2009], u kojem je također provedena kategorizacija s ciljem poboljšanja razvojnog ciklusa. Zamjerka svim dostupnim kategorizacijama je što sve razmatraju i web sjedišta, a dodatan problem predstavlja što su aplikacije uzete u obzir u prethodnim istraživanjima rađene do 2002. godine pa u njima nisu uključeni noviji trendovi iz razvoja weba koji su omogućili nastanak novih kategorija, na primjer društvenih aplikacija. Heitmann u svom istraživanju

[Heitmann2007a] iznosi kategorizaciju semantičkih web rješenja prema tri kriterija: domena aplikacije, tip arhitekture i tip aplikacije. Arhitekturu dijeli na centraliziran poslužitelj s web sučeljem, decentraliziran poslužitelji s web sučeljem, samostojeća aplikacija i peer-to-peer sustav. Dakle, Heitmann također uzima u obzir i rješenja koja ne odgovaraju web aplikacijama. Tip aplikacije određuje prema funkcionalnostima koje aplikacija posjeduje te tvrdi da aplikacija istovremeno može pripadati u više tipova. Tipove aplikacija dijeli na: semantički portal, semantičko označavanje, semantički repozitorij, semantičko stvaranje, semantička samostojeća aplikacija i semantički skriptni jezik.

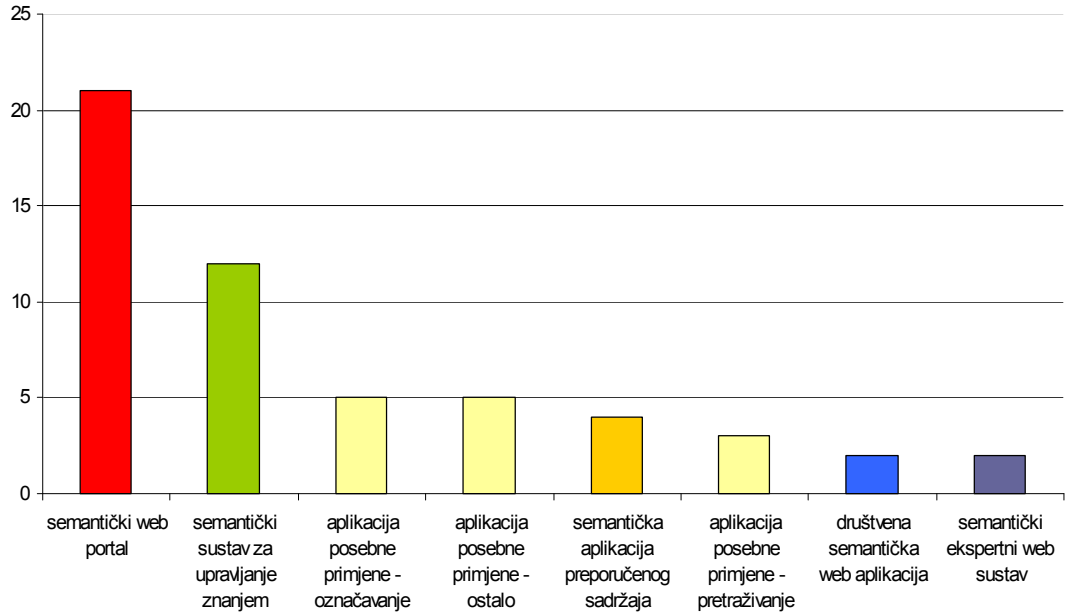
Za razliku od Heitmannovoh postavki, hipoteza ovog istraživanja je da se semantičke web aplikacije mogu kategorizirati prema njihovoj namjeni, neovisno o domeni kojoj pripadaju. Također, pretpostavlja se da svaka od tih kategorija sadrži određen skup funkcionalnosti te da namjena i funkcionalnosti određuju izgled arhitekture.

Konačno, na temelju prethodnih istraživanja formirane su sljedeće preliminarne kategorije: semantički web portal, semantički sustav za upravljanje znanjem, semantička aplikacija preporučenog sadržaja. U skladu s trendom razvoja weba, dodana je i kategorija semantička društvena web aplikacija. Za zadane kategorije uvedene su definicije u skladu s literaturom, odnosno teorijskim okvirima.

U prvom koraku obavljena je kategorizacija na probnom uzorku, što je rezultiralo revidiranjem kategorija. Zatim je provedena kategorizacija svih aplikacija odabranih za analizu. Pri analizi zanemarene su izjave samih autora o kategoriji aplikacija i one nisu utjecale na odluku procjenitelja, budući da se pokazalo da često odstupaju od teorijskih okvira. Kategorizacija je obavljena od strane dva različita procjenitelja te je došlo do dodatnog usklađivanja kategorija. U konačnici, formirane su sljedeće kategorije semantičkih web aplikacija:

- semantički web portal
- semantički sustav za upravljanje znanjem
- semantička aplikacija preporučenog sadržaja
- semantička društvena web aplikacija
- semantički ekspertni web sustav
- aplikacija posebne primjene koja semantičke tehnologije koristi u svrhu označavanja sadržaja
- aplikacija posebne primjene koja semantičke tehnologije koristi u svrhu boljeg pretraživanja

Slika 39 prikazuje zastupljenost analiziranih aplikacija po definiranim kategorijama.



Slika 39 Raspodjela web aplikacija po kategorijama

Uvjet da kategorija mora zadovoljiti teorijske okvire znači da se kod procjene aplikacija gledalo u kojoj mjeri aplikacija zadovoljava uvjete koji proizlaze iz definicija kategorija danih u nastavku. Svaka definicija jasno ocrtava namjenu aplikacije i funkcionalnosti koje se od nje očekuju.

5.4.3.1. Definicije kategorija

U ovom poglavlju uvode se definicije kategorija prepoznatih tijekom kategorizacije semantičkih web aplikacija.

Semantički web portal

Na temelju istraživanja Smitha [Smith2004] koje je proveo u cilju davanja opće definicije web portala:

> ***Definicija 5.2 a: Web portal*** *je web sustav koji služi kao personaliziran i prilagodljiv jedinstven pristup velikom skupu različitih izvora podataka koji su od značaja korisnicima portala.*

Analogno definiciji semantičke web aplikacije:

> ***Definicija 5.2 b: Semantički web portal*** *je web portal čije se uspješno izvođenje zasniva na standardima semantičkog weba.*

Budući da portal služi kao jedinstven pristup skupu različitih, heterogenih izvora podataka, funkcionalnost koju on obavezno mora posjedovati je *integracija*. Personalizacija i prilagodljivost je manje važna, budući da potpuno ovisi o domeni koju portal pokriva pa je onda i različit stupanj njihove implementacije. Semantički web portal obično koristi semantiku da bi se poboljšala integracija, pregledavanje, pretraživanje i personalizacija.

Semantički sustav za upravljanje znanjem

Dostupno je više pregledne literature na temu sustava za upravljanja znanjem [Alavi2001, Maier2004], dokumentima i sadržajem [Boiko2005]. Ponajviše u skladu s radom [Maier2004], sustav za upravljanje znanjem se smatra nadskupom svih navedenih oblika rješenja te se definira:

> ***Definicija 5.3 a: Sustav za upravljanje znanjem*** *je aplikacija i skup povezanih procesa koji određenoj organizaciji (bilo kojoj vrsti organizacije) ili grupi korisnika koja dijeli iste interese omogućuju identifikaciju, pohranu i dohvaćanje intelektualnog kapitala.*

U istraživanju koje su proveli Joo i Lee [Joo2009] prepoznata su osnovna tehnička ograničenja sustava za upravljanje znanjem na koja bi semantičke web tehnologije mogle utjecati, a to su *implementacija integracije više izvora podataka*, *implementacija pretraživanja, nezadovoljstvo korištenjem te nedovoljna dosljednost i potpunost znanja*. U istraživanju je istaknuto kako semantičke web tehnologije odgovaraju na otkrivene probleme. U skladu s istraživanjem Jooa i Leea, upravo se implementacija semantičkih tehnologija u sustavima za upravljanje znanjem nametnula kao jedna od učestalih kategorija semantičkih web aplikacija.

> ***Definicija 5.3 b: Semantički sustav za upravljanje znanjem*** *je sustav za upravljanje znanjem koji semantičke web tehnologije koristi u svrhu boljih rezultata kod integracije, pretraživanja, zadovoljstva korištenjem od strane korisnika i točnosti samog znanja kojim sustav rukuje.*

Sustavi za upravljanje znanjem imaju najčešću primjenu u privredi. Radi se o sustavima koje koristi grupa korisnika s ciljem međusobnog dijeljenja znanja. Osnovna funkcionalnost koja ih razlikuje od drugih aplikacija mogućnost je

pohrane i rukovanja različitim multimedijalnim sadržajima. Semantičke tehnologije kod ovih sustava najčešće se koriste za poboljšano označavanje dokumenata, što onda unaprjeđuje i njihovo dohvaćanje.

Semantička web aplikacija preporučenog sadržaja

Prema Resncicku i Varianu tipičan sustav preporučenog sadržaja se definira [Resncick1997]:

> ***Definicija 5.4 a: Sustav preporučenog sadržaja*** *je sustav putem kojeg ljudi unose preporuke sadržaja, koje onda sustav agregira i usmjeri odgovarajućim primateljima.*

Iz osnovne definicije izuzima se da su nužno ljudi ti koji preporuke unose, danas tehnologija dopušta razne načine identifikacije preporuka bez njihovog eksplicitnog unosa. Težina ovih aplikacija je u načinu izvedbe agregiranja preporuka i u pravilnom pronalaženju veza između preporuka i onih koji preporuke i traže. Upravo se primjenom semantičkih tehnologija na ta dva ključna problema može doći do poboljšanja operativnosti. U najjednostavnijim se slučajevima primjene semantičkih tehnologija koriste semantički opisi sadržaja ili korisnika u svrhu identifikacije dodatnih podataka metodama zaključivanja nad ontologijama. Složeniji semantički postupci koriste semantičke opise i u procesu predlaganja sadržaja, produbljivanjem mogućnosti analize podataka o sadržaju i korisnicima. Potencijal semantičkih postupaka leži u mogućnosti pronalaženja novog znanja o korisnicima, sadržaju i njihovoj međusobnoj povezanosti, što se može iskoristiti u postupku predlaganja sadržaja korisniku. U skladu s navedenim uvodi se sljedeća definicija:

> ***Definicija 5.4 b: Semantička web aplikacija preporučenog sadržaja*** *je web sustav preporučenog sadržaja u kojem se semantičke tehnologije koriste u svrhu provođenja analize na temelju koje se za korisnika pronalazi preporučen sadržaj.*

Web portali koji posjeduju funkcionalnost personalizacije često implementiraju i sam sustav preporuka, tako da se ova kategorija, u takvim uvjetima, može smatrati podskupom kategorije web portala. Odnosno, takav web portal je ujedno i sustav preporučenog sadržaja.

Semantička društvena web aplikacija

Društvena web aplikacija pojam je koji ima različita shvaćanja, obično šira od onih koja se definiraju u sklopu ovog istraživanja. Često se i pojam Web 2.0 smatra

ekvivalentom društvenih web aplikacija [O'Reilly2009]. Ovdje se uvodi stroža, uža definicija društvenih aplikacija orijentiranih društvenoj mreži korisnika. ***Dakle, aplikacija je usmjerena na komunikaciju i na kolaboraciju među korisnicima.*** Sve aplikacije u kojima interakcija između korisnika nije okosnica isključene su iz ove kategorije. Primjer je Wikipedia koja se često navodi kao primjer društvene aplikacije, no ona ipak pripada kategoriji sustava za upravljanje znanjem, budući da ne sadrži nikakav vid interakcije između korisnika.

> ***Definicija 5.5 a: Semantička društvena web aplikacija*** *je web aplikacija orijentirana društvenoj mreži korisnika koja koristi semantičke tehnologije u svrhu poboljšavanja suradnje/interakcije među njenim korisnicima.*

Društvene web aplikacije noviji su oblik aplikacija. Te su aplikacije usmjerene na bilo kakav oblik interakcije između korisnika pojedinaca, kao i unutar grupa korisnika. Budući da je FOAF doživio veliku popularnost kod semantičkog weba, upravo se on često koristi kao sredstvo koje omogućuje distribuirane korisničke profile, odnosno implementaciju društvene mreže korisnika.

Semantički ekspertni web sustav

Prema Brooksu [Brooks1987] ekspertni sustav se definira kao:

> ***Definicija 5.6 a: Ekspertni sustav*** *je računalni sustav koji sadrži mehanizam za zaključivanje kao i bazu pravila te koji na temelju ulaznih podataka i pretpostavki koristeći definirana pravila korisniku nudi zaključke i savjete.*

Dakle, ekspertni sustav je računalni sustav koji tijekom rješavanja kompleksnog zadatka simulira proces razmišljanja eksperta iz određenog područja. Ekspertni sustav koristi se pri rješavanju problema ili pri donošenju odluka. Svrha im je da zamijene eksperta na tom području, njegovo znanje i posao koji obavlja. Pravila koja se definiraju sustavom predstavljaju znanje eksperta, a zaključivanje nad tim istim pravilima predstavlja posao eksperta.

Semantičke tehnologije kod izrade ekspertnog sustava mogu se koristiti za definiranje baze pravila i zaključivanja nad njima, na primjer korištenjem OWL-a. Također, budući su na taj način podaci semantički označeni poboljšana je i njihova interpretacija.

> ***Definicija 5.6 b: Semantički ekspertni web sustav*** *je ekspertni sustav realiziran kao web aplikacija, koji semantičke tehnologije koristi u svrhu definiranja baze pravila, zaključivanja nad tim pravilima te poboljšanja interpretacije podataka.*

Ova kategorija aplikacija koristi se uglavnom u specifičnim domenama, budući da zamjenjuje eksperte područja pa se ne može ni očekivati primjena na širokom području. Česta primjena je u medicini.

Aplikacije posebne primjene

Preostale definirane kategorije čine web aplikacije koje su zamišljene kao potpora rješavanju specifičnih poslovnih problema neke domene pa za njih nije ni moguće poopćenje u jednu od prethodno spomenutih kategorija. Takve aplikacije dijele se dalje u kategorije ovisno o načinu primjene semantičkih tehnologija.

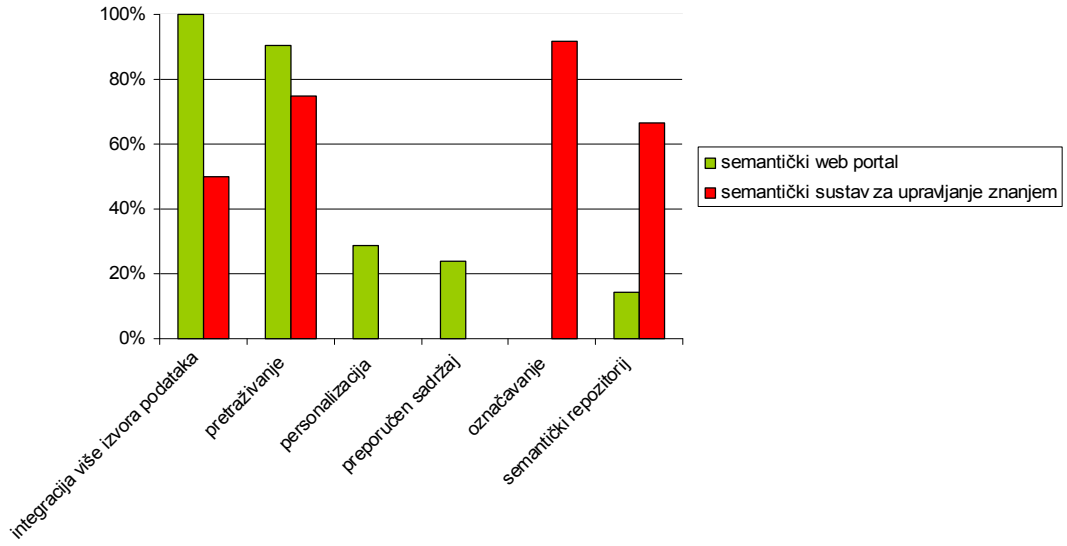
- koriste semantičke tehnologije u svrhu boljeg pretraživanja i integracije
- koriste semantičke tehnologije u svrhu označavanja podataka (podacima se daje značenje pristupom od vrha prema dnu)

Definicija 5.7 a: Semantička aplikacija posebne primjene *je web aplikacija koja je orijentirana rješavanju specifičnih poslovnih problema neke domene, a potpomognuta je semantičkim tehnologijama*.

5.4.4. Funkcionalnosti semantičkih web aplikacija

Ono što je otkriveno tijekom kategorizacije aplikacija je da se iste funkcionalnosti javljaju kod više kategorija pa se ne mogu uzimati pojedinačno kao prediktor kategorizacije. Odnosno, funkcionalnosti aplikacija po kategorijama ne čine disjunktne skupove. Na primjer, pretraživanje je karakteristika i web portala i sustava za upravljanje znanjem kao što pokazuje slika 40. Znači, za svaku kategoriju važno je gledati cjelinu koju te funkcionalnosti čine, na koji način su međusobno povezane i koju korist pružaju krajnjem korisniku. Tijekom analize važno je bilo prepoznati slučajeve uporaba semantičkih tehnologija. Slika 40 prikazuje koliki postotak aplikacija u kategorijama semantički web portal i semantički sustav za upravljanje znanjem ima implementirane otkrivene učestale funkcionalnosti. Funkcionalnosti koje se učestalo implementiraju semantičkim web tehnologijama su:

- integracija više izvora podataka
- pretraživanje
- personalizacija
- preporučen sadržaj
- označavanje
- semantički repozitorij



Slika 40 Zastupljenost funkcionalnosti po kategorijama

5.4.5. Poopćenje funkcionalnosti semantičkih web aplikacija

Kod izrade prijedloga arhitekture kategorija semantičkih web aplikacija važno je osigurati da ona podržava sve osnovne funkcionalnosti kategorije koju predstavlja. Da bi predložena arhitektura podržala što više implementacija, potrebno ju je izraditi na višoj razini apstrakcije. Stoga se nakon što su prepoznate kategorije semantičkih web aplikacija, kao sljedeći korak nameće poopćenje funkcionalnosti kako bi se mogla izraditi arhitektura na dovoljnoj razini apstrakcije. U nastavku slijede objašnjenja i poopćenja funkcionalnosti semantičkih web aplikacija.

5.4.5.1. Integracija više izvora podataka

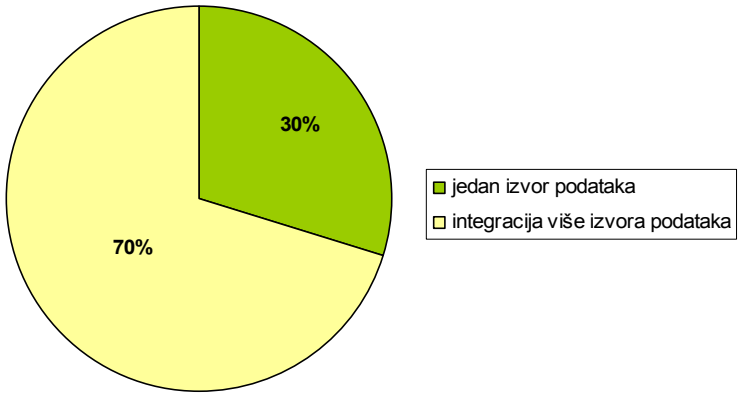
Integracija više izvora podataka realizirana je u ukupno 38 aplikacija od 54 analizirane. Slika 41 prikazuje postotak aplikacija s implementiranom integracijom. Kako je prethodno objašnjeno kod pojma mreže podataka proces integracije se odvija u dva koraka:

1) pronalaženje izvora podataka na koje se treba povezati

2) definiranje veza između podataka

Što se tiče specificiranja skupa podataka koji se integrira, odnosno što se tiče načina agregacije takvih izvora, u samim počecima razvoja semantičkih web aplikacija obično se razvijao robot koji je tražio semantičke podatke po webu. Ali, nakon 2007. godine nijedna od analiziranih aplikacija ne implementira robot kao sredstvo dohvaćanja podataka. Primjena povezanih podataka potrebu za takvom

implementacijom i isključuje. Budući da automatsko otkrivanje potencijalnih izvora podataka još uvijek nije zaživjelo, iako je ta ideja prisutna od samog početka razvoja semantičkog weba, izvori se uvijek moraju eksplicitno navesti.



Slika 41 Udio aplikacija nad integriranim podacima

Postoji više načina integracije koji se moraju razlikovati:

- integriranje nestrukturiranih izvora podataka
- integriranje strukturiranih izvora podataka
- integriranje semantički označenih izvora podataka

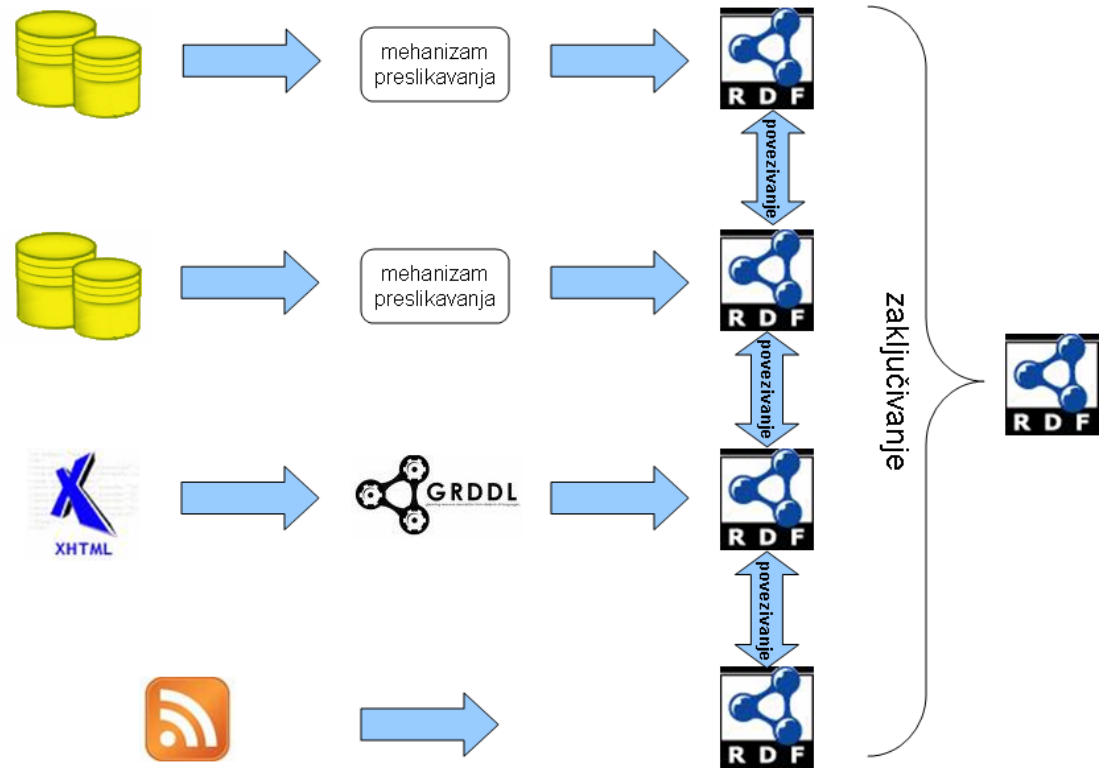
Poopćenje integracije nestrukturiranih izvora podataka

Poopćenje integracije teško je izvedivo u slučaju integriranja nestrukturiranih izvora podataka. Nestrukturirani izvori podataka obično su tekstualni dokumenti. Kako bi se iz tekstualnih dokumenata dobilo značenje sadržaja nad njima se primjenjuju razne metode iz područja pretraživanja informacija (eng. Information retrieval). Naravno, odabir algoritama i načina na koji će se koristiti, ovisi o jeziku na kojem su pisani i vrsti sadržaja. CS AKtive Space [Shadbolt2004] može se uzeti kao primjer aplikacije koja se bazira nad heterogenim nestrukturiranim izvorima podataka. U tom projektu razvili su poseban servis za svaku vrstu izvora podataka te se naglašava kako je svaki izvor morao biti posebno tretiran. Flink [Mika2005] je također dobar primjer integracije izvora podataka različitih formata, budući da integrira tekstualne podatke kao nestrukturirane - elektronsku poštu, strukturirane - podatke o publikacijama (google scholar), kao i semantičke – korisniče profile (FOAF). U spomenutom projektu integracija je također realizirana tako da se svaki izvor posebno tretira.

Sustavi za upravljanje znanjem u velikom postotku rukuju nestrukturiranim podacima, s razlikom da se kod njih sadržaj uglavnom ne povlači iz drugih izvora, nego se stvara kroz sustav. U tom slučaju, učestala je metoda integracije unos oznaka i opisnih semantičkih podataka kroz razna prilagođena sučelja. Opet, i tu je teško predložiti poopćenje budući da su sučelja specifična domeni aplikacije.

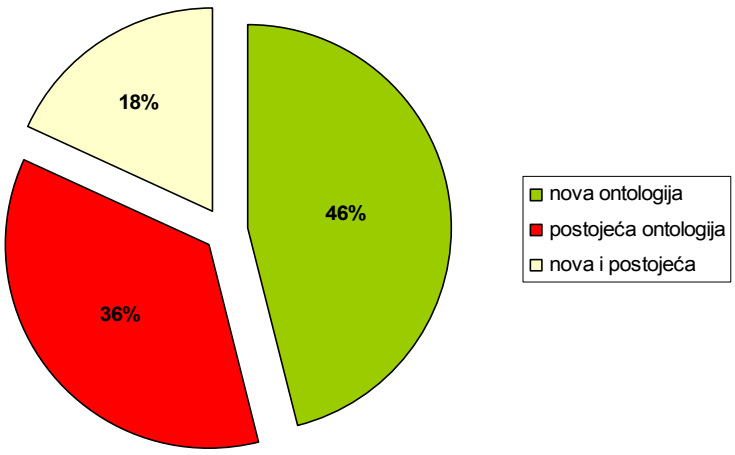
Poopćenje integriranja strukturiranih i semantički označenih podataka

Integriranje strukturiranih izvora podataka znatno je lakše jer već postoje brojna rješenja za transformiranje baza podataka, XML dokumenata i sl. u semantički označene podatke. Na tako transformirane podatke primjenjuju se metode koje se koriste i za posljednji način integriranja podataka, semantički označenih podataka. Slika 42 prikazuje proces integriranja strukturiranih, polu-strukturiranih i semantičkih podataka.



Slika 42 Integriranje heterogenih izvora podataka

Da bi se mogli semantički povezati različiti izvori podataka, potrebne su ontologije. Danas postoji niz iskoristivih ontologija, ali njihova ponovna iskoristivost ovisi o domeni aplikacije koja se realizira. Slika 43 prikazuje u kolikoj su se mjeri koristile postojeće ontologije u semantičkim web aplikacijama.

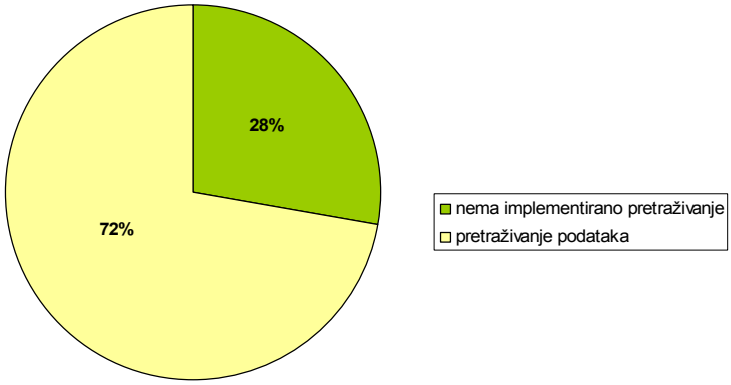


Slika 43 Korištenje ontologija (vokabulara) u web aplikacijama

Dakle, nakon što se pronađe odgovarajuća ontologija ili se kreira vlastita, ovisno o potrebama aplikacije, provodi se pretvorba podataka pohranjenih u relacijskim bazama, XML-u ili drugim strukturiranim formatima u semantičke zapise. Samo preslikavanje podataka obavlja se nekim od postojećih mehanizama, npr. D2RQ, Virtuoso, GRDDL, itd… Preslikavanje relacijskih modela u semantičke izjave, kao i postojeći alati koji služe kao podrška tom preslikavanju, biti će objašnjeni kasnije u poglavlju 7.5. Nakon što se svi izvori opišu ontologijama, potrebno ih je međusobno povezati. Obično se svi izvori podataka ne preslikaju odmah u istu ontologiju, pa je potrebno izvršiti povezivanje shema ontologija. Za primjer se opet uzima projekt Flink, gdje se podaci o publikacijama opisuju SWRC (Semantic Web for Research Communities) ontologijom, a profili FOAF vokabularom. Autora publikacije opisanog u SWRC-u treba moći povezati s njegovim profilom u FOAF-u. Uobičajena je izvedba uvođenjem „ontologija preslikavanja“ koje sadržavaju veze između pojmova korištenih ontologija. No, s tim korakom povezane su samo sheme. Dalje, potrebno je prepoznati sve iste instance. Da bi se točno povezao autor publikacije sa svojim profilom koristi se zaključivanje temeljeno na prepoznavanju jedinstvenih svojstava (identificirajućih URI-a). Za spomenut primjer projekta Flink, povezivanje se obavlja na način da se zahtijeva da e-mail adresa autora publikacije odgovara onoj iz njegovog profila. Kada se ne može uvesti eksplicitno pravilo za traženje istih instanci, koriste se metrike sličnosti.

5.4.5.2. Pretraživanje

Jako velik broj aplikacija, čak 39 od 54 analizirane, koristi semantičke tehnologije u svrhu izrade modula za poboljšano pretraživanje. Slika 44 prikazuje postotak analiziranih aplikacija s implementiranim pretraživanjem. No, implementacija takvih modula teško se može poopćiti. Iako su najčešća „hibridna rješenja“ – kombinacija klasičnog pretraživanja korištenjem ključnih riječi uz pretraživanje potpomognuto ontologijom (ključne riječi su na neki način opisane ontologijom), koraci koji se poduzimaju da bi se ostvarilo pretraživanje su ipak kontekstno ovisni. Priprema podataka, izdvajanje, označavanja sadržaja, algoritmi sličnosti, itd… ovise o vrsti sadržaja kojim se raspolaže.



Slika 44 Udio aplikacija s implementiranim pretraživanjem

Specifični slučajevi, zahtijevaju i primjenu specifičnog programskog rješenja, odnosno algoritma pretraživanja. Na primjer, u aplikaciji AnnoTerra [Ramagem2004] pretražuje se tekst te se pretraživanje ostvaruje eksplicitnim uvođenjem ključnih riječi koje su povezane s pojmovima opisanim ontologijom. Budući da se radi o pretraživanju teksta, nad njim se provode razni algoritmi iz područja pretraživanje informacija za filtriranje teksta (npr. Porter Stemmer filter, algoritam za normalizaciju riječi, heuristički algoritmi, itd…) Aplikacija SWEET [Raskin2005] također je koristila rješenje iz AnnoTerra projekta uz dodatnu prilagodbu ulaznih podataka u mehanizam koji obavlja pretraživanje.

Iako ograničenih mogućnosti, popularni su i vizualni pristupi pretraživanja – korisnici na ekranu odabiru pojmove i definiraju uvjete pretrage. Uvriježen pristup vizualnog pretraživanja je facetno pretraživanje (eng. faceted search), koje je

implementirano u ukupno 8 aplikacija (MuseumFinland [Hyvönen2005], PressIndex [Amardeilh2006], int.ere.st [Kim2007], MultimediaN E-Culture demonstrator [Schreiber2006], RKB explorer [Glaser2007], iFanzy [Bellekens2007], SIOC explorer [Heitmann2007b] i CultureSampo [Hyvonen2008]). Facetno pretraživanje je eksplorativna tehnika za strukturirane skupove podataka, koja se temelji na teoriji faceta [Oren2006]. Prikladno ga je koristiti u slučajevima kada se ne očekuje da je korisnik upoznat sa strukturom skupa podataka koji pretražuje. Kod ovog pretraživanja informacijski prostor se klasificira facetama, koje se prikazuju korisniku, te odabirom vrijednosti ponuđenih faceta korisnik filtrira dostupne informacije. Prednosti facetnog pretraživanja su:

- hijerarhije faceta daju pregled o tome kakve se informacije nalaze u repozitoriju
- hijerarhije mogu voditi korisnika pri formuliranju upita tako da koristi odgovarajuće pojmove

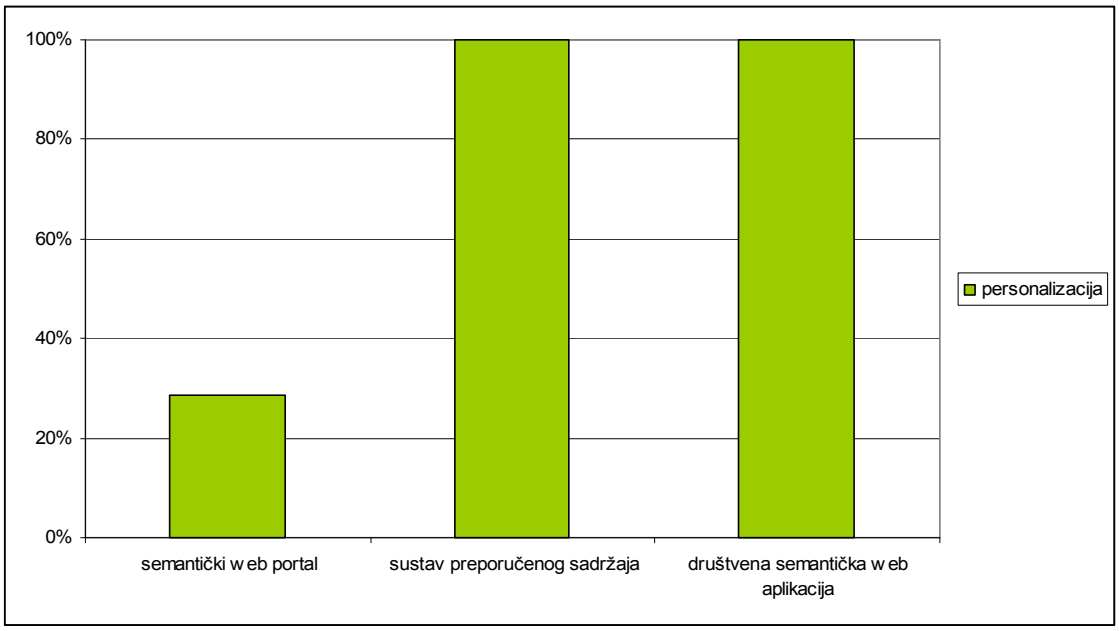
Odabir faceta koje će se nuditi na sučelju opet ovisi o domeni aplikacije. Izrada algoritama koji pretražuju po facetama, odnosno sužavaju odabir, ima uobičajene obrasce [Oren2006].

U aplikacijama posebne primjene, kao što je to npr. SWEET [Raskin2005], pretraživanje ne slijedi nikakve globalne formate i rađeno je specifično za određen oblik primjene.

Budući da je problematika pretraživanja zahtjevno područje i pripada posebnoj disciplini, a dodatno, konvergiranje jedinstvenom rješenju nije uočeno niti se smatra ostvarivim, ova funkcionalnost isključit će se iz predloženih arhitektura.

5.4.5.3. Personalizacija i preporučen sadržaj

Personalizacija, koja može obuhvaćati i preporučen sadržaj vrlo je kompleksna tema. Funkcionalnost personalizacije, uz preporučen sadržaj, dio je gotovo svakog portala i društvenih web aplikacija. Uzevši u obzir tri kategorije kod kojih personalizacija naročito ima smisla, a to su semantički web portal, sustav preporučenog sadržaja i društvena semantička web aplikacija, od ukupno 27 aplikacija njih 12 ima implementiranu personalizaciju što čini 44% aplikacija. Slika 45 prikazuje zastupljenost aplikacija s implementiranom personalizacijom u spomenutim kategorijama.



Slika 45 Zastupljenost personalizacije po kategorijama aplikacija

Cilj personalizacije je oblikovanje sustava prema željama i zahtjevima korisnika implicitno od strane sustava [Mulvenna2000]. Personalizacija je više od same preporuke sadržaja, što predstavlja samo jednu tehniku personalizacije. Osnova personalizacije je proces koji se odvija za vrijeme korištenja aplikacije od strane korisnika, a sastoji se od prikupljanja podataka o njegovim preferencijama, koji se kasnije koriste da bi mu se prilagodilo ponašanje sustava. Na primjer postavljanje jezika, lokacije, prilagođavanje izbornika, smanjivanje interakcije, i sl.

Pregled pristupa personalizacije dostupan je u radu Ananda i ostalih [Anand2005]. U analiziranim aplikacijama implementirani su razni načini ostvarenja personalizacije i preporučenog sadržaja objašnjenih u radu Ananda, te se ne može istaknuti uobičajen obrazac. Utvrđeno je da je osnovni problem svih pristupa nedovoljna razina sigurnosti, a budući da se rukuje korisničkim profilima, dakle osobnim podacima, problem se ne može zanemariti. Dati ikakvu detaljniju evaluaciju implementiranih rješenja prilično je teško, budući da se radi o prototip aplikacijama, koje onda kao posljedicu imaju nedovoljan broj korisnika i podataka. U opisu projekta MusiDB [Stegers2006] eksplicitno se navodi problem nedostatka korisnika i podataka kao uzroka što se ne može dati kvalitetna evaluacija.

5.4.5.4. Označavanje – uvođenje semantike

U analiziranim aplikacijama prepoznata su dva osnovna načina označavanja sadržaja:

- automatsko prepoznavanje i označavanje pojmova (nestrukturiranog) sadržaja
- unošenje semantičkih oznaka od strane korisnika

Za automatsko označavanje sadržaja teško je napraviti poopćenje rješenja. Upravo kao što je opisano kod integracije i pretraživanja, radi se o domenski ovisnim problemima usko vezanim uz disciplinu pretraživanje informacija pa je rješenja teško generalizirati. Naime, naglasak je na dohvaćanju informacija iz raznih oblika zapisa, a za to su potrebne posebne metode spomenutog područja.

Drugu vrstu označavanja, koja pripada pravcu istraživanja mreže oznaka, provode sami korisnici. U aplikacijama su prisutni različiti načini implementiranja označavanja. Usporedba postojećih metoda dana je u radu Kima i drugih autora [Kim2008]. U nastavku, navode se primjeri implementacija koji su otkriveni tijekom analize aplikacije.

Projekti Revyu.com [Heath2007] i GroupMe [Abel2007] implementiraju slobodno označavanje sadržaja, koje omogućava stvaranje oznaka autorima, vlasnicima ili korisnicima resursa te koje rezultira s neformaliziranim i neuređenim sustavom oznaka, odnosno nejednoznačnosti oznaka. Formalno rečeno, takvim pristupom sadržaj se označava oznakama koje se opisuju trojkama: ***označavanje: (Korisnik, Izvor, Oznaka)***. Zamjerka je tom pristupu nedostatak pravog značenja. Dakle, označava li oznaka "apple" jabuku ili New York. Rješenje na navedene probleme spomenuto je u prethodnom poglavlju. Radi se o tehnologiji MOAT, koja je također rezultat projekta uključenog u ovu analizu - „Enhancement and Integration of Corporate Social Software Using the Semantic Web“. MOAT uvodi pravu semantiku u označavanje i potpuno eliminira dvosmislenost oznaka korištenjem četvorki: ***označavanje: (Korisnik, Izvor, Oznaka, Značenje)***. S MOAT-om dolazi još niz prednosti, kao predložena infrastruktura klijent – poslužitelj. Na poslužitelju se prikupljaju sve kreirane oznake pa se na taj način nude korisniku za ponovno korištenje. U slučaju kada treba kreirati novu oznaku moguće je ostvariti vezu s pretraživačem Sindice, što je u skladu s konceptom mreže podataka. Na taj način mogu se lako pronaći novi URI-i s opisom značenja. Označavanje korišteno u projektu Faviki [SWCSUC2009] idejno je slično MOAT-u, ali pruža samo podskup

funkcionalnosti MOAT-a. Svakako je potrebno spomenuti i projekt int.ere.st [Kim2007] koji za semantičko označavanje koristi SCOT (Social Semantic Cloud of Tags) ontologiju. Taj pristup je u potpunosti okrenut folksonomijama i dijeljenju oznaka pa je pogodan za korištenje u društvenim aplikacijama. No, opet sadržava problem nejednoznačnosti, tako da bi za potpuno rješenje trebala integracija s MOAT-om. Ontologija SCOT sadrži pojmove potrebne za integriranje oznaka različitih korisnika, kreiranih u različitim aplikacijama za različite izvore, dok MOAT daje pravo značenje takvim oznakama.

5.4.5.5. Semantički repozitorij

Semantički repozitorij postao je dio gotovo svake semantičke web aplikacije. Iako je u samim počecima razvoja semantičkih web aplikacija bilo problem realizirati trajno spremište podataka, sada je tehnologija sazrela i implementacija ne predstavlja problem. Sve o trajnim spremištima i pristupu podacima pohranjenim u takvim spremištima objašnjeno je u poglavlju 3.1.4.4. Aktualan problem prisutan kod korištenja trajnih spremišta podataka, a posebno važan u kontekstu razvoja semantičkih web aplikacije je kontrola pristupa podacima. Samo jedan projekt naglašava taj problem „A Linked Open Data Resource List Management (RLM) Tool for Undergraduate Students“ [Clarke2009] i rješava ga fizičkim razdvajanjem podataka. Zanimljivo je da svi ostali projekti, iako posjeduju privatne podatke na koje bi svakako trebalo primijeniti mehanizam zaštite, potpuno ignoriraju taj problem. Premda postoje rezultati istraživanja na tom području [Bizer2009b, Dietzold2006, Jain2006, Manjunath2008, Reddivari2007], dostupnih implementacija još uvijek nema tako da je teško evaluirati dosad napravljeno. Također, za korištenje tih rješenja potrebno je znanje o načinu njihove realizacije, odnosno nisu usmjereni na krajnjeg korisnika što ne obećava njihovu skoru primjenjivost u praksi. Na primjer, niti jedan predložen pristup ne nudi sučelje putem kojeg bi se jednostavno upravljalo pravilima kontrole pristupa.

6.Arhitektura semantičkih web aplikacija

Sve treba pojednostaviti koliko je god moguće, ali ne više od toga.
Albert Einstein

Jedan je od ciljeva ovog istraživanja razrada arhitekture semantičkih web aplikacija. Predložiti jedinstvenu arhitekturu za sve semantičke web aplikacije moguće je samo na jako visokoj razini apstrakcije. U ovom istraživanju predlaže se arhitektura na nižoj razini apstrakcije, koja je onda i korisnija kao predložak za razvoj aplikacija. Kako bi se mogla definirati arhitektura na nižoj razini apstrakciji, potrebno je uvesti određenu specijalizaciju njene namjene. Upravo iz tog razloga prethodno je obavljena kategorizacija semantičkih web aplikacija. Dobivene kategorije predstavljaju specifične slučajeve primjene semantičkih web tehnologija. Za svaku kategoriju predlaže se arhitektura te se obrazlažu razlozi odabira elemenata i metodologija njihovog povezivanja. Tijekom izrade poopćenja funkcionalnosti, koje arhitekture moraju podržati, otkriveni su nedostaci njihovih implementacija u postojećim aplikacijama te se kroz razradu arhitekture predlažu rješenja za uočene nedostatke. Kao nepredviđen doprinos ovog istraživanja predložen je sustav za upravljanje identitetom u okruženju semantičkog weba. Taj sustav predstavlja komplementarno rješenje predloženim arhitekturama aplikacija kako bi zajedno činili rješenje koje može u svim segmentima odgovarati izazovima današnjih modernih web aplikacija.

Budući da je u području semantičkog weba napravljen zaokret u pravcima istraživanja nastojalo se ostvariti da predložena rješenja zadovoljavaju funkcionalnosti prepoznatih kategorija te da su pri tom u skladu s novim pravcima istraživanja, odnosno da podržavaju stvaranje mreže podataka i mreže oznaka.

U ovom se poglavlju definira značenje arhitekture aplikacija te se objašnjava proces izrade arhitektura korišten tijekom istraživanja. Potom se elaboriraju predložene arhitekture i nude odgovori na očekivane probleme pri implementaciji. Detaljno se opisuje i prijedlog sustava za upravljanje identitetom na kojeg se naslanjanju predložena rješenja.

6.1. Izrada arhitekture semantičkih web aplikacija

Definicija i detaljnija razrada arhitekture semantičkih web aplikacija zahtijeva prethodnu definiciju arhitekture aplikacija općenito. Dakle, potrebno je definirati što arhitektura predstavlja i kakva je njena uloga u razvojnom procesu aplikacija.
Prema [Perry1992] arhitektura softvera se definira:

> ***Definicija 6.1: Arhitektura softvera*** *je konfiguracija arhitektonskih elemenata i veza među njima koje koordiniraju aktivnosti elemenata s ciljem da se udovolji željenom skupu značajki postavljenih na sustav.*

Slika 46 Proces definiranja arhitekture semantičkih Java web aplikacija temeljenih na MVC obrascu

Kao smjernice pri razradi arhitekture postavljeni su uvjeti koje arhitektura mora zadovoljiti. Slika 46 opisuje proces definiranja arhitekture semantičkih web aplikacija korišten u ovom istraživanju. MVC (model-izgled-kontroler) obrazac je odabran kao temelj za dizajn arhitekture, a tehnologija kojoj se arhitektura posebno prilagođava je programski jezik Java. Razlog takvog odabira je što je definicija i razrada arhitektura semantičkih web aplikacija posredan doprinos ovog istraživanja koje je primarno usmjereno na izradu procesa prilagodbe semantičkom webu postojećih Java web aplikacija temeljenih na MVC obrascu. Također, nastojalo se ostvariti da predložena arhitektura zadovoljava trendove razvoja web aplikacija i semantičkog weba pa je kroz prethodna poglavlja napravljen pregled tih područja.

Rezultati analize semantičkih web aplikacija pokazali su da većina aplikacija ne implementira sve funkcionalnosti koje su otkrivene kao najčešći slučajevi uporabe semantičkih tehnologija pojedinih kategorija, što se očekuje da je slučaj i u praksi. Ipak, kod izrade arhitekture predlaže se generičko sveobuhvatno rješenje koje podržava sve uočene učestale funkcionalnosti kategorije za koju se radi. Takva se odluka temelji na uvjerenju da je lakše stvarati parcijalna rješenja za pojedini slučaj primjene, nego arhitekturu nadopunjavati prema zadanim uvjetima. Budući da je tijekom analize uočen malen broj aplikacija koje su rađene u skladu s pravcem istraživanja područja mreže podataka, predloženo rješenje je u potpunosti

inovativno, jer se nisu mogli u većoj mjeri koristiti prepoznati obrasci takvih aplikacija.

Osim navedenih smjernica koje su slijeđene pri izradi arhitekture, kako bi se osiguralo da predložena arhitektura nudi potpunost rješenja i da je konkurentna današnjim rješenjima, prema [Murugesan2007] postavljeni su i sljedeći uvjeti koje arhitektura mora zadovoljiti:

a) lako održiva (proširiva, prilagodljiva i jednostavna)

b) sigurna

c) pogodna za korištenje (koherentnost podataka, visoka interaktivnost, itd..)

d) skalabilna

e) robustna i pouzdana

f) prilagođena za zajedničko djelovanje s drugim aplikacijama (interoperabilnost)

Kategorizacijom semantičkih web aplikacija prepoznato je sedam osnovnih kategorija. Uzorak aplikacija podvrgnutih analizi koji pripada kategorijama *semantičke društvene web aplikacije* i *semantičkih ekspertnih sustava* bio je premalen da bi se mogla provesti kvalitetna analiza zahtjeva takvih aplikacija. Dodatno, semantički web ekspertni sustav temelji se na pravilima i zaključivanju, pa zahtijeva ontologiju visoke izražajnosti i kao takav prelazi okvire ovog istraživanja koje se moralo ipak više usmjeriti na primjenu mreže podataka. Kategorija *aplikacija posebne primjene koja semantičke tehnologije koristi u svrhu boljeg pretraživanja* kao najvažniju funkcionalnost ima pretraživanje za koju je već objašnjeno da se ne može poopćiti, pa ni uključiti u arhitekturu. Tako da je kao rezultat istraživanja predložena arhitektura sljedećih kategorija: *semantički web portal*, *semantički sustav za upravljanje znanjem* i *semantička aplikacija preporučenog sadržaja*. *Aplikacija posebne primjene koja semantičke tehnologije koristi u svrhu označavanja sadržaja* je parcijalno rješenje sustava za upravljanje znanjem, pa se arhitektura te kategorije nije posebno navodila budući da se može izvesti iz predloženih.

Tijekom poopćenja funkcionalnosti semantičkih web aplikacija, a onda dodatno i tijekom razrade arhitekture semantičkih web aplikacija pokazalo se da se korištenjem dosad razvijenih pristupa ne može ostvariti integracija podataka kojima se pristupa putem web aplikacija zaštićenog pristupa. U okviru ovog istraživanja, kao prijedlog rješenja na uočen problem, razvijen je sustav za upravljanje identitetom u okruženju semantičkog weba te se ovdje predlaže njegova arhitektura i objašnjava način na koji se koristi u semantičkim web aplikacijama.

6.2. Sustav za upravljanje identitetom

6.2.1. Opis problema

Provedena analiza pokazala je da se u učestale funkcionalnosti semantičkih web aplikacija ubrajaju *integracija više izvora podataka* i *personalizacija*. Personalizacija je usko vezana uz integraciju, tako da se te dvije funkcionalnosti obično implementiraju zajedno na način da se podaci prikupljaju iz različitih izvora podataka i prema karakteristikama korisnika pripremaju za personaliziran prikaz. Pod integracijom više izvora podataka u okviru semantičkog weba misli se na povezivanje podataka koji su dostupni putem weba. Tako dostupne podatke, ovisno o načinu pristupa, može se podijeliti u dva skupa:

- javno dostupni podaci
- podaci kojima se pristupa putem web aplikacija zaštićenog pristupa

Omogućavanje integracije podataka kojima se pristupa putem web aplikacija zaštićenog pristupa otvara pitanje sigurnosti. Odnosno, javlja se problem na koji način omogućiti i udaljenoj aplikaciji autoriziran pristup i dohvat zaštićenih podataka.

Provedena je analiza također pokazala da je personalizacija neizostavan dio društvenih aplikacija, sustava preporučenog sadržaja, a u posljednje vrijeme i semantičkih web portala. Može se reći da je personalizacija danas postala obavezan dio gotovo svake aplikacije. Kako su društvene semantičke web aplikacije isključene iz razrade arhitekture, a funkcionalnost preporučenog sadržaja može biti dio semantičkog web portala, problem pristupa zaštićenim podacima razmotrit će se upravo na primjeru semantičkog web portala.

Portali su zamišljeni kao mjesta na kojima se integriraju podaci neke domene i korisniku prikazuju na personaliziran način. Odnosno, zamišljeni su kao mjesta koja korisniku pružaju podršku pri obavljanju uobičajenih dnevnih zadataka koji pripadaju nekoj domeni. Na primjer, portal koji sadrži zbirno informacije o nadolazećim događanjima u gradu, korisniku skraćuje vrijeme koje inače provede tražeći takve informacije pojedinačno po raznim web stranicama. U današnjem vremenu, koriste se različite aplikacije da bi se zadovoljile osobne potrebe i interesi. Npr. koristi se Flickr[20] za pohranu vlastitih slika, Facebook[21] za održavanje društvene mreže, Google Calendar[22] kao organizator, itd… Svaka od aplikacija ima neku funkcionalnost razvijenu na način koji korisniku najviše odgovara što je uzrok

[20] http://www.flickr.com/
[21] http://www.facebook.com/home.php
[22] www.google.com/calendar

fragmentacije osobnih podataka po različitim aplikacijama. Za potpunost obavljanja spomenutih učestalih dnevnih zadataka došlo je do potrebe integriranja podataka domene portala i osobnih podataka korisnika koji su pohranjeni u različitim aplikacijama. Na primjer, jedan od mogućih scenarija je da korisnik koristeći portal naveden u primjeru pronađe događanje koje mu je zanimljivo, potom ga odmah želi integrirati u vlastiti kalendar koji se nalazi u drugoj aplikaciji i pri tom otkriti moguće kolizije u rasporedu te o tom istom događanju želi obavijestiti prijatelje koristeći kontakte iz treće aplikacije.

Dakle, portal koji odgovara na potrebe današnjeg korisnika prezentira integrirane podatke odabrane domene te ih povezuje s ostalim osobnim podacima korisnika pohranjenim u različitim aplikacijama. Točnije, portal integrira ***vlastite podatke portala, javno dostupne podatke weba***, kao i ***podatke različitih aplikacija zaštićenog pristupa dostupne putem weba***. Proces dohvata podataka zaštićenog pristupa od strane udaljene lokacije uključuje identifikaciju korisnika, autorizaciju udaljene aplikacije i samu akciju dohvata podataka. U analiziranim aplikacijama obično se nije vodilo računa o osjetljivosti podataka koji se integriraju, pa se nije ni vodilo računa o tome da se dohvat takvih podataka odvija na siguran način. Također, ni analiza rezultata istraživanja u području pristupa zaštićenim podacima [Grzonkowski2005, Kruk2004, Pashalidis2003, Samar1999, Suriadi2009] nije pokazala da postoji odgovarajuća infrastruktura i metode koje bi zadovoljile potrebe. No, uvidom u dostupna rješenja i posljednja dostignuća u području *identifikacije* [OpenID2009], *autorizacije* [Atwood2009], *načinu pohrane osobnih podataka korisnika* [Brickley2008] i *mehanizma pristupa podacima* [Clark2009], otkriveno je da se integriranjem rezultata tih područja može pružiti rješenje kakvom se teži. Predložen je *sustav za upravljanje identitetom*, kao i proširenje protokola SPROT, te se njihovom primjenom u semantičkim web aplikacijama ostvaruje integracija zaštićenih podataka weba. Verifikacija predloženog rješenja obavljena je u sklopu izrade semantičkog web portala – Sweb. U sljedećem poglavlju slijedi detaljno objašnjenje predloženog rješenja.

6.2.2. Karakteristike sustava za upravljanje identitetom

Prepoznato je da sustav za upravljanje identitetom za uspješno djelovanje u okruženju semantičkog weba mora imati sljedeće karakteristike:

1. omogućavanje jedinstvene prijave (eng. single sign-on)
   - jedan par identifikator/provjeritelj (npr. korisničko ime i lozinka) za sve servise
   - samo jedna akcija prijavljivanja od strane korisnika
2. korisnički profil zapisan na način koji omogućava njegovo dijeljenje, a ujedno je razumljiv računalima
3. osiguravanje sigurnosti, privatnosti i zaštite podataka

Tijekom realizacije rješenja za pristup i dohvat zaštićenih podataka ustanovilo se da je osnovni tehnički problem ostvariti *sigurnost* takvog pristupa, a pri tom ujedno očuvati *prilagođenost za zajedničko djelovanje s drugim aplikacijama* te *ugodnost korištenja. Ugodnost korištenja* ovdje podrazumijeva transparentan pristup svim izvorima podataka. Dakle, obavljanje identifikacije, prikupljanja i integriranja podataka bez eksplicitnog uključivanja korisnika u taj proces. Rješenje za taj problem prepoznalo se u primjeni mehanizma jedinstvene prijave. Kako bi se pak ostvarila *prilagođenost za zajedničko djelovanje s drugim aplikacijama* predložen je semantički zapis podataka te su razvijeni protokoli za sigurnu razmjenu semantičkih sadržaja.

Tablica 6. Razlozi odluka donošenih pri izradi sustava za upravljanje identitetom

odluka	**razlog – utjecaj na arhitekturu aplikacija semantičkog weba**
izdvajanje sustava za upravljanje identitetom	lakše održavanje aplikacija
jedinstvena prijava	ugodnost korištenja
jedan par identifikator/provjeritelj	mogućnost zajedničkog djelovanja s drugim aplikacijama
jedinstvena prijava jednom akcijom korisnika	ugodnost korištenja
semantički korisnički profil	mogućnost zajedničkog djelovanja s drugim aplikacijama
proširenje SPROT protokola	održavanje skalabilnost mogućnost zajedničkog djelovanja s drugim aplikacijama
mehanizam kontrole pristupa	sigurnost privatnost

Tablica 6 prikazuje sumarno sve odluke koje su donesene pri definiranju i izradi sustava za upravljanje identitetom za okruženje semantičkog weba, te je za svaku odluku naveden razlog zbog čega je donesena, odnosno kakav je njen utjecaj na arhitekturu semantičkih web aplikacija od kojih se očekuje da predložen sustav primjenjuju. Svaka navedena odluka detaljno je objašnjena u tekstu koji slijedi.

6.2.3. Jedinstvena prijava

Za rješenje opisanog problema potrebno je omogućiti semantičkim web aplikacijama pristup zaštićenim podacima koji su pohranjeni u više aplikacija. Kako se ne bi stalno ponavljao postupak prijave za rad u svakoj od aplikacija pojedinačno nužno je omogućiti *jedinstvenu prijavu*. Jedinstvena prijava podrazumijeva svojstvo automatskog pristupa korisnika u više sustava bez potrebe posebne prijave u svaki od njih. Dakle, na taj način se omogućava pristup svim zaštićenim podacima, a pri tom ne narušava *ugodnost korištenja*. Problem *jedinstvene prijave* istražuje se već odavno [Samar1999], a aktualizirao se pojavom, odnosno popularizacijom interaktivnih i društvenih web aplikacija. Kako se povećao broj aplikacija koje korisnik koristi na dnevnoj bazi, sve je veći broj identiteta – korisničkih profila koje bi trebao održavati. Problem održavanja korisničkog profila može se gledati s dvije strane, strane korisnika i strane razvojnog tima. Veliku stavku u održavanju sustava s vremenom su počeli činiti troškovi održavanja identiteta. Učestali problemi su zaboravljene lozinke, pogrešni i nekonzistentni podaci, potreba za ostvarenjem komunikacije s drugim servisima i najvažniji problem - održavanje sigurnosti [Samar1999]. Navedeni razlozi idu u korist ideji da se održavanje identiteta odvoji od samih aplikacija, jer bi očito takvo rješenje imalo utjecaj na smanjenje troškova njihovog održavanja. Jednostavno održavanje je i jedan od uvjeta postavljenih u ovom istraživanju na arhitekturu aplikacija semantičkog weba. Izdvajanje sustava za upravljanje identitetom izvan aplikacija uvjetuje da korisnici imaju samo jedan par identifikator/provjeritelj za pristup u više aplikacija.

6.2.4. Jedan par identifikator/provjeritelj

Kao sustav za identifikaciju i okosnica predloženog pristupa uzeta je tehnologija OpenID [OpenID2009]. OpenID odgovara na postavljenom uvjetu na rješenje, dakle podrazumijeva korištenje samo jednog para identifikatora/provjeritelja. Radi se o otvorenom, decentraliziranom i besplatnom programskom okviru za održavanje

elektronskog identiteta korisnika. Odnosno, decentraliziranom mehanizmu za jedinstvenu prijavu. Dva su razloga zbog kojeg je odabran upravo ovaj mehanizam:

1. kao identifikator koristi URI što znači da se uklapa u okvire semantičkog weba
2. globalno je prihvaćen (postoji niz aplikacija koje ga koriste)

Potrebno je detaljnije objasniti što točno znači upotreba samo jednog para identifikatora i provjeritelja. Dakle, korisnik može odabrati bilo koji servis kao svoj OpenID pružatelj usluge (npr. getopenid.com, myopenid.com, itd…) i s te strane radi se o decentraliziranom mehanizmu. Također, može odabrati i održavati neograničen broj identiteta, ako postoji potreba za tim. Znači, po potrebi korisnik može imati više URI-a, međusobno nezavisnih.

Za ostvarenje jedinstvene prijave, aplikacije moraju podržati prijavu korištenjem OpenID-a. Na taj način odabrani OpenID URI se može tretirati kao globalni jedinstveni identifikator korisnika. Uvođenje globalnog identifikatora se čini kao kontradikcija s temeljnim principima semantičkog weba, budući da se uvodi jedan oblik centralizacije. No, takav način je nužan za ostvarenje integracije privatnih podataka korisnika, jer kada bi korisnik u različitim aplikacijama imao različite URI-e integriranje podataka korisnika zahtijevalo bi primjenu nekakvog oblika izjednačavanja URI-a. Primjena takvih predloženih načina integracije, koja propisuju pravila povezanih podataka, dakle traženje uzoraka ili korištenje metrike sličnosti, uzrokuje sljedeće probleme:

1. ne garantira se točnost
2. korištenje metrike sličnosti zahtijeva da aplikacija ima pristup zaštićenim podacima nad kojima bi se onda zaključivanje moglo i izvoditi, a takav način narušava privatnost korisnika (npr. kad bi se radila usporedba imena, mail adresa i sl.)

Automatska integracija podataka korisnika koja bi garantirala točnost podataka moguća je samo uz uvođenje OpenID-a kao globalnog identifikatora. Pri tom, taj globalni URI korisnika, može se generirati na način da ne bude samogovoreći te se tako dodatno osigurava da se njegovim korištenjem ne otkrivaju nikakve osobne informacije.

Kao provjeritelj predlaže se korištenje lozinke. Iako je ranjivost tog pristupa više nego poznata [Notoatmodjo2007] i dalje je takav pristup najraširenij, a bilo kakva razrada drugih mehanizama prelazi okvire ovog istraživanja.

6.2.5. Korisnički profil – osobni podaci

Odabir tehnologije za pohranu podataka korisničkog profila ima najveći utjecaj na uvjet *podržavanja zajedničkog djelovanja s drugim aplikacijama*. Iz postavki semantičkog weba taj uvjet se može formulirati u sljedeće: korisnički profil *mora biti razumljiv računalima i mora se moći koristiti od strane više servisa*.

Odabir tehnologije za zapis korisničkog profila

Prvi koraci prema standardizaciji prenosivosti podataka već su napravljeni. Tijekom posljednje godine nastala su rješenja za uvoz/izvoz profila korisnika (npr. Facebook Foaf Generator[23]) ili drugih podataka korisnika (npr. Flickr RDF izvoznik[24] ili Google Data API[25]). Aplikacija LODr [Passant2009c] koja je bila uključena u provedenu analizu semantičkih web aplikacija upravo za cilj ima dijeljenje podataka raznih aplikacija. Dodatnu inicijativu podržavanja dijeljenja podataka i uvođenja standardizacije na tom području daje grupa „DataPortability project“[26]. Dakle, kako je rad na tom području već prepoznat, istraživanje se može nastaviti nad rezultatima drugih istraživača.

Dostupno je više radova u kojima se predlaže opis profila korisnika korištenjem FOAF vokabulara [Ankolekar2006, Grzonkowski2005, Kruk2004]. Također, osvrćući se na rezultate analize aplikacija provedene u sklopu ovog istraživanja može se reći da je FOAF vokabular već postao standard za opis profila korisnika. Ukupno je 21 aplikacija koristila postojeće vokabulare, a od toga čak 13 analiziranih web aplikacija koristi FOAF vokabular. Radi se o RDF(S) vokabularu, tako da je uvjet razumljivosti računalima ispunjen. Mogućnost integracije FOAF vokabulara i OpenID tehnologije također je prepoznata, pa je u posljednjem proširenju specifikacije FOAF-a dodana podrška za OpenID. Budući da FOAF ima ograničen skup svojstava, za opis korisničkog profila predlaže se općenitije rješenje, koje se temelji na FOAF-u, ali može koristiti proizvoljan broj RDF vokabulara.

Mogućnost zajedničkog korištenja korisničkog profila

Što se drugog uvjeta tiče, zajedničkog korištenja podataka profila, imperativ je na standardizaciji pristupa podacima. Korisnička aplikacija mora biti u mogućnosti pristupati raznim pružateljima usluga (npr. Google, Verisign, itd..), ovisno kojeg je

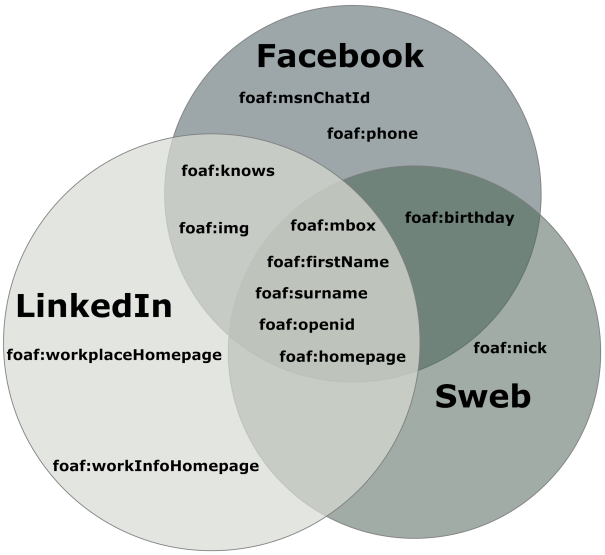
[23] http://www.facebook.com/apps/application.php?id=2626876931
[24] http://librdf.org/flickcurl/
[25] http://code.google.com/intl/hr/apis/gdata/
[26] http://www.dataportability.org/

korisnik odabrao kao svoj OpenID pružatelj usluge. Troškovi razvoja i *održavanja* aplikacije znatno su manji kada je takav pristup standardiziran. Budući da je profil zapisan RDF-om i nalazi se na udaljenoj lokaciji, prirodan odabir za pristup podacima predstavlja korištenje SPROT-a i SPARQL pristupne točke.

6.2.6. Pristup podacima korisničkog profila

Rješavanje pristupa podacima ima najveći utjecaj na uvjet arhitekture - *sigurnost.* Korisnici žele da različiti dijelovi njihovih profila budu vidljivi različitim servisima. Slika 47 prikazuje primjer dozvola različitih aplikacija nad podacima profila. Dakle, na primjer LinkedIn[27] je aplikacija kojoj je primarna namjena ostvariti poslovne kontakte, pa korisnik ne želi da su putem nje vidljivi osobni podaci kao rođendan, privatan broj telefona, itd.

Postojeća rješenja za upravljanje identitetom usmjerena su na identifikaciju i provjeru vjerodostojnosti, dok se u ovom istraživanju naglasak stavlja na autorizaciju pristupa podacima.



Slika 47 Dozvole nad podacima korisničkog profila

OpenID 2.0 specifikacija podržava pohranu i izmjenu proizvoljnih atributa [Fitzpatrick2008]. Ali takvo rješenje ipak ima dva nedostatka:

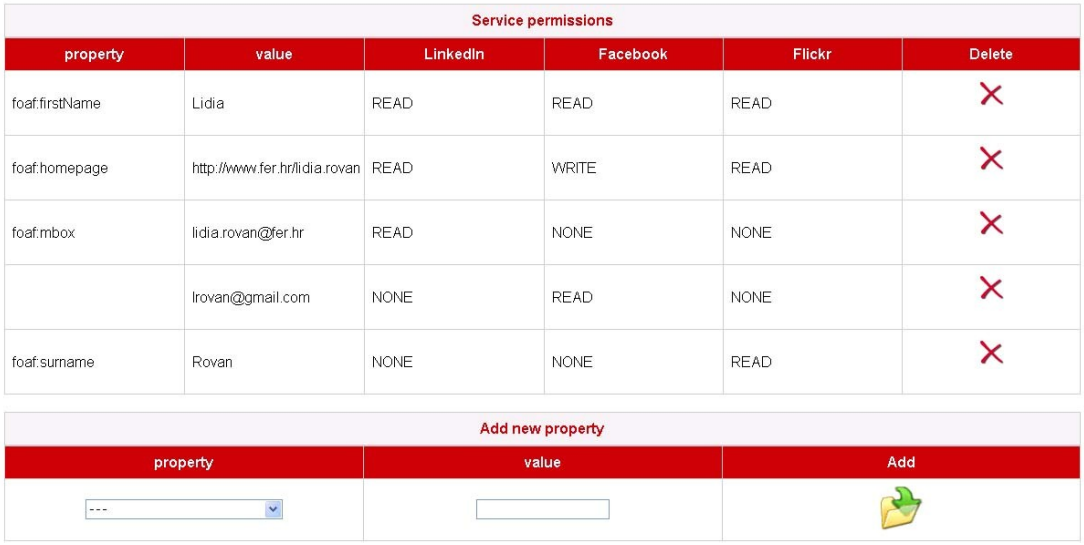
1. podaci su dostupni svim servisima koji imaju dozvolu pristupa
2. atributi nemaju definirano značenje

Korištenjem RDF-a za zapis profila drugi problem se eliminira, odnosno svim se atributima daje značenje. Rješavanje kontrole pristupa podacima profila predstavlja

[27] http://www.linkedin.com/

puno veći izazov. Kao pristupna tehnologija predlaže se korištenje SPROT protokola. Prethodno je naglašen nedostatak alata za kontrolu pristupa podacima u semantičkim repozitorijima. Pristup koji je ovdje opisan temelji se na uvjerenju da će razvoj takvih alata ipak uskoro dati većih rezultata.

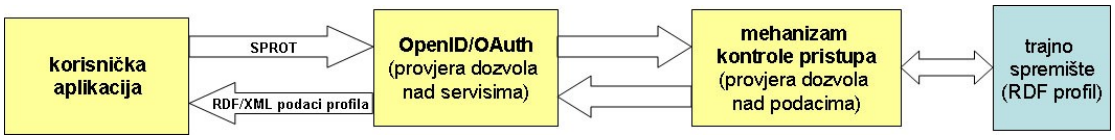
Kontrola pristupa kod predloženog sustava za upravljanje identitetom uključuje mogućnost ograničenja pristupa servisima te dodatno za svaki servis pristup (i vrstu pristupa) samo određenim podacima. Slika 48 prikazuje sučelje kroz koje korisnik dodjeljuje opisane dozvole u sustavu za upravljanje identitetom koji je realiziran u sklopu ovog istraživanja. Najmanja granularnost je dodjeljivanje dozvola nad trojkom (izjavom). Na temelju postavljenih dozvola nad trojkama, semantički repozitorij može obavljati kontrolu pristupa. Kontrola pristupa samog servisa SPARQL pristupnoj točki odvija se tehnologijom OAuth.



Slika 48 Dozvole servisa nad podacima

OAuth je protokol koji omogućava sigurnu autorizaciju servisa te se danas nameće kao standard kontrole pristupa servisima koji pristupaju privatnim podacima korisnika. Od 2009. godine koristi ga i Google za svoje pristupne servise. Predloženo rješenje pristupa korisničkim podacima pohranjenim u sustavu za upravljanje identitetom sastoji se od sljedećih osnovnih komponenti (slika 49):

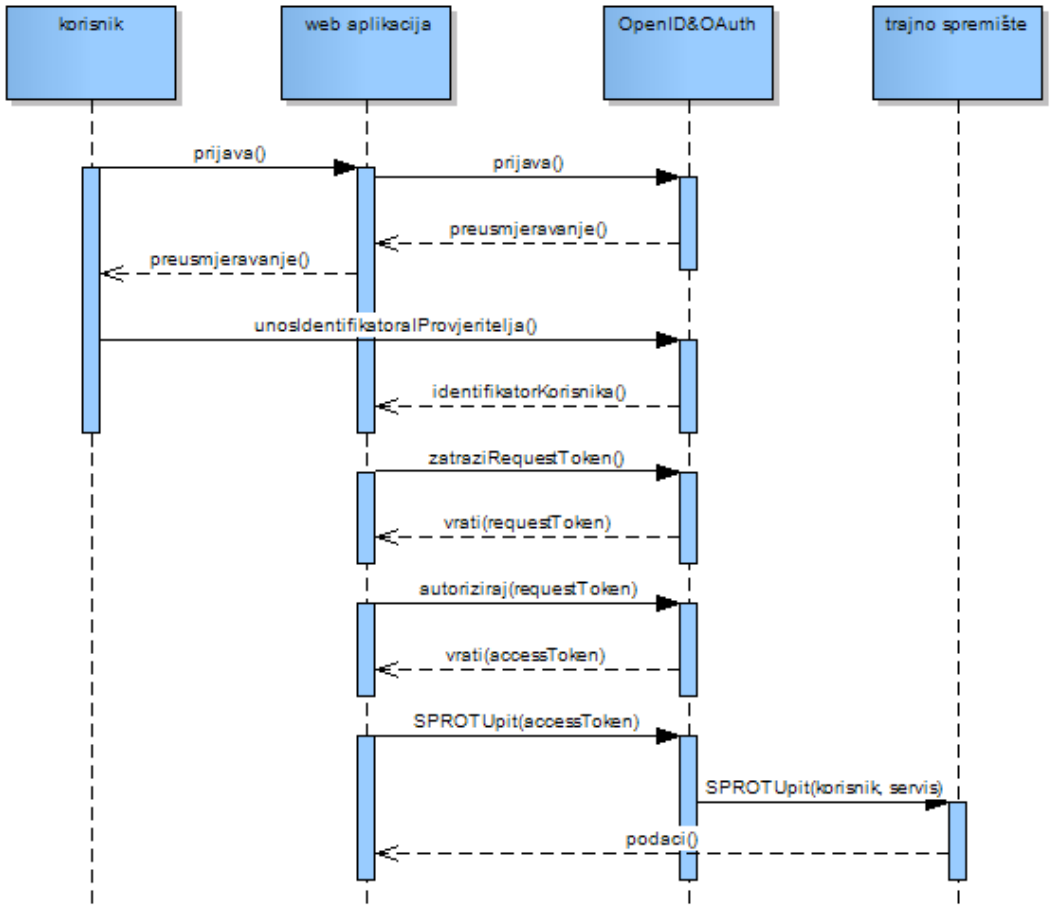
1. hibridni OpenID/OAuth pružatelj usluge
2. mehanizam kontrole pristupa
3. nadogradnja SPROT protokola



Slika 49 Pristup podacima profila korisnika

Hibridni OpenID/OAuth poslužitelj

Budući je za predloženo rješenje potrebna provjera korisnika i autorizacije servisa, realiziran je hibridni poslužitelj OpenID/OAuth. Komunikacija između korisničke aplikacije i OpenID/OAuth poslužitelja u konačnici mora rezultirati pristupom servisu putem kojeg korisnička aplikacija dohvaća podatke. Komunikacija podrazumijeva identifikaciju i autentifikaciju korisnika, kao i autorizaciju pristupa korisničke aplikacije podacima od strane korisnika. U implementaciji predloženog sustava, kako bi se dodatno poboljšala *ugodnost korištenja*, ukoliko korisnik nije prethodno dodijelio dozvole servisu putem sučelja kojeg prikazuje slika 48, može ih dodijeliti prilikom procesa dohvata podataka. Slika 50 prikazuje cjelokupni tijek komunikacije koja se odvija prilikom dohvata podataka.



Slika 50 Tijek podataka pri komunikaciji: korisnička aplikacija - sustav za upravljanje identitetom

Proširenje SPROT protokola

Kod izrade rješenja posebno se pazilo da je sustav lako koristiti tako da se cijeli opisani proces sa strane korisničke aplikacije obavlja samo jednim SPROT upitom. Kako bi se ta jednostavnost implementacije mogla postići, predlaže se nadogradnja SPROT protokola u vidu uključenja dodatnih parametara koji bi sadržavali potrebne podatke za OpenID/OAuth komunikaciju.

SPARQL URL inicijalno se sastoji od tri dijela:

1. URL SPARQL pristupne točke
2. Imena grafova nad kojima se postavlja upit
3. SPARQL upit

I sačinjen od ta tri dijela izgleda ovako:

```
GET/endpoint/profile/ HTTP/1.1
Host:  http://www.sweb.zpr.fer.hr
Content-Type: application/atom+xml
query="SELECT%20DISTINCT%20%3Fname%20%3Fmbox%0D%0AWHERE%20%7B%20%3Fx%20foaf%
3Aname%20%3Fname%3B%20%09%20%0D%0Afoaf%3Ambox%20%3Fmbox%7D%0D%0A"
```

Prema specifikaciji OAuth 1.0 [Atwood2009] potrebni parametri za provođenje OAuth komunikacije dodaju se u Authorization zaglavlje HTTP protokola. Primjer proširenog SPARQL URI-a onda izgleda ovako:

```
GET/endpoint/profile/ HTTP/1.1
Host:  http://www.sweb.zpr.fer.hr
Content-Type: application/atom+xml
query="SELECT%20DISTINCT%20%3Fname%20%3Fmbox%0D%0AWHERE%20%7B%20%3Fx%20foaf%
3Aname%20%3Fname%3B%20%09%20%0D%0Afoaf%3Ambox%20%3Fmbox%7D%0D%0A"
Authorization: OAuth
OAuth_token="1%2Fab3cd9j4ks73hf7g",
OAuth_signature_method="HMAC-SHA1",
OAuth_signature="WyKGTLsxOfTLcDIVwH5hHeqzpwI%3D",
OAuth_consumer_key="sweb",
OAuth_timestamp="1231956529",
OAuth_nonce="57451704142536",
OAuth_version="1.0"
```

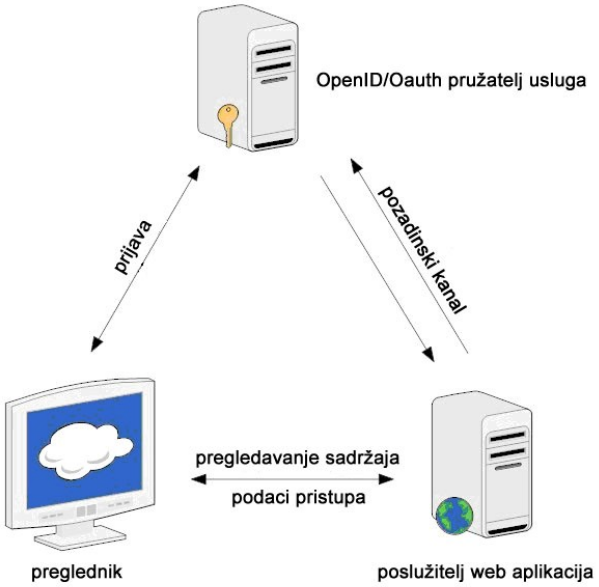
Mehanizam kontrole pristupa

OAuth komunikacija odvija se izmjenom znački (eng. tokena). Kada korisnička aplikacija pribavi *access token*, posljednji u nizu, znači da ima dozvolu pristupa pružatelju usluga. Budući da je svaki *access token* specifičan za jednog korisnika i korisničku aplikaciju, lako je iz njega očitati potrebnu informaciju: *korisnika* i *korisničku aplikaciju (servis)*. Ta dva parametra prosljeđuju se dalje trajnom spremištu podataka te se u skladu s mehanizmom za kontrolu pristupa odvija

odgovarajuće filtriranje upita. Dohvaćeni rezultati upita vraćaju se u skladu sa SPROT specifikacijom.

6.2.7. Jedinstvena prijava jednom akcijom korisnika

Od samog početka istraživanja kao uvjet prihvaćenosti aplikacija naglašava se važnost njihovog jednostavnog i ugodnog korištenja. Upravo zbog zadovoljavanja korisnika nastoji se ostvariti transparentan pristup svim servisima koje aplikacija kontaktira tijekom izvođenja. Problem je ostvariti prijavu u više aplikacija (npr. portal koji koristi podatke Facebooka i podatke Google) samo *jednom akcijom* korisnika i na taj način potpuno ostvariti jedinstvenu prijavu. Iako ovaj dio u okviru ovog istraživanja nije realiziran, analizom postojećih pristupa pokazalo se da je rješenje izvedivo. Predlaže se izgradnja mehanizma jedinstvene prijave jednom akcijom unutar samog preglednika. VeriSign's SeatBelt za preglednik Firefox[28] već podržava višestruku prijavu samo jednom akcijom i ono što je još važnije pruža zaštitu na "napad pecanjem" što se već zamjeralo OpenID-u kao njegov najveći nedostatak [Lee2008]. Dakle, predlaže se nadogradnja takvog pristupa. Slika 51 ilustrira komunikaciju koja se odvija kod automatskih višestrukih prijava. Samo jednom se odvije komunikacija između preglednika i OpenID pružatelja usluge, na korisnikov zahtjev. Ostale prijave u sustav odvijaju se prvo komunikacijom između preglednika i web aplikacije, u svrhu dostavljanja podataka o prijavi, te potom između aplikacije i OpenID pružatelja usluga u svrhu dobivanja svih ostalih dozvola.



Slika 51 Višestruka prijava jednom akcijom korisnika

[28] https://pip.verisignlabs.com/seatbelt.do

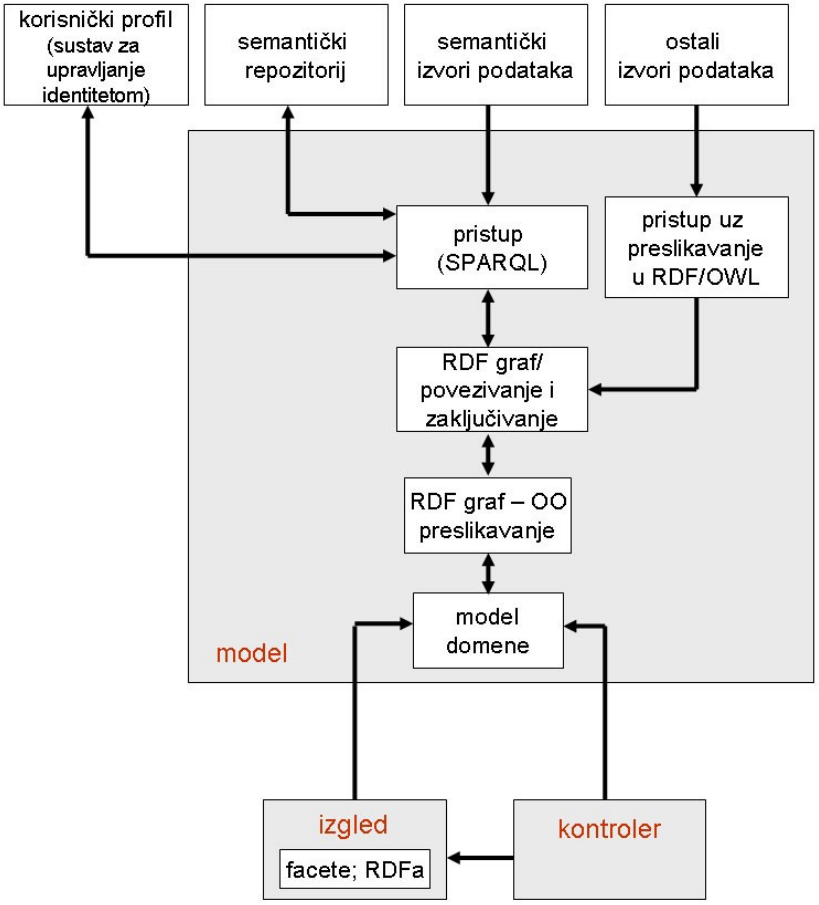
6.3. Semantički web portal

Provedenom analizom semantičkih web aplikacija pokazano je da je semantički web portal najzastupljenija kategorija semantičkih web aplikacija. 39% aplikacija ukupnog uzorka odgovara kategoriji semantičkih web portala.

Prepoznate funkcionalnosti semantičkog web portala su:

- integracija više izvora podataka
- personalizacija
- označavanje
- pretraživanje
- semantički repozitorij

Predloženu arhitekturu semantičkog web portala, koja podržava sve prepoznate funkcionalnosti prikazuje slika 52.



Slika 52 Arhitektura semantičkog web portala

6.3.1. Razrada arhitekture semantičkog web portala

Semantički web portal implementira najveći broj funkcionalnosti karakterističnih za semantičke web aplikacije pa većina objašnjenih elemenata arhitekture u ovom poglavlju vrijedi i za arhitekture preostalih kategorija semantičkih web aplikacija predloženih u ovom istraživanju. Kao potvrda vjerodostojnosti predložene arhitekture, korištenih tehnologija i podržanih funkcionalnosti izgrađen je semantički web portal – Sweb[29], koji će također kasnije biti detaljno opisan. U nastavku opisuje se zasebno predložena arhitektura dijelova model, izgled i kontroler MVC obrasca.

6.3.1.1. Model

Semantički web portal integrira podatke iz više heterogenih izvora podataka. Kako bi se aplikacija mogla uspješno izvoditi nad takvim skupom podataka, potrebno je podatke prikupiti i potom ih integrirati. Budući se radi o semantičkoj web aplikaciji svi izvori moraju se semantički opisati i naposljetku podaci prikupljeni iz tih izvora moraju tvoriti jedan ili više RDF grafova. Arhitektura koja se predlaže vrijedi za Java web aplikacije, što opet dodatno znači da se RDF grafovi moraju konačno preslikati u Java objekte koji se koriste u ostalim dijelovima aplikacije. Dakle, u model dijelu MVC arhitekture semantičkog web portala implementiraju se sljedeće funkcionalnosti:

- prikupljanje podataka
- transformiranje dohvaćenih podataka u RDF grafove
- povezivanje grafova
- dohvat odabranog skupa podataka
- preslikavanje RDF grafova u objektni model

Elementi arhitekture u model dijelu obrasca koje prikazuje slika 52 predstavljaju implementaciju prethodno navedenih funkcionalnosti i predstavljaju logičke slojeve kôda unutar modela MVC obrasca.

Obavljanje opisanog procesa integracije ima najveći utjecaj na sljedeće uvjete postavljene na arhitekturu: *skalabilnost, održavanje, robustnost, pouzdanost i sigurnost*.

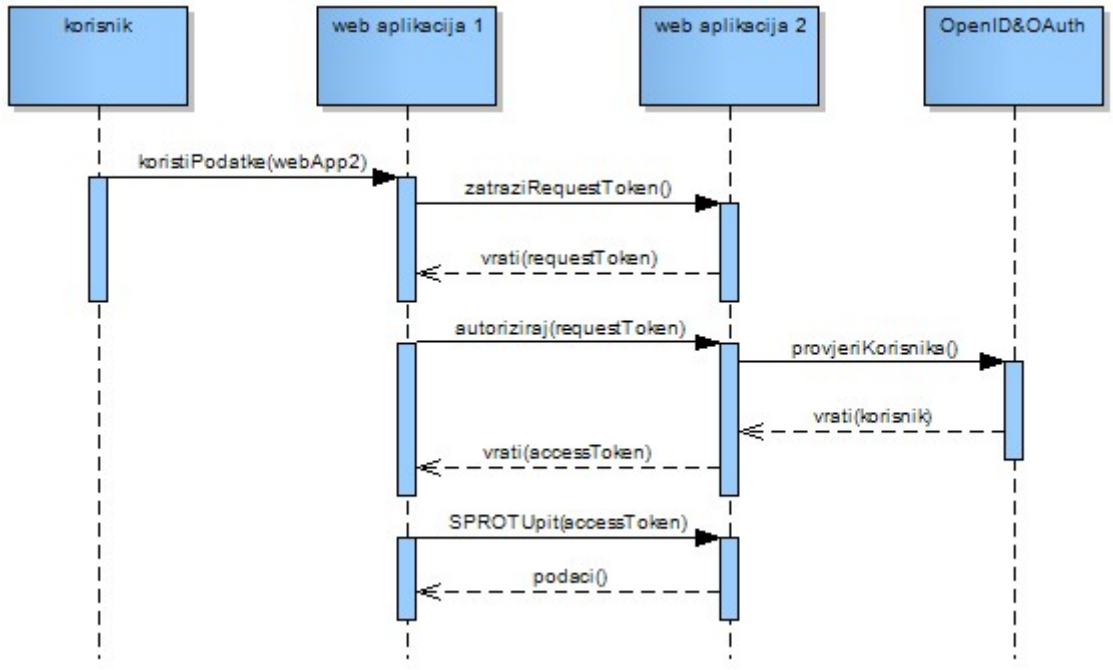
Prikupljanje podataka

Portali svojim sadržajem pokrivaju određenu domenu pa se, prema trenutnim standardima semantičkog weba, skup izvora podataka koje oni pri tom koriste mora

[29] http://sweb.student.hr

eksplicitno definirati. Važno je osigurati da aplikacija uspješno radi i u slučaju nedostupnosti izvora, što je u ideji semantičkog weba scenarij na koji se obavezno mora računati. Osiguravanjem uspješnog rada aplikacije i u slučaju prestanka rada nekog od izvora podataka ostvaruje se uvjet arhitekture – *pouzdanost*.

Kao što je prethodno objašnjeno, izvori podataka koje portal koristi za svoj uspješan rad mogu biti i izvori zaštićenog pristupa. Dohvat javno dostupnih semantičkih podataka jednoznačan je, radi se o upitima putem SPROT-a ili pristupa servisima (npr. GRDDL, RSS/RDF feed, itd.). No, pristup zaštićenim izvorima podataka ipak se razlikuje i zahtijeva dodatno objašnjenje. Za pristup zaštićenim podacima predlaže se korištenje sustava za upravljanje identitetom realiziranog u ovom istraživanju. U tom je slučaju, korisnik je taj koji inicira komunikaciju s aplikacijom koja sadrži zaštićene podatke, a sva se ostala komunikacija odvija između aplikacija bez intervencije korisnika. Slika 53 prikazuje što sve uključuje komunikacija između semantičkog web portala (web aplikacija 1) i aplikacije zaštićenog pristupa (web aplikacija 2) posredstvom predloženog sustava za upravljanje identitetom u svrhu dohvata podataka korisnika.



Slika 53 Komunikacija između aplikacija u svrhu dijeljenja podataka

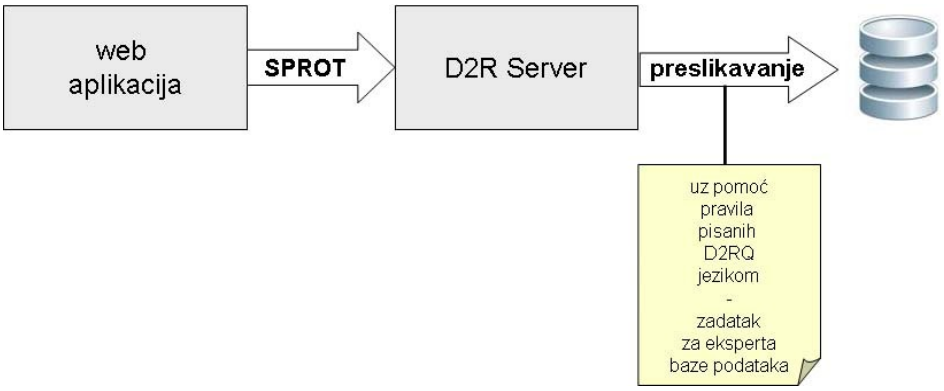
Transformacija podataka

Izvori podataka koje portal integrira ne moraju nužno biti semantički izvori podataka. Za svaki izvor koji ne pruža podatke u RDF-u kao izlaz, implementira se algoritam, odnosno proces preslikavanja u RDF. Uobičajeni ne-semantički izvori podataka su relacijske baze podataka, web resursi XML dijalekta i XML podaci.

1. Transformacija podataka relacijskih baza podataka

Postoje gotova rješenja koja se mogu koristiti za preslikavanje podataka iz relacijskih baza podataka u RDF zapise. Na primjer, D2R server[30] ne samo da služi za preslikavanje podataka u RDF, nego i za njihovu objavu po načelima povezanih podataka. D2R server se temelji na D2RQ alatu, kojim se definiraju pravila preslikavanja modela. Dakle, rješenja na tom području pokazuju zrelost i mogu se ozbiljno razmatrati za korištenje u praksi.

Za primjenu alata koji služe kao potpora preslikavanju potrebno je napredno znanje strukture relacijske baze i značenja podataka, kao i ontologija koje opisuju značenje domene podataka koji se preslikavaju. Tijekom pregleda područja razvoja web aplikacija naglašena je prednost REST aplikacija, pa se u skladu s tim za pristup preslikanim podacima predlaže kreiranje SPARQL pristupne točke preko koje aplikacija može vršiti dohvat podataka. Slika 54 prikazuje komunikaciju koja se odvija prilikom dohvata podataka relacijske baze SPROT protokolom i korištenjem D2R servera kao alata za preslikavanje.



Slika 54 Primjer preslikavanja relacijskih baza podataka u RDF

Semantičko opisivanje podataka relacijske baze podataka zadatak je karakterističan za prilagodbu postojećih web aplikacija semantičkom webu pa je detaljno objašnjenje tog procesa predstavljeno kasnije.

[30] http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/

2. Transformacija web resursa XML dijalekta (npr. XHTML stranice)

Transformaciju se preporučuje realizirati korištenjem standardne semantičke tehnologije GRDDL. Budući je GRDDL W3C standard postoje programski okviri koji pružaju mogućnost rukovanja takvim izvorima. Najveći problem je što svaka promjena izvora podataka (web resursa) zahtijeva i prilagodbu algoritma transformacije, jer je izravno vezan uz sadržaj web stranice.

3. Transformacija XML podataka

Uobičajena su dva pristupa transformiranja i odabir ovisi u velikoj mjeri o karakteristikama izvora kojim se raspolaže (npr. očekivana količina podataka).

1. preslikavanje korištenjem XSLT tehnologije – XSLT (XPath) kôd koji XML podacima daje semantiku ručno se stvara; ukoliko je kompleksna struktura XML podatka javlja se problem *održavanja* algoritma
2. preslikavanje u Java objekte - postoji niz programskih okvira koji se mogu u tu svrhu upotrijebiti; nakon što se XML podaci transformiraju u Java objekte mogu se dalje koristiti programski okviri za njihovu pretvorbu u RDF (npr. Velocity) ili se može direktno iščitavati vrijednosti pohranjene u Java objekte i pohranjivati u RDF trajno spremište; najveći nedostatak ovog pristupa je *skalabilnost,* jer se svi podaci privremeno pohranjuju u Java objekte

4. RSS feed

Programski okviri za razvoj semantičkih aplikacija imaju podršku za rukovanje RSS feed-ovima (na primjer Jena), no pružaju podršku za rukovanje samo osnovnom strukturom, na primjer oznakama *title*, *item* i *description*. Sav sadržaj pohranjen unutar oznaka RSS-a opet treba tretirati kôdom napisanim za tu namjenu.

Povezivanje RDF grafova

Nakon obavljenih potrebnih transformacija svi su podaci dostupni kao RDF grafovi. Sljedeći korak pripreme podataka za njihovo konačno korištenje u aplikaciji je povezivanje takvih grafova. Očekuje se da su podaci dobiveni iz različitih izvora podataka opisani različitim vokabularima koje treba izjednačiti. Izrada "ontologije preslikavanja" nije po svojoj prirodi zadatak koji se povjerava Java programerima. Kako su kod semantičkih web aplikacija podaci centar razvoja aplikacija, uloga eksperata za model podataka u razvojnim timovima je jako važna.

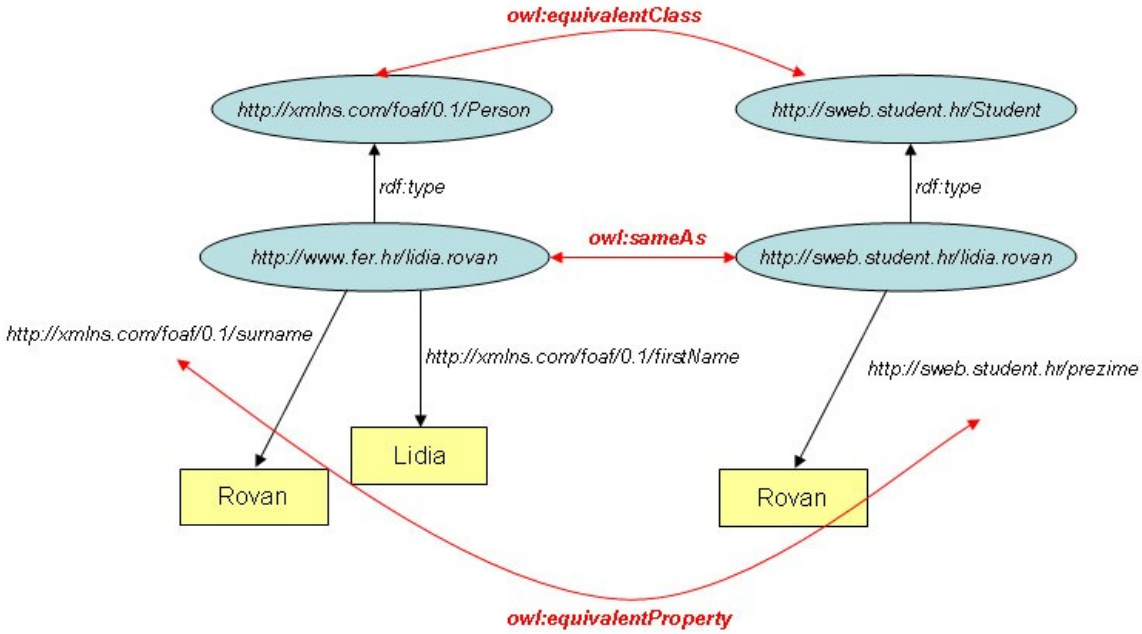
Najjednostavnija realizacija izjednačavanja je na način da se "ontologija preslikavanja" doda u RDF/OWL model aplikacije zajedno sa svim dobivenim grafovima izvora i nad njima izvrši zaključivanje. Druga mogućnost je primjena zaključivanja nad RDF grafovima pohranjenim u memoriji korištenjem programskih okvira ili uključivanjem nekog vanjskog programskog okvira namijenjenog isključivo zaključivanju (npr. Pellet[31]), no na većoj količini podataka takav pristup pokazuje loše rezultate. U poglavlju u kojem su predstavljena RDF trajna spremišta navedeni su načini zaključivanja i performanse koje se očekuju.

Prilikom izrade portala Sweb, pokazalo se da uvođenje veza između samih vokabulara nije osobit problem, već se kao problem pokazalo povezivanje instanci koje mora rezultirati potpunom točnošću. Povezivanje podataka na razini instanci, ovisno o vrsti podataka koji se povezuju, realizira se na sljedeći način:

1. ***Privatni podaci*** – Integracija se obavlja uvođenjem "globalnog identifikatora" osobe. U kombinaciji s predloženim sustavom za upravljanje identitetom pruža jednoznačan proces integracije koji održava privatnost i sigurnost.
2. ***Podaci koji zahtijevaju potpunu točnost –*** Moguće je da postoje dva entiteta sa svim jednakim svojstvima osim samog URI-a koji ih identificira, a da ipak predstavljaju dva *različita* objekta iz svijeta. Na primjer, moguće je da postoje kolegiji s različitim URI-em koji ih identificira, ali istog naziva, broja ECTS-a, semestra u kojem se izvodi, izvođača, itd… Očito, primjenom metoda sličnosti ne može se riješiti navedeni problem. Prilikom izrade Sweba nije se pronašlo nikakvo rješenje koje bi podržavalo autonomiju izvora, automatizaciju integracije, a pri tom garantiralo točnost rezultata integracije. Zbog toga rješenje koje se ovdje predlaže nameće ograničenje na proces objavljivanja podataka od strane različitih izvora podataka. Svi entiteti, koji bi mogli biti dvoznačni koriste samo *jedan identificirajući URI*. Pitanje je kako odrediti koji je URI pravi. U provedenoj se analizi slučaja pokazalo da se može utvrditi koji je izvor podataka vlasnik takvog entiteta, odnosno koji je izvor njegovo iskonsko mjesto nastanka. URI koji je objavio taj izvor podataka uzima se kao "globalni identificirajući URI" i drugi sustavi ga moraju preuzeti. Ovom se rješenju pribjegava u cilju ostvarenja *skalabilnosti* aplikacija, jer nije potrebna nikakva manualna intervencija u svrhu otkrivanja veza između instanci.

[31] http://clarkparsia.com/pellet/

3. ***Podaci koji ne zahtijevaju potpunu točnost*** **–** U slučaju kada se dvosmislenost kao rezultat integracije može tolerirati, tada su primjenjive metode sličnosti. Pod ovu kategoriju spada i sadržaj označen od strane korisnika (npr. korištenjem MOAT-a). Oznake postavljene na taj način ne mogu se, naravno, uzeti sa sigurnošću jer ovise o preciznosti korisnika pa se takvim pristupom potpuna točnost značenja ni ne može garantirati.



Slika 55 Povezivanje RDF grafova

Dohvat određenog skupa podataka i njegovo preslikavanje u Java objekte

Nakon što se uspostavi jedinstven RDF graf koji povezuje sve izvore podataka, podaci se iz njega mogu lako dohvaćati SPARQL upitima. No, tako dohvaćene podatke potrebno je pohraniti u Java objekte da bi se mogli dalje koristiti u aplikaciji. Taj proces transformacije predstavlja most iz objektnog svijeta u svijet trojki i od velike je važnosti pri izradi aplikacije. Naime, ukoliko se dobro napravi dizajn RDF – OO preslikavanja, može se zadržati razdvajanje uloga koje propisuje MVC. Dakle, Java programer koristi samo Java klase i ne mora biti svjestan vrste modela nad kojim aplikacija radi te se rukovanje semantičkim, odnosno RDF podacima, može u potpunosti prepustiti ekspertima na području ontologija. Znači, način implementacije preslikavanja RDF modela u objektni model ima direktan utjecaj na očuvanje separacije uloga, što je jedna od karakteristika MVC obrasca.

Problematika preslikavanja RDF modela u objektni model slična je onoj preslikavanja relacijskih modela u objektne modele. Osnovne razlike između RDF

modela i objektnog modela navedene su u radu Orena i ostalih [Oren2007]. Neke posebnosti RDF-a, na primjer da se svojstva mogu naslijediti iz više klasa, ne mogu se jednostavno implementirati Java jezikom. Stoga se u radu [Oren2007] kod problema preslikavanja RDF modela u objektne ističe prednost dinamičkog jezika (npr. Ruby) kojim je lakše riješiti navedene probleme. Pojednostavljeno, preslikavanje sadržava:

- preslikavanje RDFS klasa u klase objektnog jezika
- preslikavanje RDFS svojstava u članske funkcije klase
- preslikavanje samih resursa u instance klasa

Preslikavanje, naravno, nije tako jednostavno i jednoznačno. Prisutni su problemi nasljeđivanja, određivanja odgovarajućih tipova podataka, vidljivost funkcija i klasa, itd., no detaljno navođenje i objašnjavanje navedenih problema prelazi okvire ovog istraživanja.

Rezultati istraživanja na ovom području rezultirali su dostupnim programskim okvirima. Prvo nastalo rješenje, koje se još uvijek nadograđuje i razvija je RDFReactor [Volkel2006]. Podržava generiranje modela iz RDF sheme ili OWL ontologije te posjeduje API za postavljanje SPARQL upita i vraćanje rezultata u definiranim objektima modela podataka. Najveći nedostatak je nemogućnost preslikavanja istog objekta u više pojmova vokabulara, što je postao učestali scenarij, pogotovo u sklopu stvaranja mreže podataka. Ostali dostupni programski okviri su: Som(m)er[32], Elmo[33] i JSR11[34]. No, navedeni okviri su većinom još u razvoju pa se za njihovu ozbiljniju primjenu u aplikacijama ipak treba još pričekati.

Osnovna zamjerka svim dostupnim programskim okvirima je što ne pružaju podršku označavanju sadržaja u prezentacijskom sloju (web stranicama). Dakle, samo preslikavanje RDF modela u objektni model je izvedivo. Korištenjem tako dobivenog objektnog modela prikazuju se podaci u prezentacijskom sloju, koji pripada izgled dijelu MVC obrasca. Podatke je potrebno označiti kako bi preglednici semantičkog weba takav sadržaj mogli razumjeti i koristiti. Model, koji se koristi za prikaz dinamičkog sadržaja aplikacije, tj. objektni model, mora pružati mogućnost dohvata podatka zajedno s njegovim meta opisom. Ta veza „značenje – vrijednost podatka“ mora ostati sačuvana sve do prikaza podatka na ekranu. Mogući način prilagodbe modela podataka označavanju objašnjeni su kasnije kod opisa procesa prilagodbe uvođenja značenja prikazanog sadržaja.

[32] https://sommer.dev.java.net/sommer
[33] http://www.openrdf.org/
[34] http://www.jcp.org/en/jsr/detail?id=311

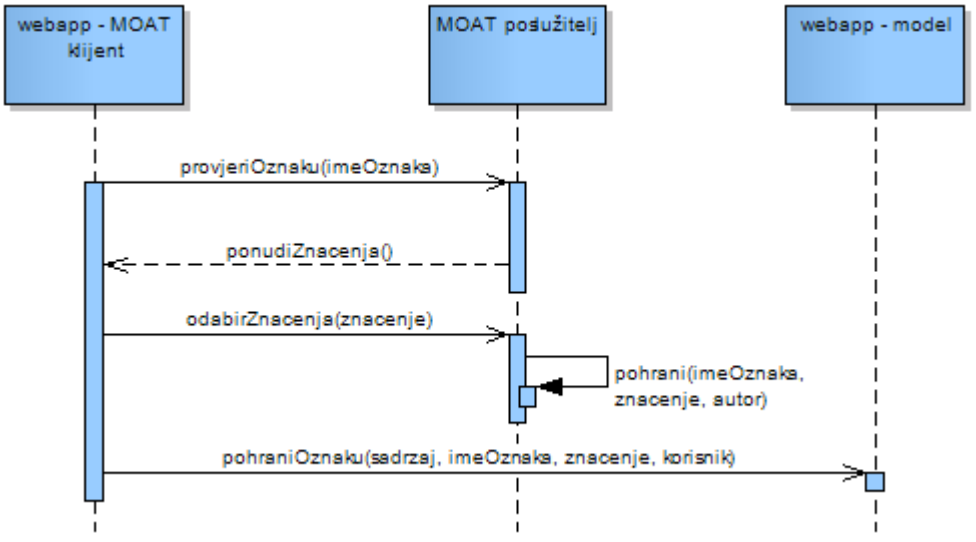
6.3.1.2. Izgled

Označavanje sadržaja kojeg generira aplikacija

RDFa je preporučena tehnologija za obogaćivanje prezentacije semantikom. Trenutno nema programskih okvira za Java jezik koji bi pomogli u procesu označavanja web prikaza, pa se implementacija obavlja na način da ih se ručno dodaje prilikom izrade web stranica. Postoji rješenje za jezik Ruby koje ipak sadrži priličan broj nedostataka koji su navedeni u radu [Rovan2008b]. Prethodno je istaknuto da se pravilnim dizajnom modela može sačuvati veza sa značenjem podataka te bi svakako trebalo voditi računa o tom pri implementaciji rješenja te bi se na taj način omogućilo donekle automatsko označavanje sadržaja. Sve prednosti korištenja takvog pristupa navedene su u radu [Rovan2008b].

Označavanje sadržaja od strane korisnika

Kako bi se omogućilo i samom korisniku kreiranje oznaka (npr. forum, blog, komentari, itd…), što je danas mogućnost gotovo svakog portala, predlaže se korištenje tehnologije MOAT. Naime, korištenjem MOAT infrastrukture implementacija je jednostavna i ne narušava svojstva MVC arhitekture. Budući da se zahtijeva interakcija s korisnikom, koja naravno mora biti dinamična, prikladno ju je realizirati primjenom razvojnih paradigmi bogatih Internet aplikacija (npr. Ajax). Komunikaciju koja se tako odvija između web aplikacije (MOAT klijenta) i MOAT poslužitelja prikazuje slika 56.



Slika 56 Osnovna komunikacija između MOAT klijenta (web aplikacije) i poslužitelja

Semantičko pretraživanje

Tijekom analize aplikacija kao važna funkcionalnost izdvojilo se pregledavanje i pretraživanje po facetama. Na temelju ontologije koja pokriva domenu aplikacije, odaberu se kategorije koje će predstavljati facete. Korisniku se tada nudi vizualno pretraživanje korištenjem faceta. Njihovim odabirom na sučelju te raznim kombiniranjem operatora među njima (presjek, unija, itd..) obavljaju se upiti nad RDF grafovima. Sama izrada algoritama (koje se implementira u model dijelu obrasca) ipak ovisi o domeni aplikacije i željenoj točnosti pretraživanja. Kao tehničko rješenje korisničke strane pretraživanja, predlaže se primjena paradigmi iz bogatih internet aplikacija (npr. tehnike "uzmi i spusti") kako bi se očuvala ugodnost korištenja.

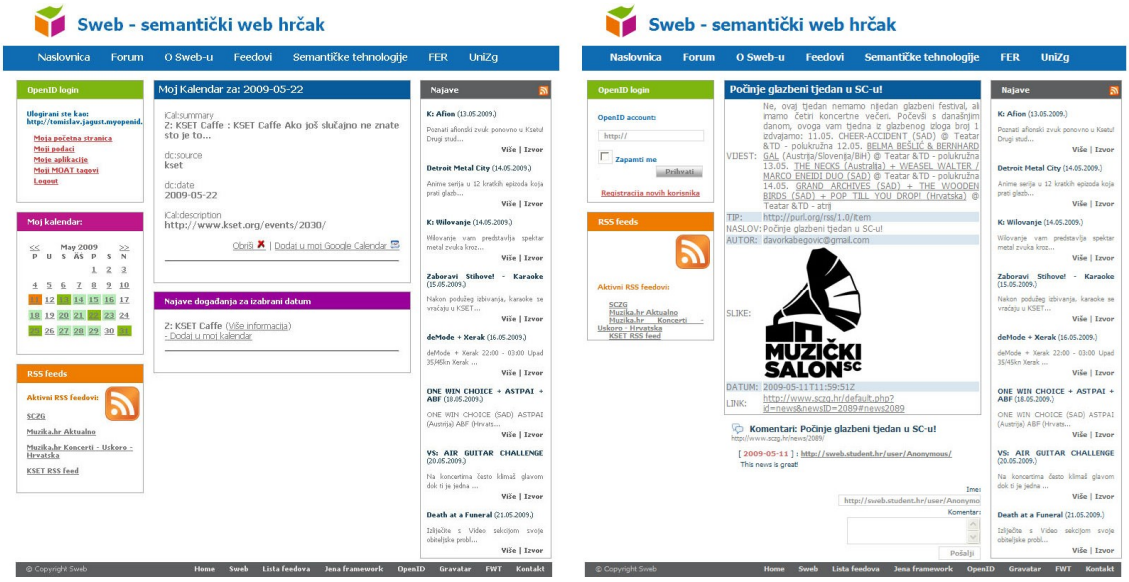
6.3.1.3. Kontroler

Ovaj dio obrasca neovisan je o semantici. Radi se o kôdu koji je zadužen za kontrolu HTTP komunikacije između klijenta i poslužitelja, dakle potpuno je neovisan o semantičkim tehnologijama.

6.3.2. Sweb – verifikacija predložene arhitekture

Kroz izradu semantičkog web portala - Sweb[35] verificirana je predložena arhitektura i provjereni sugerirani implementacijski detalji. Sweb je semantički web portal namijenjen studentskoj populaciji, zamišljen kao mjesto na kojem se integriraju podaci dohvaćeni iz različitih servisa, koje student koristi na dnevnoj bazi [Rovan2008a]. Na primjer, datum ispita, raspored predavanja, informacije o događanjima u gradu, itd.. Portal prikuplja podatke, integrira ih, kombinira i prezentira specifično za svakog korisnika. Usluge koje portal pruža mogu se podijeliti u dvije kategorije: *informacije o događanjima* i *osobne usluge studenta*.

[35] http://sweb.student.hr



Slika 57 Izgled semantičkog web portala - Sweb

Informacije o događanjima prikupljaju se iz RSS/RDF feed-ova sljedećih web lokacija:

- Studentski centar Sveučilišta u Zagrebu[36] – vijesti
- Hrvatski glazbeni portal[37] – vijesti i najave događanja
- KSET: klub studenata Fakulteta elektrotehnike i računarstva[38] – najave događanja

Osim navedenih osnovnih informacija o događanjima, portal nudi mogućnost prikaza potpuno personaliziranog rasporeda studentskih obaveza. Informacije o obavezama studenata prikupljaju se iz nekoliko informacijskih sustava fakulteta. Kako je s upotrebom raznih aplikacija za e-učenje i drugih aplikacija koje se koriste u akademske svrhe nastupila fragmentacija informacija o studiju studenta, student do svih potrebnih informacija o samo jednom kolegiju dolazi korištenjem tri ili četiri različite aplikacije. Na primjer, jedna sadržava raspored predavanja, druga termine ispita, treća rezultate ispita, itd… Tako da je jedna od osnovnih usluga portala integriran pregled nad svim aktivnostima studenta. Student takav pregled može sinkronizirati sa svojim osobnim kalendarom, koji se može nalaziti i u nekoj drugoj aplikaciji, na primjer Google kalendaru. Dodatno, kao aspekt socijalizacije korisnici mogu pisati komentare na objavljene vijesti, mogu označavati prikazan sadržaj ili mogu sudjelovati u raspravama na forumu.

[36] www.sczg.hr
[37] www.muzika.hr
[38] www.kset.org

Tablica 7. Pregled ustupaka u arhitekturi kod razvoja Sweba

	održavanje	sigurnost	jednostavnost korištenja	skalabilnost	robusnost i pouzdanost	interoperabilnost
jedinstvena prijava	nije dio W3C standarda; mogući problemi s kompatibilnošću	bez ustupaka	bez ustupaka	bez ustupaka	jedna točka pogreške	bez ustupaka
pohrana podataka	bez ustupaka	nedostatak mehanizma kontrole pristupa	potreba za boljim administracijskim alatima	nije testirano na velikom broju podataka	nije testirano	bez ustupaka
RSS/RDF, GRDDL prikupljanje podataka	različiti formati izvora podataka zahtijevaju vlastiti transformator	odnosi se samo na javno dostupne izvore	informacija može kasniti zbog potrebne pohrane u međuspremnik	novi format izvora zahtijeva specifikaciju i potencijalno novi transformator	osigurano rukovanje nedostupnim izvorom	bez ustupaka
prikupljanje podataka iz SPARQL krajnjih točaka	bez ustupaka	nije dio W3C standarda	bez ustupaka	bez ustupaka	mogući problemi na velikom broju podataka kod korištenja metoda preslikavanja (D2R)	bez ustupaka
integracija podataka	potreba za pomirbom podataka na višoj razini (djelomična centralizacija)	nije dio W3C standarda	nije potpuno automatizirano	potreba za ručnim dodavanjem novih veza ontologija preslikavanja	bez ustupaka	bez ustupaka
označavanje podataka	bez ustupaka	bez ustupaka	bez ustupaka	bez ustupaka	bez ustupaka	bez ustupaka
korisničko sučelje	bez ustupaka	bez ustupaka	bez ustupaka	bez ustupaka	bez ustupaka	bez ustupaka

Prilikom izrade Sweba verificiran je prijedlog arhitekture semantičkog web portala. Budući da Sweb integrira uz javno dostupne podatke i zaštićene podatke korisnika, također je kroz njegovu izradu verificiran i sustav za upravljanje identitetom, kao i predloženi protokoli razmjene sadržaja. Implementacija Sweba pokazala je u kojoj je mjeri trenutno moguće udovoljiti postavljenim tehničkim uvjetima nad arhitekturom aplikacija. Pregled učinjenih ustupaka u arhitekturi prikazuje tablica 7. Iz tablice se može iščitati da je uvjet sigurnosti u najmanjoj mjeri zadovoljen. Osnovni problem je nedostatak kontrole pristupa trajnim spremištima podataka, kao i pristupa SPARQL krajnjim točkama. Upotreba predloženih mehanizama razmjene podataka dijelom rješava taj problem, no i taj pristup bi imao značajnu korist od mehanizama kontrole pristupa. Na održavanje i skalabilnost utječe broj ne-semantičkih izvora podataka. Za svaki takav izvor mora se pisati poseban transformator kojim se podaci preoblikuju u RDF. Na tom području je teško uvesti standardizaciju, tako da se ne očekuje znatniji pomak. Integracija podataka ima utjecaj na održavanje i skalabilnost. Budući da je nemoguće koristiti automatski oblik integracije, jer se zahtijeva potpuna točnost podataka, svaki novi izvor

podataka traži ručno nadopunjavanje ontologije preslikavanja, a u nekim slučajevima i donošenje dogovora o izjednačavanju identifikatora različitih izvora podataka.

Kako bi se ostvarilo označavanje prikazanog sadržaja dodatno je implementiran i MOAT poslužitelj[39], koji se koristi iz Sweba. Komunikacija između Sweba i MOAT poslužitelja ostvarena je upravo kako pokazuje slika 56 te se tijekom implementacije nisu otkrili mogući problemi.

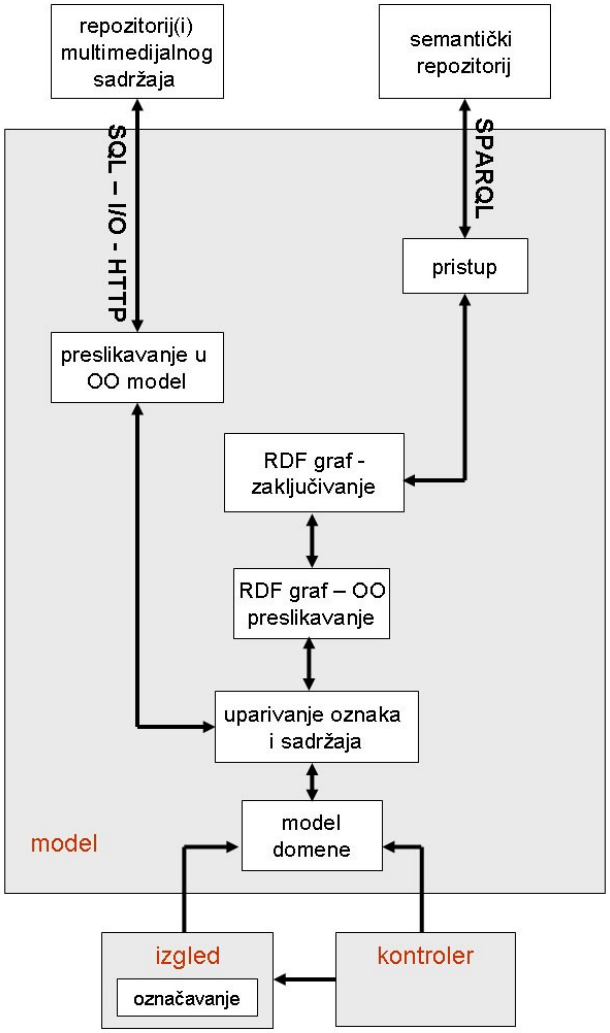
Realizirani sustav za upravljanje identitetom posjeduje potencijalan problem jedinstvene točke pogreške. U slučaju prestanka rada sustava korisniku će biti onemogućen pristup u više aplikacija.

6.4. Semantički sustav za upravljanje znanjem

Druga najzastupljenija kategorija semantičkih web aplikacija je semantički sustav za upravljanje znanjem. Od analiziranih web aplikacija koje su kategorizirane kao sustav za upravljanje znanjem, dakle njih ukupno dvanaest, četiri aplikacije predstavljaju nadogradnju Wiki aplikacija. DBpedia [Auer2007] - vjerojatno trenutno najpopularnija semantička web aplikacija, semantička je verzija Wikipedije. Wiki sustavi se iz tog razloga izuzimaju iz detaljnijeg razmatranja, jer je DBpedia posljednje dostignuće na tom području i već priznato rješenje.

Arhitektura sustava za upravljanje znanjem, koja se predlaže u ovom istraživanju, odnosi se na sustave za upravljanje znanjem koji nisu namijenjeni širokoj populaciji, nego im je osnovna namjena korištenje unutar organizacije. Slika 58 prikazuje predloženu arhitekturu sustava za upravljanje znanjem.

[39] http://moat.student.hr/

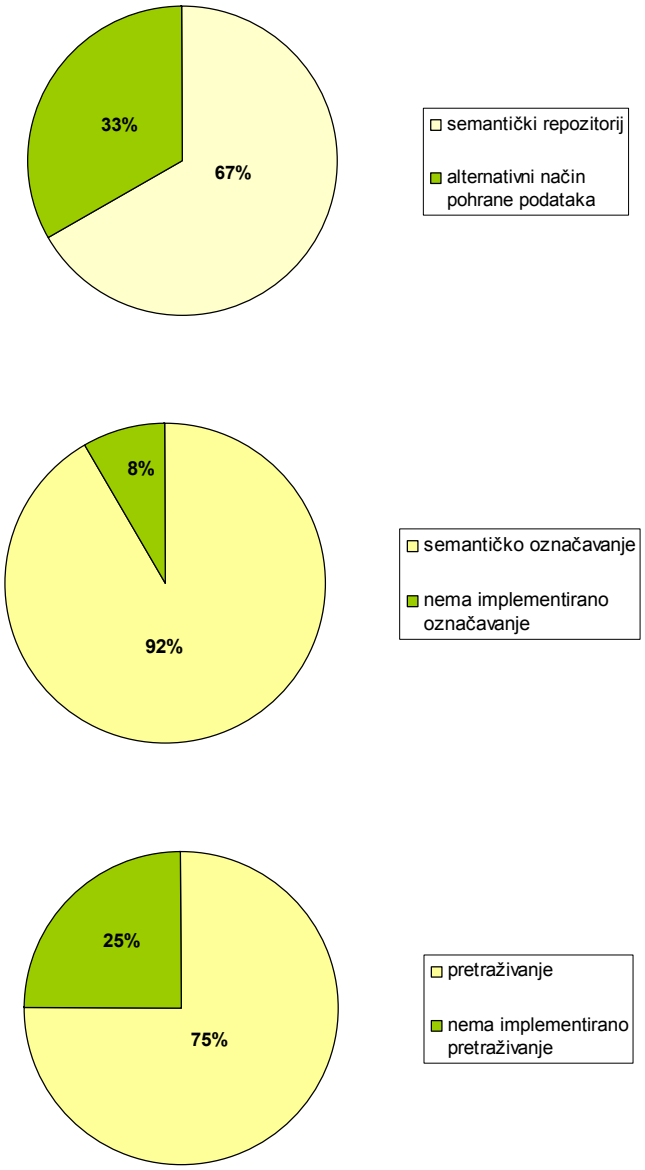


Slika 58 Arhitektura semantičkog sustava za upravljanje znanjem

Najvažnije funkcionalnosti semantičkog sustava za upravljanje znanjem su implementacija *semantičkog repozitorija*, *označavanje sadržaja* i *pretraživanje*. Slika 59 prikazuje postotak implementacija spomenutih funkcionalnosti u sustavima za upravljanje znanjem. Postizanje veće točnosti znanja (dosljednosti i potpunosti znanja) primjenom zaključivanja dio je definicije aplikacija ove kategorije. Kako bi se taj zadatak obavio što kvalitetnije, potrebno je implementirati dodatne mehanizme uz osnovno semantičko zaključivanje. No, kako je samo u aplikacijama PressIndex [Amardeilh2006] i Doris [Bhagdev2007] korišten dodatan mehanizam za realizaciju zaključivanja, a uz to se radi o primjeni algoritama iz područja „prepoznavanja informacija“ detaljnija razrada te funkcionalnosti izostavlja se iz prijedloga arhitekture. Također, moguće je da sustav za upravljanje znanjem integrira više

izvora podataka pa se za implementaciju te funkcionalnosti mogu koristiti elementi arhitekture semantičkog web portala.

Za razliku od semantičkih web portala kod kojih je bilo lako obaviti poopćenje njegovih funkcionalnosti, pa tako i cjelokupnog dizajna, sustavi za upravljanje znanjem su usmjereni na stvaranje koristi određenim zajednicama, pa tako posjeduju brojne specifičnosti. Njihov konačan izgled, algoritmi koji se u njima primjenjuju, pa i sam izgled sučelja u velikoj mjeri ovisi o vrsti organizacije, kao i vrsti posla za koji su namijenjeni.



Slika 59 Udjeli funkcionalnosti u sustavima za upravljanje znanjem

6.4.1. Razrada arhitekture sustava za upravljanje znanjem

Osnovna namjena aplikacija ove kategorije je kreiranje i pohranjivanje sadržaja koju može obavljati korisnik i/ili se može vršiti automatski. Za razliku od web portala naglasak nije na integraciji i skupnom pregledu podataka iz više izvora podataka, iako integracija također može biti jedna od funkcionalnosti. Naglasak je na što boljoj iskoristivosti semantičkih repozitorija u vidu pohrane sadržaja, te na što uspješnijim metodama pronalaženja pohranjenog sadržaja. Osigurati što kvalitetniju pretragu pohranjenog sadržaja važno je primarno zbog uspješnog dijeljenja znanja, odnosno sadržaja između različitih korisnika. Jedna od metoda poboljšavanja pretraživanja uočenih tijekom analize semantičkih web aplikacije je označavanje sadržaja. Označavanje može vršiti korisnik ili se taj zadatak može obavljati programski, dakle automatski.

Velika je razlika u implementaciji navedenih načina označavanja sadržaja, tako da se kod ručnog označavanja sadržaja većina implementacije bazira na komunikaciji između *model* i *izgled* dijelova MVC obrasca, dok je kod automatskog označavanja sadržaja cijela implementacija u *model* dijelu obrasca MVC. U nastavku, slijedi razrada arhitekture za svaki dio MVC obrasca.

6.4.1.1. Model

Trajna spremišta

U samim počecima semantičkog weba realizacija semantičkog web repozitorija bila je velik problem. Danas ipak postoje gotova rješenja koja uspijevaju zadovoljiti potrebe korisnika. U sustavima za upravljanje znanjem naglasak je na pohranjivanju multimedijalnog sadržaja, dakle video zapisa, fotografija, tekstualnih dokumenata, prezentacija, itd... Dakle, svih vrsta dokumenata koji se danas koriste u radu neke organizacije. RDF trajna spremišta još uvijek nisu dorasla izazovu pohranjivanja takvog sadržaja. Tehnički gledano, multimedijalan sadržaj bi se mogao pohraniti u postojećim RDF trajnim spremištima. Naime, moguće je pohraniti resurs binarnog sadržaja te se u tu svrhu koriste tipovi podataka xsd:base64binary i xsd:hexbinary. No, semantička spremišta nisu prilagođena za jednostavno rukovanje takvim zapisima niti za njihovo jednostavno arhiviranje. Zbog navedenog razloga za pohranu multimedijalnog sadržaja teži se rješenju kakvo je primjerice realizirano u projektu CONFOTO [Nowack2006]. Predlaže se odvajanje spremišta multimedijalnog sadržaja od RDF spremišta. Kako su podaci na taj način fizički razdvojeni potrebno je unijeti logiku u aplikaciju koja će vršiti povezivanje

multimedijalnih podataka s njihovim semantičkim opisom pohranjenim u semantičkom repozitoriju.

Kako bi sustav za upravljanje znanjem uspješno djelovao u okruženju mreže podataka, sve je pohranjene podatke potrebno objaviti u skladu s načelima koncepta povezanih podataka.

Realizacija modela i izgleda kod ručnog označavanja sadržaja

Kako je u većini slučajeva korisnik taj koji kreira i označava sadržaj, poželjno je osigurati dinamičnu interakciju kakva se koristi u bogatim internet aplikacijama da bi se postiglo što veće zadovoljstvo korisnika pri obavljanju navedenih poslova. Naime, moduli koji služe kao pomoć pri označavanju trebaju u kratkom vremenu nuditi prijedloge na temelju semantičke usporedbe sadržaja koji se označava i trenutnog sadržaja baze znanja (semantički repozitorij). Kod dizajna aplikacije mora se staviti naglasak na oblikovanje sučelja takvih modula, odabira prikladnih komponenti (izbornici, liste, polja za upis teksta) i sl., a sve u svrhu omogućavanja unosa kvalitetnijih opisnih podataka od strane samog autora. Budući da su sustavi za upravljanje znanjem prvenstveno namijenjeni nekoj organizaciji, uobičajeno je da se u njima koristi terminologija specifične domene i da će isti sadržaj koristiti osobe koje pripadaju srodnim jezičnim zajednicama. Kako heterogenost korisnika ovih sustava nije jako izražena, oznake koje postavi jedan od korisnika vjerojatno su razumljive i svim njegovim bliskim suradnicima. Samim tim, uvođenje semantike na globalnoj razini nije od presudnog značaja pa je dovoljno za oznake sadržaja koristiti samo pojmove pohranjene u bazi znanja aplikacije.

Tehnička rješenja ručnog označavanja sadržaja koja su prethodno analizirana (npr. MOAT), u ovakvim aplikacijama ipak imaju nedostatak. Da bi se povećala točnost pri pronalaženju informacija, potrebno je koristiti kompleksnije metode označavanja. Primjer je implementacija označavanja u aplikaciji EachWiki [Zhang2009] - unose se oznake, veze, kontekstno ovisni atributi, itd… Dakle, nije dovoljno samo unošenje oznaka sadržaja, koje zapravo predstavljaju ključne riječi s definiranim značenjem, potrebno je uvesti i naprednije metode označavanja. Takve je postupke teško generalizirati pa se detaljnije razmatra samo uvođenje oznaka. Također, poželjno je osigurati da mehanizam za označavanje sugerira oznake korisniku pa se predlaže nadogradnja klasične MOAT arhitekture u vidu implementiranja algoritma koji na osnovu sadržaja koji se označava i sadržaja semantičkog repozitorija (baze znanja) korisniku nudi skup oznaka.

Realizacija modela automatskog označavanja sadržaja

Realizacija automatskog označavanja sadržaja odvija se prikladnom metodom iz područja pretraživanje informacija (npr. kategorizacija dokumenata). Ovakav se pristup obično koristi u slučajevima kada nije korisnik taj koji stvara sadržaj, već se on automatski pronalazi i po mogućnosti preoblikuje u semantičke podatke (bazu znanja) ili samo označava. Pressindex [Amardeilh2006] je tipičan primjer automatskog pronalaženja informacija te pretvaranja prikupljenog sadržaja u semantički.

Pretraživanje

Iako su moguće razne implementacije pretraživanja, ovdje predložen pristup zasniva se na pretraživanju korištenjem oznaka. Korištenjem ontologije za opis oznaka (npr. MOAT), pretraživanje označenih sadržaja može se obavljati SPARQL upitima, primjenom mehanizama zaključivanja, metrika sličnosti, itd... Oznake imaju značenje pohranjeno u bazi znanja pa je moguće zaključivanjem nad bazom znanja i metrikama sličnostima pronaći semantički bliske sadržaje što je jako važno za realizaciju pretraživanja.

Uparivanje sadržaja i oznaka

Ovaj je dio posebno izdvojen u predloženoj arhitekturi kako bi se skrenula pozornost na sljedeće:

- svaki multimedijalan sadržaj mora imati pripadajući URI
- oznake su pohranjene u semantičkom repozitoriju s referencom na URI sadržaja (npr. po specifikaciji MOAT-a)

Oznake i multimedijalni sadržaj dohvaćaju se iz dva različita izvora te je za pripremu krajnjeg objektnog modela, popunjenog svim podacima potrebnim za ispravan rad aplikacije, potrebno u aplikaciji osigurati „sloj“ koji bi osigurao pravilno povezivanje podataka.

Problem koji se ovdje dodatno javlja je taj što se prema načelima povezanih podataka svaki URI mora moći razriješiti. No, kako se ovdje radi o sustavu koji se koristi unutar organizacije prilagodba mreži podataka nije od posebno velike važnosti.

6.4.1.2. Izgled

Kod izrade sučelja aplikacija ove kategorije, posebna se pažnja mora obratiti na web stranice putem kojih korisnik obavlja označavanje sadržaja. Na tim je stranicama jako važno osigurati dinamiku, dakle primijeniti metode bogatih Internet aplikacija, kako bi korisnici taj posao obavljali na najlakši mogući način.

Uvođenje RDFa oznaka u izgled primarno se uvodi zbog pretraživača i agenata semantičkog weba. No, kako su aplikacije ove kategorije namijenjene uporabi unutar organizacije uvođenjem RDFa opisa u izgled ostvaruju se i druge prednosti. Korištenje RDFa oznaka za označavanje prikazanih podataka po uzoru na RLM aplikaciju [Clarke2009], naročito kod aplikacija ove kategorije može pružiti značajnu korist glede tehničke realizacije aplikacije. Naime, u RLM aplikaciji naglasak je također na označavanju sadržaja te je moguće direktno rukovanje RDF modelima u sučelju, koji opisuju sadržaj. Ti su RDF modeli naravno, opisani korištenjem RDFa oznaka. Nakon svih promjena nad resursima koje korisnik obavlja u sučelju tehnikama „uzmi i spusti“, samo se jednom obavi komunikacija s poslužiteljem u svrhu spremanja promjena nad modelom. S tehničke strane ovo rješenje pruža jednostavnost implementacije, a opet zbog elegancije njegovog korištenja donosi i zadovoljstvo korisnika.

6.4.1.3. Udovoljavanje tehničkim uvjetima postavljenim nad arhitekturu

Pregled svih odluka donesenih prilikom definiranja arhitekture u svrhu zadovoljavanja tehničkih uvjeta postavljenih nad arhitekturom prikazuje tablica 8.

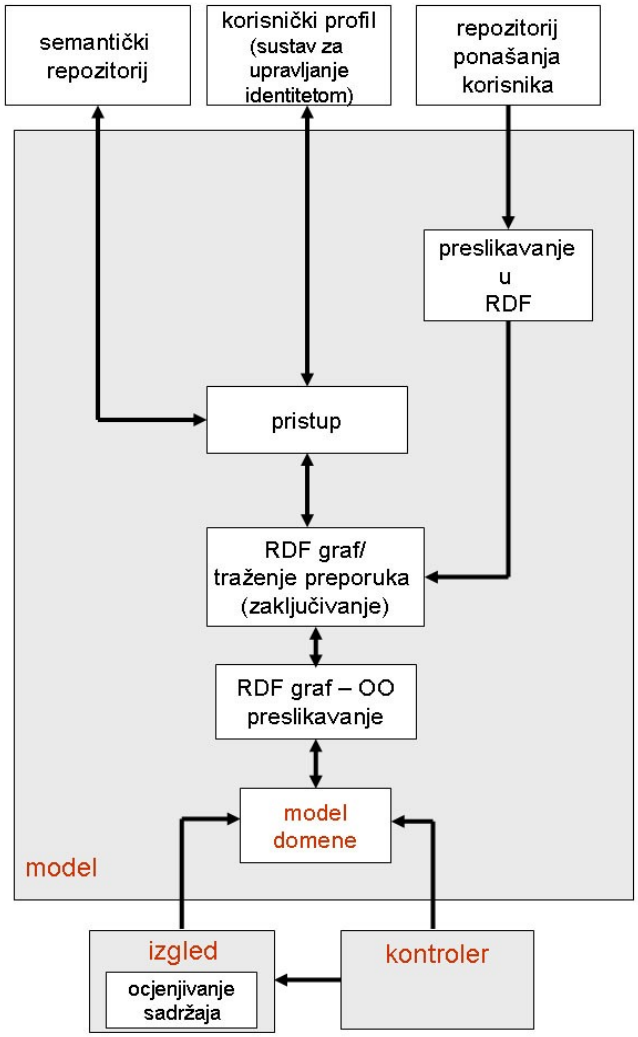
Tablica 8. Udovoljavanje tehničkim uvjetima postavljenim nad arhitekturu

	održavanje	**sigurnost**	**jednostavnost korištenja**	**skalabilnost**	**robusnost i pouzdanost**	**interoperabilnost**
izdvajanje spremišta multimedijalnog sadržaja	✓	x	x	✓	x	x
podrška dinamičkog označavanja sadržaja	x	x	✓	x	x	x
korištenje MOAT arhitekture za označavanje	✓	x	✓	✓	x	✓
korištenje ontologije za opis oznaka	✓	x	x	✓	x	✓
uvođenje RDFa oznaka u izgled	x	x	✓	x	x	x

Radi se o dijelovima arhitekture koji nisu obavezni za ostvarenje tražene funkcionalnosti aplikacija ove kategorije, ali su uvedeni kako bi se poboljšala kvaliteta arhitekture.

6.5. Semantička web aplikacija preporučenog sadržaja

Osim aplikacija koje su tijekom analize semantičkih web aplikacija eksplicitno svrstane u ovu kategoriju, pri analizi učestalih načina implementacija razmatrala su se i rješenja koja sadrže funkcionalnost preporučenog sadržaja, a dio su web portala ili drugih aplikacija. Predložena arhitektura obuhvaća samo problematiku preporuke sadržaja pa je za izradu semantičkih web portala ili aplikacija drugih kategorija koje implementiraju tu funkcionalnost potrebno kombinirati elemente više arhitektura predloženih u ovom istraživanju. Slika 60 prikazuje predloženu arhitekturu semantičke web aplikacije preporučenog sadržaja.



Slika 60 Arhitektura aplikacije preporučenog sadržaja

6.5.1. Razrada arhitekture web aplikacije preporučenog sadržaja

Postoje razni načini realizacije funkcionalnosti pronalaženja preporučenog sadržaja. Analizom aplikacija ustanovljeno je učestalo korištenje sljedećih modula:

- traženje sadržaja u skladu s preferencijama definiranim u profilu korisnika - primjeri su SIMAC [Herrera2006], Foafing the music [Celma2005] i MuseumFinland [Hyvönen2005]
- ocjenjivanje sadržaja od strane korisnika te nuđenje sadržaja pronalaženjem sličnosti između pozitivno ocijenjenih sadržaja - primjer je CHIP [Aroyo2007]
- praćenje ponašanja korisnika (kao alternativa ocjenjivanju – ako je korisnik pregledao sadržaj, znači da je za njega i zainteresiran) - primjeri su Foafing the music [Celma2005] i MuseumFinland [Hyvönen2005]

Poopćenje funkcionalnosti preporučenog sadržaja teško je realizirati. Za svaku domenu aplikacije potrebno je primijeniti posebne algoritme i načine realizacije navedenih modula. Na temelju analiziranih aplikacija te proučenog teorijskog okvira o načinima personalizacije [Anand2005] i tehnologijama semantičkog weba predlaže se rješenje u kojem su implementirani svi navedeni moduli uočeni u analiziranim aplikacijama. Pri njihovoj realizaciji koriste se semantičke tehnologije kako bi se poboljšala točnost postupka traženja personaliziranih preporuka. Kako bi bilo moguće implementirati navedene module, aplikacijom je potrebno podržati:

- praćenje ponašanja korisnika
- korisnički profil korisnika
- ontologija domene
- semantičko zaključivanje

Iako postoje brojne različite implementacije algoritama za predlaganje sadržaja korisniku, u ovom se istraživanju predlaže nadogradnja hibridnog postupka primjenom semantičkih tehnologija. Potrebno je spomenuti da su se u skorije vrijeme počeli javljati pristupi izračuna preporučenog sadržaja korištenjem oznaka. No, kako se radi o novijem pristupu za kojeg još nije dostupna dovoljno kvalitetna evaluacija, neće se koristiti u ovom istraživanju.

Osnovu ovdje predloženog postupka izračuna preporučenog sadržaja predstavlja kombinacija postupaka filtriranja sadržaja i skupnog filtriranja te se taj postupak oplemenjuje semantičkim zaključivanjem kao načinom izračuna preporučenog sadržaja. Osnova semantičkog hibridnog postupka izračuna preporučenog sadržaja je zaključivanje kojim se pronalaze veze između korisničkih zahtjeva i semantički

opisanih podataka domene aplikacije. Podaci za koje se ustanovi da su povezani s korisničkim zahtjevima predlažu se korisniku. Postupak se sastoji od dvije faze:

- filtriranje sadržaja - za donošenje odluka o predlaganju sadržaja korisniku u ovoj fazi, uz informacije o podacima i domeni podataka, koriste se i podaci o ponašanju korisnika. Korisniku se predlažu podaci slični podacima koje je prethodno pregledao, na temelju poveznica iz profila i DOI (degree of interest) indeksa. DOI indeksom se mjeri važnost podatka za korisnika. Vrijednosti indeksa mogu biti u rasponu [-1, 1], pri čemu -1 označava maksimalnu negativnu povezanost, a 1 maksimalnu pozitivnu povezanost korisnika i podatka. Dva podatka su slična ako sadrže iste ili slične atribute, ali i ako su semantički povezani preko razreda u hijerarhiji modela domene (npr. pripadaju istom razredu ili pripadaju razredima s istim nadrazredom). Moguće je uvođenje raznih metrika sličnosti podataka, ovisno što pokaže dobre rezultate u praksi. Uvođenje semantičke sličnosti rješava problem prevelike specijalizacije predloženih podataka, koji je karakterističan za filtriranje sadržaja.
- skupno filtriranje - ova faza započinje formiranjem susjedstva. Na temelju interesa korisnika pohranjenih u korisničkom profilu stvara se vektor ocjena kategorija, u kojem su pohranjene ocjene kategorija koje korisnik (prema profilu) smatra pozitivnim i negativnim. Kategorije se ocjenjuju prema DOI indeksu podataka koji im pripadaju. U susjedstvo se odabire određen broj korisnika koji su ocijenili većinu (npr. 70%) kategorija iz vektora ocjena kategorija i te kategorije su ocijenili slično ocjenama u vektoru ocjena kategorija korisnika. Interes korisnika predviđa se na temelju korisničkih profila susjeda i odabiru se podaci koji su slični podacima s kojima su susjedi povezani s pozitivnom vrijednošću DOI indeksa.

U sklopu opisanog postupka kombiniranjem faza skupnog filtriranja i filtriranja sadržaja postižu se bolji rezultati nego svakim od tih postupaka odvojeno [Anand2005]. Druga faza hibridnog postupka personalizacije zasnovanog na zaključivanju zasnovana je na skupnom filtriranju i izvršava se ako u prvoj fazi korisniku nisu ponuđeni podaci jer nijedan podatak ne zadovoljava stupanj sličnosti potreban za predlaganje. Postupak skupnog filtriranja koji se predlaže u sklopu ovog hibridnog načina personalizacije napredniji je od tradicionalnog skupnog filtriranja jer donosi točnije rezultate i rješava neke od problema tradicionalnog skupnog filtriranja. Na primjer, ne javlja se problem predlaganja potpuno novih sadržaja,

dakle onih koji još nisu mogli biti ocijenjeni jer su novi u aplikaciji. Taj problem eliminira se traženjem sličnosti kategorija podataka, a ne podataka koji im pripadaju.

6.5.1.1. Priprema modela

Na dizajn modela utječu sljedeće funkcionalnosti aplikacije preporučenog sadržaja: omogućiti praćenje ponašanja korisnika, uvesti semantičko spremište za ontologiju domene i podržati semantički profil korisnika.

Praćenje ponašanja korisnika

Ponašanje različitih korisnika se identificira analizom akcija koje korisnik poduzima tijekom korištenja aplikacije. Ovisno o tome koje se akcije prate (snimaju), a uobičajeno se uzima u obzir svaki pregled sadržaja, može doći do velikog broja podataka i učestalom unosu u spremište. Semantički repozitorij u tom slučaju nije prikladan. U praksi je učestalo rješenje otkrivanja ponašanja pregledom web logova. Potrebne podatke dobivene iz web logova potrebno je preoblikovati i pohraniti u semantičke zapise korištenjem ontologije koja opisuje ponašanje korisnika. Rad [Schmidt2007] sadrži primjer i način korištenja ontologije za opis ponašanja korisnika pri pronalaženju preporučenog sadržaja.

Semantički profil korisnika

Predložen postupak izračuna preporučenog sadržaja zahtijeva poznavanje interesa korisnika, kao i drugih osobnih podataka (npr. demografski podaci). Znači, potrebno je osigurati rukovanje podacima korisničkog profila. Problem privatnosti istaknut je problem aplikacija preporučenog sadržaja [Anand2005], pa je u rješenje potrebno integrirati *sustav za upravljanje identitetom* opisan na početku ovog poglavlja. Integracijom sustava za upravljanje identitetom korisnik sam može definirati koje ovlasti ima aplikacija nad njegovim osobnim podacima.

Semantički repozitorij

U semantičkom repozitoriju pohranjen je semantički opis domene nad kojom se preporuke vrše. Način primjene algoritma koji će se primijeniti za pronalaženje sličnosti teško je sugerirati (npr. određivanje semantičke sličnosti), jer značajno ovisi o domeni nad kojom se preporuke traže.

Detalji oko realizacije modela, dakle dohvat podataka, preslikavanje u RDF, kao i načina na koji se koristi sustav za upravljanje identitetom ne navode se ovdje posebno, budući da su već objašnjeni kod prethodno predloženih arhitektura.

6.5.1.2. Izgled

U slučaju kada se traži od korisnika da ocjenjuje sadržaje, aplikacija mora biti jako dinamična, odnosno mora koristiti paradigme bogatih internet aplikacija. U ovoj vrsti aplikacija interakcija s korisnikom je jako važna, budući da je vrijednost povratne informacije korisnika jako velika, tako da se posebna pažnja mora obratiti na kreiranje sučelja aplikacija.

6.5.1.3. Udovoljavanje tehničkim uvjetima postavljenim nad arhitekturu

Pregled svih odluka donesenih prilikom definiranja arhitekture u svrhu zadovoljavanja tehničkih uvjeta postavljenih nad arhitekturom prikazuje tablica 9. Radi se o odabiru elemenata arhitekture koji nisu obavezni za ostvarenje tražene funkcionalnosti aplikacija ove kategorije, ali su uvedeni kako bi se poboljšala kvaliteta arhitekture.

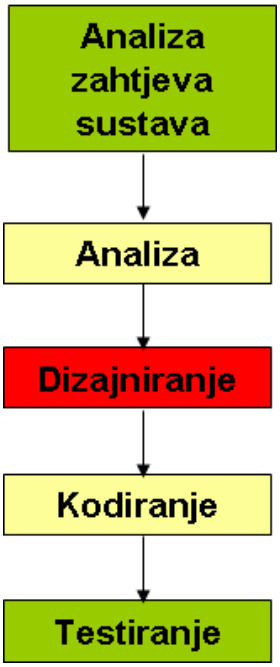
Tablica 9. Udovoljavanje tehničkim uvjetima postavljenim nad arhitekturu

	održavanje	**sigurnost**	**jednostavnost korištenja**	**skalabilnost**	**robusnost i pouzdanost**	**interoperabilnost**
uvođenje sustava za upravljanje identitetom	✓	✓	x	x	x	✓
izdvajanje repozitorija za praćenje ponašanja korisnika	✓	x	x	✓	✓	x
dinamika sučelja	x	x	✓	x	x	x

6.6. Razvojni proces semantičke web aplikacije

Moraš činiti stvari za koje misliš da ih ne možeš učiniti.
Eleanor Roosvelt

Na osnovu prijedloga i razrade arhitekture osnovnih kategorija semantičkih web aplikacija, mogu se prepoznati koraci njihovog razvojnog procesa. Slika 61 prikazuje uobičajene korake razvoja informacijskih sustava. Različitim bojama na slici ilustrirane su razlike između razvojnog procesa običnih Java web aplikacija i semantičkih Java web aplikacija temeljenih na MVC-u. Zelenom bojom označene su faze koje su iste za semantičke i klasične Java web aplikacije, žutom bojom faze koje se manje razlikuju i crvenom bojom faza koja se u potpunosti razlikuje.



Slika 61 Razvojni proces informacijskih sustava

Oblikovanje modela podataka, zadatak koji se znatno razlikuje kod izrade spomenutih aplikacija, odvija se kroz faze analize i dizajniranja. Faza kodiranja razlikuje se utoliko što su potrebna dodatna znanja za ostvarenje pristupa heterogenim izvorima podataka, transformiranje podataka u RDF, integriranje i samo rukovanje semantičkim (RDF) podacima.

Budući da je osnovna razlika ipak u oblikovanju modela podataka, u nastavku se detaljnije objašnjava što sve ta faza podrazumijeva kod razvoja semantičkih web aplikacija.

6.6.1. Oblikovanje modela podataka

Put do izrade fizičkog modela podataka podrazumijeva prikupljanje zahtjeva, izradu konceptualnog modela te potom utjelovljenje u neku od implementacija (npr. relacijska baza podataka). Postoje različite tehnike za izradu konceptualnih modela podataka na temelju prikupljenih zahtjeva, kao i za njihovu pretvorbu u logičke i fizičke modele. RDF je inspiriran prirodnim jezikom baš kao i tehnika modeliranja ORM (Object Role Modeling) pa se ORM nameće kao dobar odabir za izradu konceptualnih modela ontologija. Predloženo je i njegovo proširenje koje je prilagođeno korištenju za kreiranje ontologija [Jarrar2002]. ORM-om je moguće modelirati samo konceptualne modele podataka, dok je UML-om (Unified Modeling Language) moguće modelirati cijele sustave, pa se UML i nametnuo kao standard za modeliranje objektno-orijentiranih aplikacija. Iako je ORM semantički bogatiji model podataka, postoji niz njegovih nedostataka u odnosu na UML koji su prezentirani u radu [Fertalj2006].

Oblikovanje podataka u UML-u podrazumijeva korištenje dijagrama slučaja uporaba, dijagrama klasa i dijagrama objekata. Kako bi se moglo UML model podataka pretvoriti u zadani logički model (relacijski, objektno-relacijski, semantički, itd…) korištenjem zadanih tehnika modeliranja koriste se proširenja UML-a. U radu [Gasevic2004] predlaže se UML proširenje za pretvorbu UML dijagrama klasa u OWL model.

Kreiranje nove ontologije za potrebe aplikacije zapravo je rijedak slučaj u razvoju semantičkih web aplikacija. Dizajn semantičkog modela podataka ima sasvim novu dimenziju budući da su ontologije (RDF i OWL modeli podataka) namijenjene za dijeljenje i ponovnu iskoristivost [Uschold1998]. Zamišljeno je da postoje javno dostupni rječnici koji bi se onda za potrebe određene aplikacije povezivali i po potrebi dodatno proširivali. Dakle, dizajn ne počinje od nule, već se mora nastaviti na već postojeće globalne ontologije. Kratak pregled metodologija kreiranja ontologija, kao i osvrt na timove koji bi trebali sudjelovati u njenom nastajanju dan je u radu [Vranešić2009].

U idealnom se slučaju na osnovu zahtjeva prikupljenih od korisnika, što je uvijek neupitno prva faza razvoja sustava, pronađu dostupne ontologije (rječnici) koje

odgovaraju problematici. U slučaju kada takvi rječnici ne postoje moraju se stvoriti. Po svojoj prirodi razvoj semantičkih modela podataka sasvim je drugačiji od stvaranja modela podataka kao što su relacijski modeli ili XML modeli. Osnovna je razlika u tome što su modeli podataka koji su dosad bili uobičajeni u razvoju aplikacija orijentirani kao podrška obavljanju zadataka određene aplikacije za čije se potrebe rade i nisu namijenjeni dijeljenju. No, ontologije zbog postavljenih ciljeva ponovne iskoristivosti i dijeljenja, moraju biti općenite i što je moguće neovisnije o pojedinim zadacima i njihovoj konačnoj primjeni [Spyns2002]. Kako kod modeliranja svijeta oko nas stvari nisu uvijek tako jednoznačne, obično već pri izradi konceptualnih modela neke odluke donosimo na osnovu predviđene primjene tih pojmova, točnije na osnovu dogovora između osoba zaduženih za razvoj aplikacije i krajnjih korisnika. Takvih detalja u javno objavljenoj ontologiji ne smije biti. Predlaže se da se u aplikacijama koriste što više globalne, javno dostupne ontologije, koja moraju biti što općenitije, dok se za njihovu primjenu u aplikacijama uvodi potrebna razine specijalizacije [Jarrar2003]. Budući da rijetko postoji već dostupna ontologija koja pokriva cijelu domenu aplikacije, potrebno je uložiti trud u povezivanje i nadogradnju takvih javno dostupnih ontologija kako bi ih se moglo zajednički koristiti.

Dakle, razvojni proces semantičkih web aplikacija u sklopu faze analize i dizajna, uključuje i posebnu disciplinu *inženjerstvo ontologija*. Inženjerstvo ontologija podrazumijeva kreiranje, integriranje i nadogradnju ontologija za primjenu u nekoj specifičnoj aplikaciji. Opisane zadatke inženjerstva ontologija trebali bi provoditi eksperti tog područja, odnosno osobe s naprednim znanjem logike i jezika ontologija.

7.Prilagodba postojećih web aplikacija za semantički web

Kad god imam težak posao, dodijelim ga lijenoj osobi i sigurno će pronaći jednostavan način da ga obavi.

Walter Chrysler

Osnovni je cilj ovog istraživanja prijedlog procesa prilagodbe postojećih web aplikacija semantičkom webu. Tijek cjelokupnog procesa prilagodbe ovisi o identificiranim razlozima prilagodbe. Može postojati više razloga za poduzimanje prilagodbe određene aplikacije, te se za svaki razlog posebno predlaže metodologija prilagodbe. Primjena predloženih metodologija prema definiranim pravilima čini cjelokupni proces prilagodbe. U ovom poglavlju, osim definiranja tijeka cjelokupnog procesa prilagodbe postojećih web aplikacija semantičkom webu, daje se i detaljno objašnjenje svake metodologije pojedinačno, kao i ocjena njene primjenjivosti.

7.1. Postojeći pristupi

Uslijed nedostatka rezultata istraživanja na području inženjerstva semantičkih web aplikacija, dakle nedostatka cjelokupne slike izrade semantičkih web aplikacija, dosadašnji je rad drugih autora uglavnom bio koncentriran na prilagodbu pojedinih dijelova aplikacija semantičkom webu. Preciznije, značajniji napredak je postignut na području prilagođavanja modela podataka semantičkom webu i uvođenju semantike u prezentaciju aplikacije.

Na području preoblikovanja relacijske baze podataka u ontologiju rezultati su već odavno postignuti [Astrova2004, Stojanovic2002], a danas su dostupni i alati koji pomažu pri tom zadatku [Sahoo2009] te koji se mogu koristiti u praksi. Također, rezultati su ostvareni i na polju uvođenja semantike u prezentacijski sloj [Dill2003, Koivunen2005, Staab2001]. Nedostaci navedenih istraživanja istaknuti su u radu [Rovan2008b].

Za razliku od spomenutih istraživanja, ovo je istraživanje imalo za cilj pružiti potpunost rješenja prilagodbe web aplikacija, a ne samo jednog njihovog aspekta. U radu koji je prethodio ovom istraživanju [Rovan2006] analizirano je na koje sve dijelove višeslojne web aplikacije semantičke web tehnologije imaju utjecaj i koje bi se semantičke tehnologije mogle iskoristiti u pojedinom sloju. No, definiranje potpunog procesa prilagodbe moguće je bilo ostvariti samo uz prethodno provedeno

istraživanje inženjerstva semantičkih web aplikacija kako bi se došlo do rezultata nad kojima je moguće temeljiti daljnji rad.

7.2. Proces prilagodbe

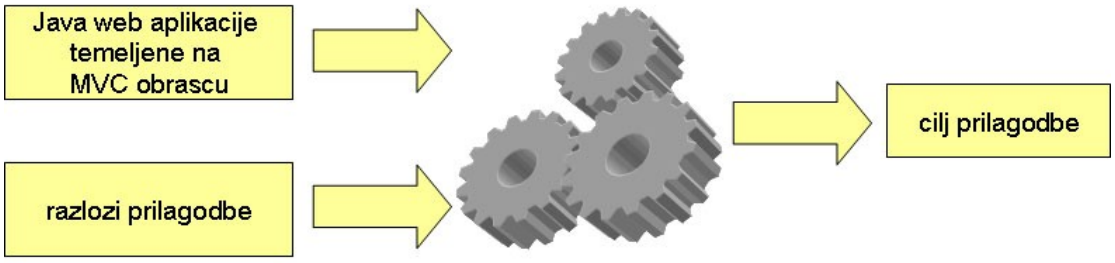
Izradi procesa prilagodbe moguće je pristupiti tek nakon što se definira što proces prilagodbe podrazumijeva.

> ***Definicija 7.1:*** *Pod* ***procesom prilagodbe*** *podrazumijeva se proces mijenjanja dizajna i implementacije sustava u svrhu udovoljavanja novo postavljenim uvjetima, ali da se pri tom ne degradira nijedna od postojećih funkcionalnosti aplikacije.*

Obično se podrazumijeva da je svaki proces prilagodbe popraćen metodologijom i alatom koji pomaže u tom procesu [Klischewski2006]. Pri tom se metodologija definira kao:

> ***Definicija 7.2:*** ***Metodologija*** *podrazumijeva metode, pravila i postulate discipline na koju se metodologija odnosi. Odnosno, metodologija je procedura ili skup procedura koje služe kao smjernice za ostvarenje nekog cilja* [Merriam2009].

Izrada prijedloga procesa prilagodbe zahtijeva definiciju *ulaza u proces*, *razloga prilagodbe* i *cilja prilagodbe* (slika 62).



Slika 62 Proces prilagodbe

Ulaz u proces prilagodbe predstavljaju web aplikacije koje se temelje na MVC obrascu i pisane su jezikom Java.

Analiza postojećih semantičkih rješenja istaknula je uobičajene funkcionalnosti web aplikacija koje implementirane semantičkim web tehnologijama pokazuju poboljšanja u odnosu na klasična rješenja. Dakle, radi se o funkcionalnostima koje su karakteristične i web aplikacijama, a ne samo semantičkim web aplikacijama. Motivacija za prilagodbu aplikacije semantičkom webu je potreba za implementacijom takve funkcionalnosti ili, ukoliko je već posjeduje, potreba za

poboljšanjem njenog izvođenja uvođenjem semantičkih web tehnologija. Osnovni prepoznati razlozi prilagodbe su sljedeći:

- poboljšavanje/uvođenje integracije više izvora podataka
- poboljšavanje/uvođenje funkcionalnosti pretraživanja
- poboljšavanje/uvođenje funkcionalnosti personalizacije
- poboljšavanje/uvođenje funkcionalnosti preporučenog sadržaja
- uvođenje značenja sadržaja u prezentaciju

Nije nužno da motiv za prilagodbu aplikacije bude samo jedan od navedenih razloga, već je moguće da u određenom slučaju postoji više razloga za poduzimanje prilagodbe aplikacije.

Rezultat procesa prilagodbe je semantička web aplikacija. Kako bi se od postojeće web aplikacije mogla napraviti semantička web aplikacija, potrebno je poznavati razvojni proces semantičkih web aplikacija i ta znanja primijeniti pri procesu prilagodbe. Arhitekture semantičkih web aplikacija predložene u ovom istraživanju služe kao predložak za njihovu izgradnju, dok su kroz razradu arhitektura dani osnovni napuci za njihovu realizaciju. Dakle, rezultati analize semantičkih web aplikacija koriste se pri definiranju načina realizacije procesa prilagodbe.

Arhitektura semantičkih web aplikacija predložena je posebno za svaku definiranu kategoriju. Na osnovu prepoznatih razloga prilagodbe postojeće aplikacije odabire se arhitektura odgovarajuće kategorije semantičkih web aplikacija te se biraju elementi koji se preuzimaju i implementiraju u svrhu prilagodbe. Aplikacija koja se dobije kao konačan rezultat procesa prilagodbe također se može svrstati u neku od kategorija semantičkih web aplikacija.

Predloženi proces prilagodbe popraćen je metodologijom, odnosno koracima koji govore *kada*, *kako* i *zašto* ih se poduzima. U slučajevima u kojima je procijenjeno da je moguće, predlaže se i korištenje odgovarajućeg alata (postojećeg i/ili izrađenog u okviru ovog istraživanja) koji pomaže u automatizaciji prilagodbe. Nad prijedlog procesa prilagodbe postavljeni su sljedeći uvjeti:

- zadržati postojeću funkcionalnost aplikacija
- uvesti minimalne potrebne promjene
- zadržati pozitivne karakteristike MVC obrasca

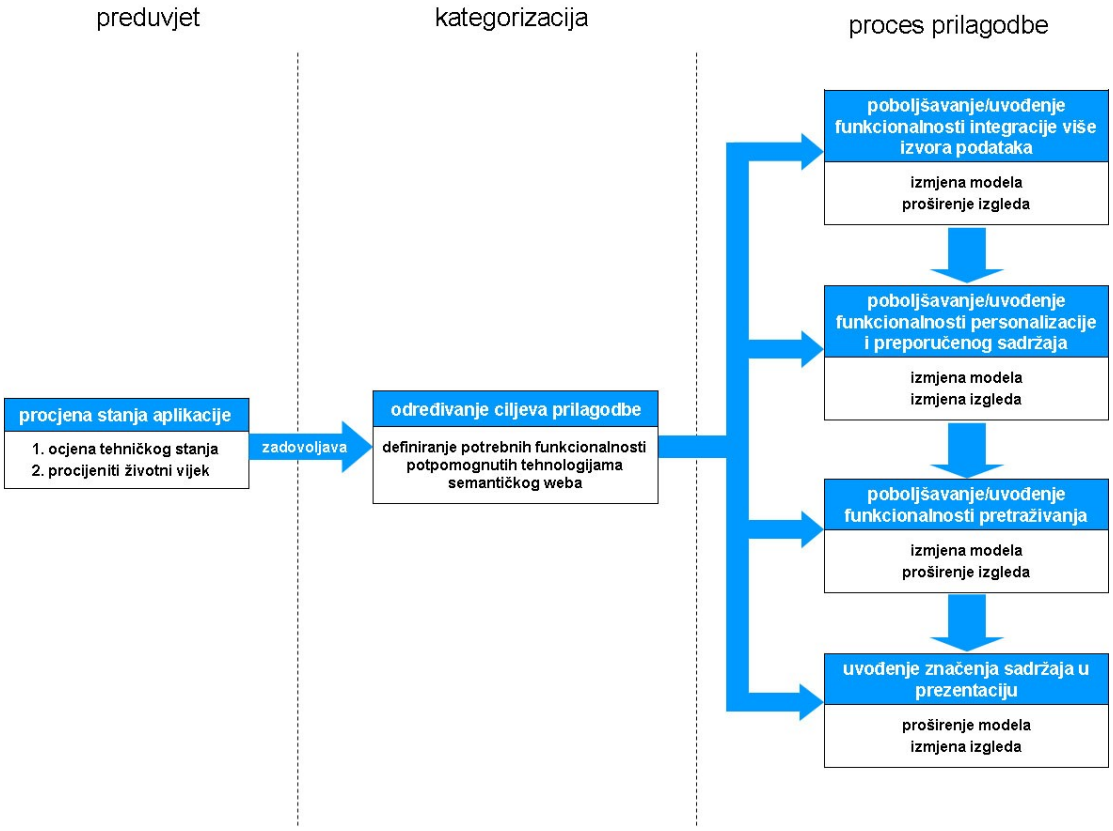
Zamišljeno je da proces prilagodbe obuhvati samo minimalne nužne promjene postojeće aplikacije kako bi prilagodbu bilo što jednostavnije provesti. Također, imperativ je da se sva postojeća funkcionalnost aplikacije zadrži uz eventualna

poboljšanja takvih funkcionalnosti. Uvođenje nužnih promjena ne smije narušiti pozitivne karakteristike MVC obrasca, pri čemu se posebna pažnja morala posvetiti očuvanju separacije uloga osoba uključenih u razvoj web aplikacija.

U nastavku je objašnjen cjelokupni proces prilagodbe postojećih web aplikacija semantičkom webu.

7.3. Realizacija procesa prilagodbe

Osnovna je zadaća procesa prilagodbe implementacija semantičkih tehnologija u postojeću web aplikaciju s ciljem unapređivanja poslovanja te aplikacije. Ne mora svaka aplikacija nužno biti pogodna za prilagodbu. Aplikacija mora zadovoljavati postavljene tehničke kriterije, te je dodatno potrebno ustanoviti očekuje li aplikaciju relativno dug životni vijek, jer u protivnom postoji vjerojatnost da prilagodba nije isplativa. Obavljanje takve provjere, ocjene tehničkog stanja aplikacije i procjene njenog životnog vijeka postavlja se kao preduvjet ulaska u proces prilagodbe. Samo ukoliko aplikacija zadovolji navedenu provjeru može se podvrgnuti i samom procesu prilagodbe.



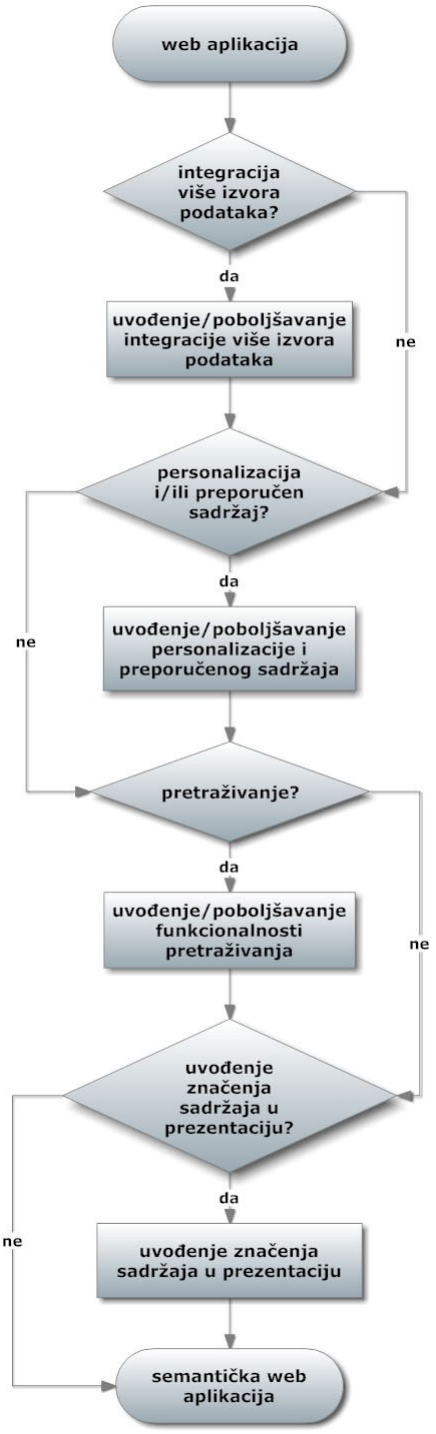
Slika 63 Proces prilagodbe web aplikacija semantičkom webu

Predlaganje odgovarajućeg tijeka procesa prilagodbe zahtijeva prethodno obavljanje kategorizacije ulaznih aplikacija na osnovu specificiranih razloga za poduzimanje prilagodbe. Slika 63 ilustrira korake cjelokupnog opisanog procesa.

Kako implementacija svake pojedine funkcionalnosti (razloga prilagodbe) podrazumijeva primjenu posebne metodologije, kada se primjenjuje više njih, onda je promjene potrebno poduzimati prema sljedećoj prioritetnoj listi:

1. poboljšavanje/uvođenje integracije više izvora podataka
2. poboljšavanje/uvođenje funkcionalnosti personalizacije
3. poboljšavanje/uvođenje funkcionalnosti preporučenog sadržaja
4. poboljšavanje/uvođenje funkcionalnosti pretraživanja
5. uvođenje značenja prikazanog sadržaja

Kroz kategorizaciju postojećih aplikacija, na osnovu specificiranih razloga prilagodbe, određuje se ulaz u proces prilagodbe te slijed koraka koji se dalje poduzimaju. Slika 64 prikazuje redoslijed po kojem se poduzimaju prilagodbe aplikacije.



Slika 64 Tijek procesa prilagodbe

U nastavku, pojašnjeni su koraci cjelokupnog predloženog procesa prilagodbe i dana je ocjena primjenjivosti predloženih metodologija prema kojima se obavlja proces prilagodbe.

7.4. Procjena stanja aplikacije

Prije no što se krene s provedbom procesa prilagodbe aplikacije, potrebno je procijeniti isplativost poduzimanja takvog zahvata. Odnosno, potrebno je provesti analizu troškova i koristi koja bi pokazala očekivanu isplativost poduzimanja promjena.

Proces prilagodbe namijenjen je aplikacijama koje se temelje na MVC obrascu i koje su pisane jezikom Java. Kao prvi korak potrebno je provjeriti udovoljava li aplikacija tim zadanim kriterijima. Čest je slučaj da je inicijalno aplikacija i bila rađena prema MVC obrascu, no da je tijekom održavanja aplikacije došlo do degradacije njene logičke arhitekture. Dakle, nužno je provesti reverzni inženjering kojim bi se utvrdilo zadovoljava li aplikacija svojom logičkom arhitekturom MVC obrazac.

Kao osnovni kriterij procjene isplativosti poduzimanja procesa prilagodbe uzima se očekivani životni vijek aplikacije. Životni vijek aplikacije ovisi o zaradi koja se tom aplikacijom ostvaruje, kao i o troškovima njenog održavanja te on prestaje u trenutku kada se iznos zarade i troškova održavanja približe [Wiederhold2006]. Dakle, procjena životnog vijeka aplikacije svodi se na procjenu iznosa zarade i troškova održavanja aplikacije.

Izračun zarade ovisi o mnogim parametrima, te se znatno razlikuje u slučaju kad se radi o aplikacijama koje su razvijane striktno za kupce i u slučaju kada su razvijene od strane organizacije za čije se potrebe rade [Wiederhold2006]. Također, kod velikih sustava značajan prihod se može ostvarivati i od naknada održavanja. Procjena iznosa zarade je vrlo kompleksan izračun na koji više utječu ekonomski parametri, nego samo tehničko stanje aplikacije.

Osim procjene zarade, drugi važan kriterij procjene životnog vijeka aplikacije je procjena troška održavanja. Brojne studije su potvrdile da postoji uska veza između održavanja, načina dizajniranja i implementacije aplikacija [Feng2006, Schneidewind1987]. Dakle, tehničko stanje aplikacije ima direktan utjecaj na trošak održavanja. Očito da proces prilagodbe aplikacije semantičkom webu također utječe na trošak održavanja budući da se njegovim poduzimanjem mijenja tehničko stanje aplikacije. U svrhu procjene troškova održavanja potrebno je provesti analizu 'što-ako'. Implementacijom novih tehnologija sustav raste, pa se očekuje da će i njegovo održavanje biti zahtjevnije. No, također je moguće da se kroz prilagodbu implementiraju novi moduli i tako zamijene stari koji su stvarali značajan trošak održavanja uslijed lošeg dizajna ili izvedbe. Aproksimaciju je teško pružiti, jer se

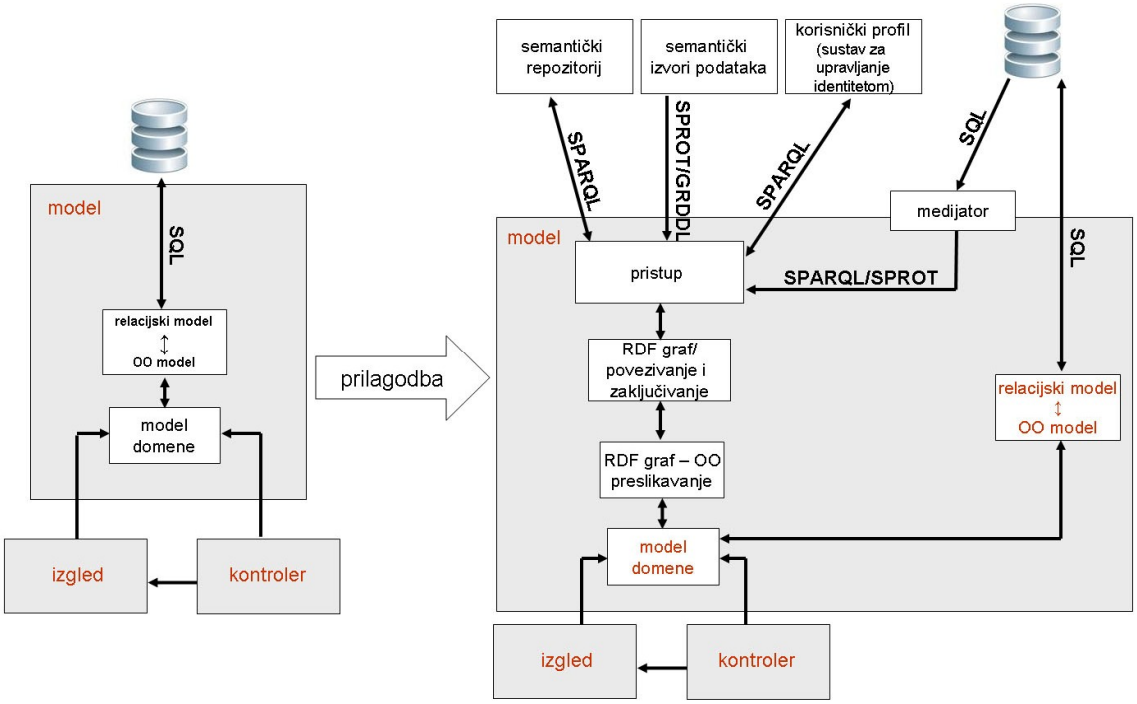
procjena temelji na parametrima koje je gotovo nemoguće izmjeriti, kao na primjer kvaliteta i fluktuacija osoba koje održavaju aplikaciju.

MVC obrazac je preporučen dizajn za Java web aplikacije zbog lakog održavanja i proširivanja [Singh2002]. Upravo iz tog razloga kao predmet analize za prilagodbu uzimaju se samo aplikacije koje se temelje na MVC obrascu, jer obećavaju dulji životni vijek. Kako bi primjena semantičkih tehnologija u postojeće rješenje imala što manji utjecaj na eventualno povećanje troška održavanja, proces prilagodbe je osmišljen na način da aplikacija i nakon njegovog izvršavanja zadrži logičku arhitekturu prema MVC obrascu.

7.5. Poboljšavanje/uvođenje integracije više izvora podataka

Potreba za web aplikacijama koje djeluju nad integriranim skupom podataka iz više izvora podataka sve je češća, pa tako i potreba za uvođenjem integracije dodatnih izvora podataka u već postojeće aplikacije. Od samog početka razvoja semantičkog weba ističe se korisnost semantičkih web tehnologija pri obavljanju integracije heterogenih izvora podataka. Kako se radi o problemu koji se odavno proučava, postoje rezultati na tom području u obliku raznih alata i tehnika, koji se u većoj mjeri mogu koristiti kao podrška ovom procesu prilagodbe. Integracija više izvora podataka uključena je u prijedlog arhitekture semantičkog web portala, tako da se za ostvarenje ovog dijela procesa prilagodbe za izgradnju ciljne web aplikacije preuzimaju elementi arhitekture *semantičkog web portala* (slika 52).

Problem koji se ovdje proučava jest kako klasičnu arhitekturu koja se temelji na samo jednom izvoru podataka, a pri tom se pretpostavlja relacijskoj bazi podataka, proširiti elementima opisanim u sklopu semantičkog web portala kojima se ostvaruje integracija s drugim izvorima podataka (slika 65). Ciljna arhitektura gotovo je identična onoj semantičkog web portala koju prikazuje slika 52. Kako je pretpostavka da se polazna arhitektura temelji na samo jednom nesemantičkom izvoru podataka, ciljna arhitektura također sadržava samo relacijsku bazu podataka kao nesemantički izvor. Dodatna razlika u odnosu na predloženu arhitekturu semantičkog web portala je što se zadržava iz arhitekture postojeće aplikacije klasičan, SQL pristup postojećoj relacijskoj bazi iz razloga koji će biti kasnije objašnjeni u poglavlju 7.5.1.1.



Slika 65 Polazna i ciljna arhitektura procesa prilagodbe poboljšavanje/uvođenje integracije više izvora podataka

Većina zadataka koji se trebaju obaviti kod uvođenja podrške integracije više izvora podataka u postojeću web aplikaciju preuzimaju se iz inženjerstva semantičkih web portala i objašnjeni su prilikom razrade arhitekture semantičkog web portala. Kod ovog procesa prilagodbe naglasak se stavlja na semantičko opisivanje postojećeg modela podataka i obavljanja minimalnih izmjena u aplikaciji koje omogućuju uspješno izvođenje aplikacije nad takvim semantičkim modelom.

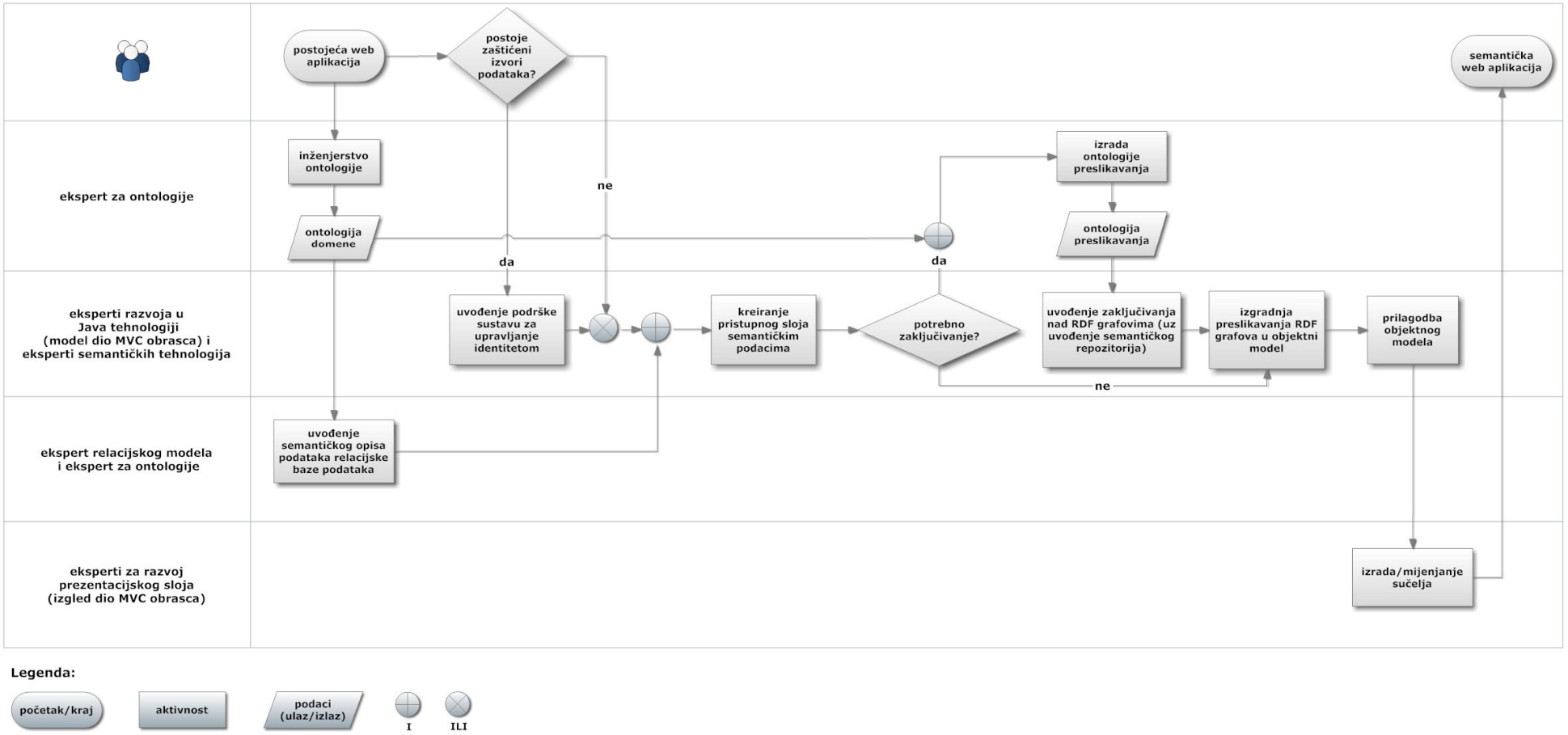
7.5.1. Metodologija prilagodbe

Uvođenje integracije u postojeću aplikaciju zahtijeva poduzimanje aktivnosti iz inženjerstva semantičkih web aplikacija, te dodatno semantičko opisivanje podataka postojeće relacijske baze, kao i povezivanje novih elemenata arhitekture s već postojećom infrastrukturom.

Primjena predložene metodologije prilagodbe podrazumijeva poduzimanje aktivnosti koje prikazuje tablica 10. Za svaku aktivnost navedeno je koje joj aktivnosti moraju prethoditi te koji se predloženi postupak u ovom istraživanju mora konzultirati oko detalja realizacije.

Tablica 10. Metodologija prilagodbe uvođenja integracije više izvora podataka

broj aktivnosti	preduvjeti	opis aktivnosti	pripadnost aktivnosti prethodno predloženom postupku
1	nema	inženjerstvo ontologije – pronaći dostupne ontologije koje pokrivaju sadašnji model podataka, izgraditi dodatne ontologije i međusobno ih povezati	inženjerstvo semantičkih web aplikacija
2	nema	omogućiti pristup zaštićenim podacima – uvođenje podrške sustavu za upravljanje identitetom	inženjerstvo semantičkih web aplikacija
3	1	podatke postojeće relacijske baze podataka semantički opisati	proces prilagodbe integracije više izvora podataka
4	1	izraditi "ontologiju preslikavanja" u svrhu povezivanja RDF grafova	inženjerstvo semantičkih web aplikacija
5	2, 3	kreirati pristupni sloj semantičkim podacima	inženjerstvo semantičkih web aplikacija
6	4, 5	omogućiti zaključivanje (povezivanje) nad dobivenim RDF grafovima – podrazumijeva i uvođenje semantičkog repozitorija	inženjerstvo semantičkih web aplikacija
7	6	izgraditi sloj preslikavanja RDF grafova u objektni model (uz prilagodbu objektnog modela)	inženjerstvo semantičkih web aplikacija
8	7	dodatno prilagoditi objektni model (udruživanje objekata iz više izvora)	proces prilagodbe integracije više izvora podataka
9	8	izgraditi/izmijeniti sučelje	proces prilagodbe integracije više izvora podataka



Slika 66 Tijek procesa prilagodbe uvođenja integracije više izvora podataka u postojeće aplikacije po zaduženjima

Pojedine aktivnosti procesa prilagodbe mogu se izvoditi paralelno, a neke nije nužno provoditi u svim slučajevima kao što to prikazuje slika 66. Na slici je također naznačeno koja znanja trebaju posjedovati osobe angažirane za obavljanje pojedine aktivnosti.

U nastavku se detaljno razmatraju samo aktivnosti koje su specifične prilagodbi aplikacije: semantičko opisivanje podataka relacijske baze podataka, kao i integracija novih arhitektonskih elemenata s postojećim. Posebna se pažnja daje načinima preslikavanja podataka relacijske baze u semantičke podatke, budući da upravo taj korak predstavlja srž problema.

7.5.1.1. Preslikavanje podataka relacijske baze podataka u semantičke podatke (RDF)

Integracija podataka iz više heterogenih izvora ostvaruje se tek nakon što se podaci iz svih izvora semantički opišu. Dakle, prvi izazov koji se nameće kod ove vrste prilagodbe je kako podatke postojeće relacijske baze semantički opisati da ih se može integrirati s drugim izvorima podataka. Prvenstveno, potrebno je jasno definirati razlike između relacijskog i semantičkog modela.

Razlika modela

U suštini, preslikavanje relacijskog modela podataka u semantički model izvodi se na temelju sličnosti njihovih konceptualnih E-R modela. U relacijskom modelu, svaki entitet predstavljen je kao tablica u bazi podataka, svako svojstvo (atribut) entiteta postaje stupac u toj tablici, a veze između entiteta određene su stranim ključevima. Svaki redak u tablici opisuje instancu entiteta, jedinstveno identificiranu primarnim ključem. U ekvivalentnom semantičkom modelu istog konceptualnog modela vrijedi da je entitet subjekt označen jedinstvenim identifikatorom (URI-em), predikat predstavlja svojstvo entiteta, a objekt vrijednost svojstva. Veze između entiteta su veze između subjekata.

Dakle, kako bi se dobio ekvivalentan semantički model relacijskog modela implementiranog u relacijskoj bazi podataka, osnovni algoritam je sljedeći:

1. stvoriti RDFS razred za svaku relaciju
2. pretvoriti sve primarne i strane ključeve u URI-e
3. dodijeliti URI (predikat) svakom atributu
4. dodijeliti predikat tipa rdf:type za svaku n-torku i povezati ga s URI-em RDFS razreda koji odgovara toj tablici
5. za svaki atribut koji nije dio niti primarnog niti stranog ključa, stvoriti trojku koja sadrži URI primarnog ključa kao subjekt, URI atributa kao predikat i vrijednost atributa kao objekt

Navedeni postupak osnova je za sve postupke preslikavanja podataka relacijskih baza u RDF podatke. Različiti alati koriste različite nadogradnje tog osnovnog principa, a opisivanje dodatnih algoritama prelazi okvire ovog istraživanja. U radovima [Byrne2009, Tirmizi2008] izneseni su problemi koji mogu nastupiti tijekom automatskog preslikavanja uslijed razlika u suštini modela i koji ukazuju na to da je ipak potrebna intervencija korisnika tijekom preslikavanja. Potpuna automatizacija procesa u vidu automatskog preslikavanja modela i podataka, uz pomoć alata bez intervencije korisnika je nemoguća. Zbog toga je vrlo važno da su osobe koje provode proces prilagodbe relacijske baze podataka semantičkom modelu dobro upoznate sa svim razlikama tih modela, kako bi tijekom prilagodbe mogle donositi kvalitetne odluke. U nastavku navode se i kratko analiziraju samo najčešći primjeri problema koji se mogu javiti pri preslikavanju modela.

1. Kompozitni primarni ključevi

U relacijskom modelu kompozitni primarni ključevi predstavljaju sasvim dobar dizajn, ali u semantičkom modelu identifikator mora biti samo jedna vrijednost, a ne kombinacija više njih, tako da se predlaže uvođenje surogatnih ključeva pri procesu preslikavanja.

2. Relacije nastale iz N:N veza E-R modela

Slijedeći predložen algoritam, preslikavanje modela koji uključuje relaciju koja je nastala iz N:N veze u E-R modelu rezultirao bi redundantnim resursima kao što prikazuje slika 67 (redundantan resurs označen je crvenom bojom). Ne bi se trebali stvarati novi resursi na osnovu takvih relacija, već bi one trebale samo predstavljati predikat između resursa koji su stvoreni na osnovu relacija na koje se referencira relacija nastala iz N:N veze.

predmet

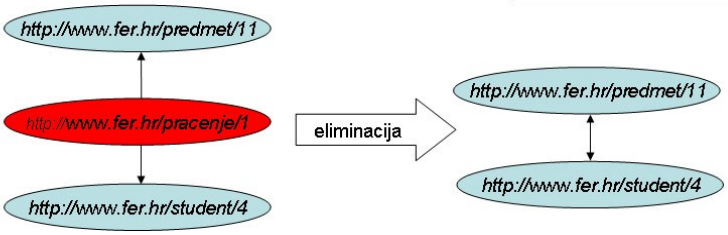
idPredmet	nazPredmet	kratPredmet
11	Programiranje i programsko inženjerstvo	PiPI
12	Matematika 1	Mat1
...	...	...

student

idStudent	ime	prezime
4	Lidia	Rovan
6	Zoran	Rovan
...	...	...

pracenjeNastave

idNastavaPracenje	idStudent	idPredmet
1	4	11
2	4	12
3	6	11
...	...	...



Slika 67 Eliminiranje nepotrebnih čvorova

3. Nasljeđivanje

U relacijskim modelima ne postoji eksplicitno iskazivanje nasljeđivanja, već postoje različiti načini na koje se nasljeđivanje može implementirati. Razlikuju se tri načina implementacije nasljeđivanja, a od toga moguće je automatizirati samo jednog od njih. Dakle, nasljeđivanje se može riješiti na načine:

- Relacije specijalizacije imaju vlastite primarne ključeve uz strane ključeve koji pokazuju na primaran ključ relacije generalizacije: na ovaj način se realiziraju i druge vrste veza, pa je nemoguće automatizirati kreiranje nasljeđivanja u ontologiji. Primjer je odnos relacija predmet i orgjed koji prikazuje slika 69.
- Relacije specijalizacije i generalizacije imaju jednake primarne ključeve. Kod specijalizacije taj ključ je ujedno i strani ključ koji pokazuje na primaran ključ generalizacije. Ovaj način realizacije se može preslikati u nasljeđivanje u ontologiji.

- Nastaje jedna relacija koja sadrži atribute generalizacije i specijalizacije. Na ovaj način, također je nemoguće prepoznati nasljeđivanje budući da svi atributi pripadaju jednoj relaciji, pa ništa ne ukazuje na postojanje nasljeđivanja.

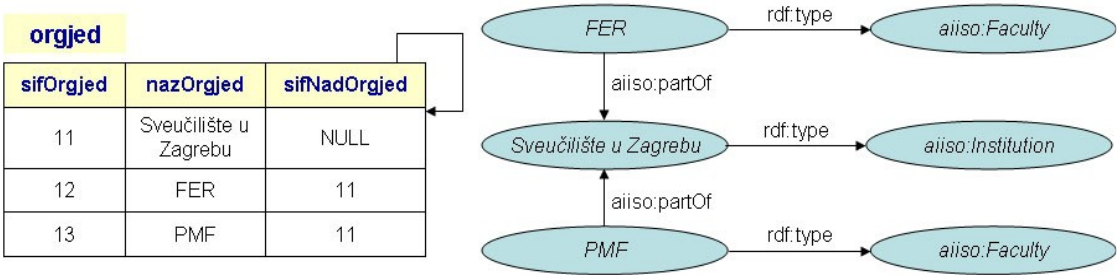
4. Karakteristike veza

U relacijskom modelu ne mogu se definirati karakteristike veza, kao na primjer tranzitivna veza ili simetrična veza. Takve veze nikako ne mogu biti rezultat automatskog preslikavanja modela te se moraju kasnije ručno dodati.

5. NULL vrijednosti

NULL vrijednosti se po opisanom algoritmu ne preslikavaju u RDF trojke, ali one ipak nekad imaju značenje i potrebno je poznavanje relacijskog modela da bi se značenje moglo implementirati u RDF model.

Slika 68 prikazuje primjer u kojem NULL vrijednost ne znači da podatak nije poznat, nego ima značenje da organizacijska jedinica nema svoju nadređenu te ovisno o ontologiji u koju se preslikava može značiti da ta n-torka pripada drugoj RDFS klasi u odnosu na ostale n-torke relacije.



Slika 68 Primjer značenja NULL vrijednosti

6. Kodirane vrijednosti

U relacijama se ponekad nalaze kodirane vrijednosti, dakle skraćenice uz koje se veže neko značenje. Na primjer za spol se evidentira slovo M ili Ž umjesto punog naziva. Uvoditi vrijednost takvog atributa u RDF model nema smisla, pa bi se trebalo uvesti pravilo koje za takve atribute uvodi drugo značenje.

Iz navedenih primjera vidljivo je da proces preslikavanja nije sasvim jednoznačan i zahtijeva intervenciju korisnika kako bi se uspješno proveo. Osim navedenih problema, velik problem automatskih preslikavanja predstavlja što su u konačnici podaci opisani ontologijom koja je stvorena tijekom preslikavanja (npr. relacija

predmet postaje owl klasa predmet, naziv predmeta svojstvo predmet:nazPredmet, itd…). Takva ontologija nije zadovoljavajuća u konceptu mreže podataka. Načela povezanih podataka nalažu da se ontologije međusobno dijele i da se uvedu veze između shema i samih podataka. Znači da je nastojanje da se preslikavanje relacijskog modela izvrši u već postojeće ontologije, koliko je to god moguće.

Osim navedenih problema preslikavanja modela i podataka, postoji i suštinska razlika u karakteristikama modela koja se manifestira kod zaključivanja nad podacima opisanim takvim modelima. Naime, zaključivanje nad podacima relacijske baze podataka podrazumijeva načelo „zatvorenog svijeta“. Dakle, sve ono što ne postoji u relacijskoj bazi smatra se da je pogrešno, dok je semantički web vođen načelom „otvorenog svijeta“, odnosno sve ono što nije sadržano u bazi znanja smatra se nepoznatim. Takvo ponašanje je karakteristično za baze znanja, koje dopuštaju nepotpune modele kako bi nad njima primjenom zaključivanja mogli doći do novog znanja. Navedena razlika utječe na to da koncept ograničenja ima sasvim različito značenje u ta dva svijeta. Na primjer, pravilo stranog ključa između relacija orgjed i predmet, koje prikazuje slika 69, spriječilo bi unos n-torke (1102, 'Matematika 1', 'Mat1', 1209) u relaciju predmet budući da ne postoji zapis u relaciji orgjed koji odgovara organizacijskoj jedinici sa šifrom 1209. To isto ograničenje u semantičkom modelu bi se preslikalo na način da postoji veza između predmeta i organizacijske jedinice (aiiso:teaches) koja kao domenu (rdfs:domain) definira klase koje odgovaraju organizacijskim jedinicama, a kao raspon klase (rdfs:range) koje odgovaraju predmetima. Unos trojke

<*prefix:predmet/1209* aiiso:teaches *prefix:*predmet/1102>

u bazu znanja ne bi se spriječio, jer resurs *prefix:predmet/1209* predstavlja instancu klase predmet, a ne organizacijske jedinice, već bi zaključivanjem došlo do zaključka da taj resurs odgovara klasi organizacijske jedinice, budući da upravo ta klasa odgovara domeni veze (svojstva) aiiso:teaches.

orgjed

sifOrgjed	nazOrgjed
12	FER
13	PMF

predmet

idPredmet	nazPredmet	kratPredmet	sifOrgjed
1100	Programiranje i programsko inženjerstvo	PiPI	12
1209	Linearna algebra	LA	13

Slika 69 Primjer referencirane relacije

Moguće je uvesti ograničenja u semantički model koja bi zatvorila svijet. Za navedeni primjer to bi značilo da se mora uvesti pravilo da su klase organizacijske jedinice i predmeta međusobno disjunktne. Budući da uvjeti koji su postavljeni nad proces prilagodbe nalažu da se zadrži funkcionalnost aplikacije koja se prilagođava, moraju se uvesti dodatna pravila koja „zatvaraju svijet" kako se ne bi narušilo poslovanje aplikacije.

Realizacija preslikavanja

Preslikavanje relacijskog modela podataka u semantički model, odnosno ontologiju, predmet je istraživanja i duže nego što traje sama ideja semantičkog weba. Trenutno je interes u tom području jako velik, tako je osnovana i W3C grupa[40] koja nadzire njegov razvoj. Budući se istraživanje u tom području provodi već odavno dostupni su alati koji mogu pomoći kao potpora pri provođenju procesa prilagodbe. Recentan rad [Sahoo2009] sadrži pregled svih dostupnih metoda i alata za preslikavanje relacijskih modela u RDF/OWL modele (npr. D2RQ [Bizer2007], Dartgrid [Wu2006b], R_2O [Barrasa2004], Virtuoso, itd…). Alati koji ne nude automatiziran proces, na prethodno navedene probleme preslikavanja odgovaraju na način da su razvili svoje posebne jezike u kojima se definiraju pravila preslikavanja. Na taj način, korisnik u potpunosti kontrolira što će se i na koji način preslikati, a u mogućnosti je i definirati proizvoljne ontologije kao cilj preslikavanja.

Postojeće pristupe preslikavanja može se podijeliti prema vrsti trajnog spremišta semantičkih podataka i njihov odabir direktno utječe na konačan izgled arhitekture aplikacije:

- svi podaci iz relacijske baze podataka pretvaraju se u semantičke zapise i kao takvi pohranjuju se u trajno spremište
- uvodi se medijator pomoću kojeg se na zahtjev može dohvatiti semantički opisan podatak iz relacijske baze podataka

Prvi način transformiranja, koji eliminira daljnju potrebu za relacijskom bazom, ima više nedostataka:

1. trenutno stanje razvoja trajnih spremišta semantičkih podataka i posebno SPARQL-a kao načina pristupa podacima ne odgovara na zahtjeve današnjih aplikacija

[40] http://www.w3.org/2005/Incubator/rdb2rdf/

2. moguće je da druge aplikacije također koriste podatke iz relacijske baze podataka koja se želi transformirati, pa bi trebalo i sve takve aplikacije prilagoditi za pristup semantičkim podacima

Uvjet postavljen nad proces prilagodbe nalaže da se obavezno mora zadržati postojeća funkcionalnost aplikacije. Kako su trenutno ograničene mogućnosti trajnih spremišta semantičkih podataka i upitnog jezika SPARQL, potpuna bi eliminacija relacijske baze podataka narušila performanse obavljanja poslovanja aplikacije pa se kao imperativ postavlja zadržavanje relacijske baze podataka i uvođenje medijatora.

Medijator radi na načelu preslikavanja SPARQL upita u odgovarajuće SQL upite. Dakle, relacijska baza podataka ostaje kao element arhitekture, ali se stvara mogućnost pristupa toj istoj bazi korištenjem SPARQL upita. Definirana preslikavanja, napravljena u skladu s prethodno objašnjenim algoritmom, koriste se kao pravila po kojima se izvršava pretvorba SPARQL upita u SQL upite.

U skladu s uvjetima koji su postavljeni na ovo istraživanje, medijator koji je predložen u procesu prilagodbe mora zadovoljiti novije pravce istraživanja semantičkog weba, dakle da se pomoću njega semantički opisani podaci mogu objaviti prema načelima povezanih podataka. Postoje dostupni alati koji udovoljavaju svim navedenim uvjetima (npr. D2R Server).

7.5.2. Promjene MVC obrasca - model

Osnovna razlika koja se uvodi na dio MVC obrasca koji se odnosi na model je uvođenje pristupa prema više izvora podataka i to međusobno različitih vrsta izvora podataka. Osim što je potrebno uključiti i ostvariti pristup prema novim izvorima podataka, dodatno je potrebno omogućiti pristup semantičkim tehnologijama postojećoj relacijskoj bazi podataka. Relacijskoj bazi podataka daje se semantički opis prema prethodno objašnjenom pristupu. Važno je napomenuti da nije potrebno sve podatke relacijske baze semantički opisati, nego samo one koji se koriste pri integraciji. Također, potrebno je očuvati postojeći pristup relacijskoj bazi podataka budući je zaključeno da ona mora i dalje egzistirati kao element arhitekture. Prema navedenom, u dijelu MVC obrasca koji se odnosi na model moraju istovremeno postojati sljedeći različiti načini pristupa podacima:

- SQL upiti nad podacima pohranjenim u relacijskoj bazi podataka
- SPARQL (SPROT) upiti nad semantički opisanim podacima relacijske baze podataka

- SPROT, SPARQL ili GRDDL metoda pristupa prema semantički opisanim podacima svih ostalih izvora podataka
- SPROT upiti po predloženoj proširenoj specifikaciji (OpenID & OAuth) za dohvat podataka iz zaštićenih izvora

Kreiranje pristupnog sloja prema navedenim izvorima podataka obavlja se na isti način kao i kod izrade semantičkog web portala. Nakon što se kreira pristupni sloj, potrebno je korištenjem ontologije preslikavanja omogućiti obavljanje zaključivanja nad dohvaćenim podacima, kako u svrhu otkrivanja novog znanja, tako i u svrhu povezivanja podataka. Željeni podaci nakon obavljenog preslikavanja moraju se iz RDF grafova preslikati u objektni model kako bi se mogli dalje koristiti u aplikaciji. Svi navedeni postupci poznati su iz inženjerstva semantičkog web portala i objašnjeni u poglavlju 6.3 pa se neće ovdje detaljno razmatrati.

Uvođenje novog izvora podataka znači uvođenje dodatnih podataka, pa tako i proširenje objektnog modela kojeg aplikacija koristi. U sustavima koji djeluju nad integriranim izvorima podataka podaci pohranjeni u objektima potječu iz raznih izvora podataka, pa se može dogoditi i da se svojstva samo jednog objekta nalaze u modelima dva ili više izvora podataka. Dakle, važno je uočiti da se podaci svih integriranih izvora na kraju preslikaju u samo jedan, zajednički, objektni model.

U postojećoj aplikaciji podaci koje objekti sadržavaju dohvaćaju se iz relacijske baze podataka. Kako bi se mogle dohvaćati ispravne instance podataka, objekti sadrže svojstva koja čine primaran ključ te se na taj način vrši izjednačavanje objekata i podataka u relacijskoj bazi podataka. Sličan način potrebno je primijeniti i za dohvat instanci iz semantičkih izvora. Predlaže se da svaki objekt koji sadrži svojstva iz semantičkih izvora podataka naslijedi klasu RDFObject koja zadovoljava specifikacije JavaBean-a:

```
public class RDFObject {

    private String URI;

    public String getURI() {
        return URI;
    }

    public void setURI(String URI) {
        this.URI = URI;
    }

}
```

Slika 70 Klasa RDFObject

Na taj način osigurava se da svaki objekt posjeduje identifikator koji mu odgovara u semantičkom webu.

Kod integracije podataka koja zahtijeva potpunu točnost javlja se problem povezivanja podataka dohvaćenih direktno iz relacijske baze podataka i onih dohvaćenih iz semantičkih izvora. Problem je kako postaviti URI objektima korištenim u aplikaciji, a dohvaćenim iz nesemantičkih izvora, kako bi se korištenjem tih istih URI-a mogle dohvatiti točne instance iz semantičkih izvora.

Moguće rješenje je da se URI objekta generira pomoću vrijednosti njegovih svojstava. Na primjer, URI predmeta se generira pomoću osnove web puta: *http://server/*, dijela koji određuje o kojem se entitetu radi: *predmet* i šifre predmeta koja određuje točno predmet o kojem se radi: *123131*. Sadržaj cijelog URI-a za navedeni primjer je: *http://server/predmet/123131*. Primjer generiranja URI–a za objekt prikazuje slika 75. Očito je da je za ovakvo generiranje URI-a unutar aplikacije potrebno poznavati obrazac po kojem se oni izgrađuju. Također, važno je primijetiti da nije dozvoljena autonomija aplikacija od semantičkih izvora podataka, budući da obrasci po kojem se izgrađuju URI-i u oba slučaja moraju biti jednaki.

Drugo moguće rješenje je u kontradikciji s pravilima kreiranja semantičkih modela (ontologija). Radi se o pristupu u kojem se u semantički model dodijele resursima svojstva koja odgovaraju ključevima u relacijskoj bazi podataka (npr. redni broj), iako oni nisu prirodna svojstva entiteta koji se modelira. Takva svojstva je onda donekle opravdano koristiti lokalno unutar aplikacije, jer njihovo globalno značenje nije poznato niti ga se treba uvoditi.

7.5.3. Promjene MVC obrasca - izgled

U izgled dijelu MVC obrasca potrebno je uvesti promjene kojima se prezentiraju novi integrirani podaci. Dakle, potrebno je prilagoditi postojeće web (JSP) stranice novim funkcionalnostima te izgraditi nove koje će sadržavati pregled nad integriranim podacima. Razvojna paradigma je potpuno ista onoj prema kojoj je izgrađena i postojeća aplikacija.

7.5.4. Ocjena primjenjivosti

7.5.4.1. Opis problema

Proces prilagodbe koji uključuje uvođenje podrške integracije više izvora podataka verificiran je na realnom primjeru. U svrhu davanja ocjene primjenjivosti

ovog procesa prilagodbe, web aplikacija "Nastavnički portal", koja predstavlja jedan od modula višegodišnjeg projekta "Informacijski sustav visokih učilišta", podvrgnuta je prilagodbi. Nastavnički portal je web aplikacija namijenjena nastavnicima. Putem nastavničkog portala nastavnici su u mogućnosti pratiti razne informacije o nastavi i ispitima, kao što su raspored nastave, različiti popisi studenata, raspored ispita te razni statistički pokazatelji. Kontinuirano praćenje nastave na Fakultetu elektrotehnike i računarstva provodi se na način da student tijekom semestra polaže više ispita, testova na računalu, obavlja domaće zadaće itd… Sve informacije vezane uz kontinuirano praćenje studenta prate se drugim informacijskim sustavom – web aplikacijom AHyCo. Motivacija ovog procesa prilagodbe je omogućiti nastavnicima da putem Nastavničkog portala imaju na jednom mjestu sve informacije vezane uz vlastiti kolegij. Dakle, potrebno je omogućiti da postojeća aplikacija "Nastavnički portal" uključi dodatan izvor podataka, onaj nad kojim radi aplikacija AHyCo u svrhu integriranja podataka o kontinuiranom praćenju.

7.5.4.2. Motivacija za odabir semantičkih tehnologija

AHyCo i Nastavnički portal su aplikacije rađene potpuno različitim tehnologijama, AHyCo je razvijan programskim jezikom C# te kao trajno spremište podataka koristi relacijsku bazu podataka MS SQL Server (ahycoDB). Nastavnički portal je Java web aplikacija, koja kao trajno spremište podataka koristi relacijsku bazu podataka Informix (isvuDB). U analizi mogućih metoda integracije, koje ne uključuju semantičke tehnologije, prepoznate su sljedeće mogućnosti:

- integracija na razini relacijskih baza podataka (dozvoliti pristup aplikacije Nastavnički portal relacijskoj bazi podataka ahycoDB)
- razmjena podataka servisima - nastavničkom portalu se na zahtjev dostavljaju potrebni podaci

Spomenute su samo metode koje omogućavaju integraciju podataka u trenutnom vremenu, dakle bez vremenskog kašnjenja. Kako detaljno objašnjenje svakog pristupa pojedinačno ne pripada području ovog istraživanja, navode se osnovne karakteristike pristupa te se stavlja naglasak na njihove nedostatke i usporedbu s realizacijom integracije uporabom semantičkih tehnologija. Tablica 11 prikazuje sumarno karakteristike svih navedenih pristupa.

Tablica 11. Usporedba različitih načina integriranja podataka

	integracija na razini relacijskih baza podataka	**razmjena podataka servisima**	**integracija primjenom semantičkih tehnologija**
pristup podacima (upitni jezik)	različite sintakse SQL-a; nemoguće postaviti jedinstven upit nad podacima iz obje relacijske baze podataka; nužno poznavanje oba modela podataka	poziv servisa	jedinstven pristup SPARQL upitnim jezikom; nužno poznavanje samo jednog zajedničkog modela podataka; moguće postaviti jedinstven upit nad oba izvora podataka
fleksibilnost	proizvoljan način dohvata podataka	ograničen skup podataka; promjene skupa koji se dohvaća samo na zahtjev	proizvoljan način dohvata podataka
sigurnost	moguće ograničenje dozvola nad podacima korištenjem mehanizma kontrole pristupa relacijske baze podataka	potpuno podržana	podržana uz implementaciju sustava za upravljanje identitetom predloženog u okviru ovog istraživanja
povezivanje (izjednačavanje) podataka	problem utvrđivanja zajedničkog značenja; problem različitih tipova podataka koji se integriraju	nužno objašnjenje značenja podataka koji su rezultat dohvata servisa; povezivanje se odvija u objektnom modelu	svi izvori su opisani semantičkim modelom, koji je moguće povezati u samo jedan model; moguće je koristiti tehnike traženja semantičke sličnosti podataka

Pristup razmjene podataka servisima zahtijeva najveći trud osoba koje su zadužene za razvoj sustava s kojim se vrši integracija. Također, radi se o pristupu koji je najteže održavati, jer svaka promjena relacijskog modela ili zahtjeva korisnika traži promjenu servisa, kao i ponovno utvrđivanje značenja između vlasnika i korisnika servisa.

Pristup u kojem se dozvoljava aplikaciji koja integrira podatke pristup relacijskoj bazi podataka fleksibilniji je, jer omogućava postavljanje proizvoljnih upita nad bazom podataka, tako da je lako utjecati na promjene skupa podataka koji se dohvaćaju. No, kako bi se upiti mogli ispravno obavljati potrebno je napredno znanje značenja relacijskog modela. Sigurnost ovog pristupa moguće je ostvariti ispravnom dodjelom dozvola nad relacijskom bazom podataka korištenjem mehanizama kontrole pristupa. Problem koji se ipak javlja kod ovog pristupa je što se na taj način može samo ograničiti skup podataka koji se dohvaća, ali ne i način na koji se to čini, što može prouzrokovati probleme u performansama sustava. Opisan slučaj u kojem se traži integracija podataka zahtijeva povezivanje podataka pohranjenih u dva različita sustava za upravljanje podacima (SQL Server i Informix) pa iako je aplikaciji omogućen pristup u obje baze podataka, ne mogu se postavljati upiti koji bi istovremeno dohvaćali podatke iz oba izvora podataka *(eng. federated query)*.

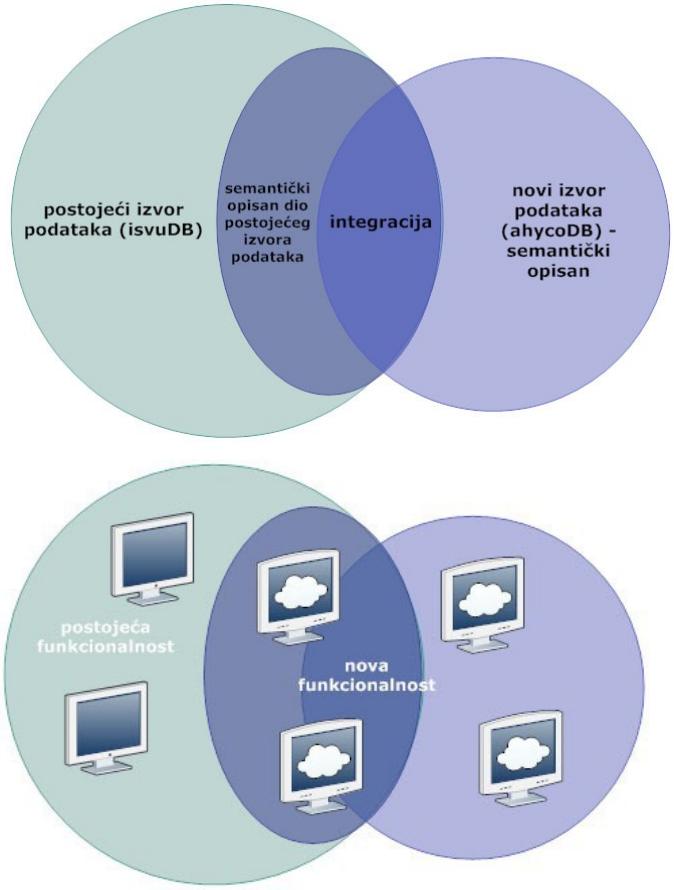
Integracija opisanih izvora podataka uporabom semantičkih tehnologija također zahtijeva intervenciju osoba koje su zadužene za održavanje relacijske baze podataka s kojom se želi postići integracija. Predlaže se da se provede prethodno opisan način semantičkog opisivanja podataka relacijskih baza podataka. Na taj način kreira se SPARQL pristupna točka putem koje sve udaljene aplikacije mogu dohvaćati podatke iz relacijske baze podataka. Podaci su opisani ontologijom koja opisuje značenje podataka sadržanih u relacijskoj bazi podataka. Kako je podacima pridijeljeno značenje, svi problemi koji se inače vezuju uz klasičan način obavljanja integracije (npr. strukturna, sintaktička i semantička heterogenost) ovim pristupom su eliminirani. Također, semantičke tehnologije omogućavaju postavljanje SPARQL upita nad svim integriranim izvorima istovremeno.

7.5.4.3. Tijek procesa prilagodbe

Semantički opis podataka

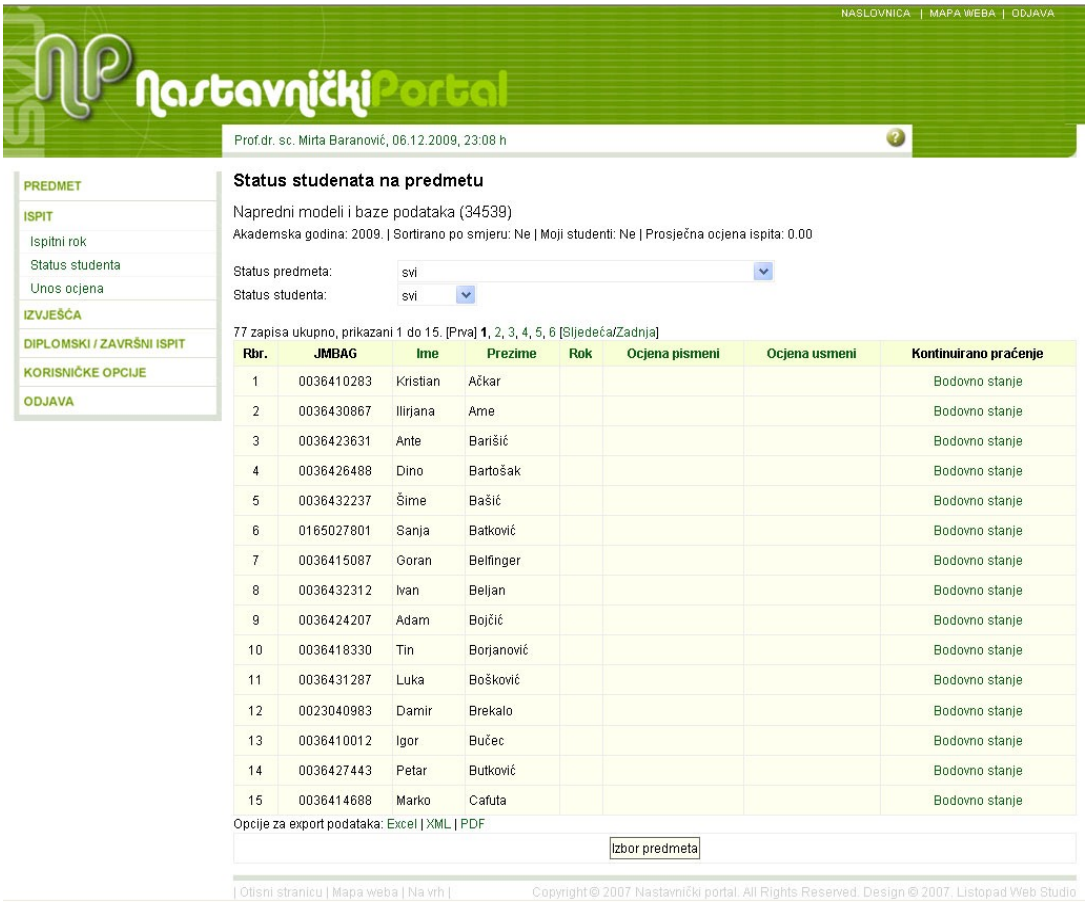
Izvor podataka koji je potrebno integrirati, dakle izvor s podacima o kontinuiranom praćenju studenata dostupan je putem SPARQL pristupne točke. Kako je nad procesom prilagodbe postavljen uvjet da se poduzimaju samo nužni minimalni koraci za ostvarenje integracije podataka samo je dio podataka postojeće relacijske baze podataka koju koristi Nastavnički portal semantički opisan. Semantički opis uvodi se za onaj dio relacijske baze koji je potrebno integrirati s novim izvorom podataka. Ostali dijelovi aplikacije ostaju nepromijenjeni tako da i dalje djeluju nad postojećim pristupnim slojem izvoru podataka (slika 71). Uvođenje

semantičkog opisa relacijske baze podataka ostvareno je alatom D2R Server, a razlozi takvog odabira objašnjeni su kod prijedloga procesa prilagodbe u poglavlju 7.5.1.1.



Slika 71 Integracija izvora podataka

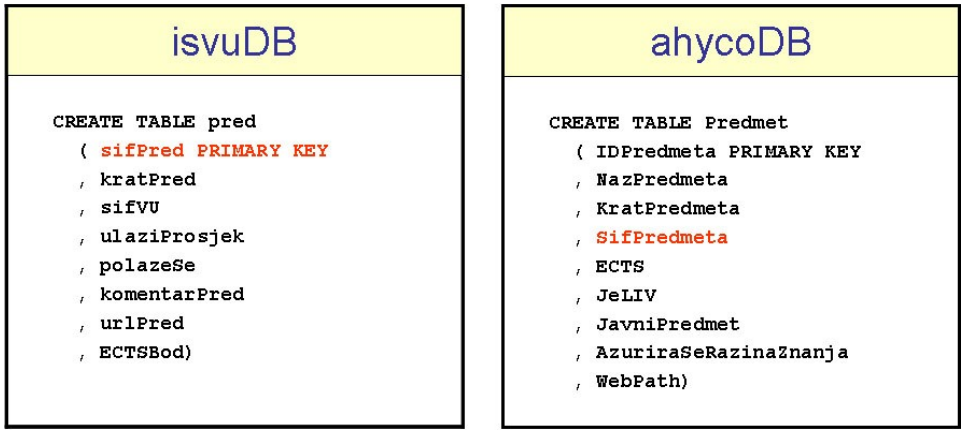
Specifičnost integracije izvora podataka potrebnih za ovaj promatrani slučaj je što se nije morala kreirati "ontologija preslikavanja". Naime, oba izvora podataka preslikana su korištenjem iste ontologije što je eliminiralo potrebu za definiranjem poveznica između shema. Što se izjednačavanja instanci tiče, kod odabira načina realizacije koristilo se iskustvo stečeno izradom semantičkog web portala - Sweb. Dakle, priroda izvora podataka koji se integriraju i funkcionalnosti koja se implementira nad njima nalaže da se podaci moraju potpuno točno povezati. Tijekom izrade Sweba pokazalo se da se u tom slučaju moraju uvesti globalni URI-i. Na primjer, za prikaz rezultata kontinuiranog praćenja studenta na odabranom predmetu proširuje se funkcionalnost već postojećeg prikaza „Status studenata na predmetu“ (slika 72). Odabirom opcije „Bodovno stanje“ za pojedinog studenta prikazuju se rezultati njegovog kontinuiranog praćenja.



Slika 72 Postojeća funkcionalnost aplikacije - status studenata na predmetu

Kako bi se prikazali potpuno točni rezultati studenta na odabranom predmetu, očito je da je potrebno izjednačiti instance *student*, *predmet* i *akademska godina* iz oba izvora podataka. Akademsku godinu je lako izjednačiti neovisno o URI-u koji joj je dodijeljen, jer se radi o resursu koji ima jednoznačna svojstva. Predmet i student ipak treba izjednačiti po URI-u koji označava te resurse, jer bilo kakva druga metoda ne garantira točnost. Prema pravilima definiranim prilikom definiranja arhitekture semantičkog web portala, uvodi se globalni identificirajući URI, te ga zadaje onaj sustav koji je vlasnik tih podataka. U promatranom slučaju vlasnik je ISVU sustav, pa su se pri preslikavanju ahycoDB relacijske baze za studente i predmete definirali URI-i isti onima definiranim pri preslikavanju podataka isvuDB relacijske baze u RDF. Taj zadatak je bilo lako realizirati budući da atributi po kojima se tvori URI su prisutni u obje relacijske baze. Primjer atributa relacije predmet kod jedne i druge relacijske baze prikazuje slika 73. Iako atribut koji sadrži šifru predmeta (sifPredmeta) nije primaran ključ u relaciji predmet ahycoDB sustava, jednak je šifri predmeta u ISVU sustavu te ga stoga jednoznačno označava. Prilikom preslikavanja

ahycoDB baze podataka atribut SifPredmeta koristi se za generiranje URI-a instanci predmeta koji su jednaki onima nastalim prilikom preslikavanja ISVU baze (slika 74).



Slika 73 Relacija predmet u relacijskim bazama koje se integriraju

```
map:PredmetMap a d2rq:ClassMap;
   d2rq:uriPattern"http://www.isvu.hr/nastavnici/course/@@pred.sifpred@@";
   d2rq:class univ-ont:Course;
   d2rq:dataStorage map:Database1;
   .
```

Slika 74 Isječak iz definicije preslikavanja ahycoDB u RDF

Relacije student u oba sustava kao primaran ključ imaju atribut JMBAG, tako da je generiranje identificirajućih URI-a za studente jednoznačno. Iako se za izjednačavanje instanci studenta moglo primijeniti traženje uzorka, dakle dozvoliti autonomiju izvora i objaviti ISVU URI u obliku *http://www.isvu.hr/nastavnici/student/0036368145* i AHyCo URI u obliku *http://ahyco.fer.hr/student/0036368145* pa izjednačiti po JMBAG-u koji je dio oba URI-a i koji je za svakog studenta jedinstven, takvo rješenje se smatra lošijim. Ostvariti izjednačavanje na opisan način zahtijeva ugradnju poslovne logike, koja bi posebno rukovala entitetima za koje se eksplicitno mora navesti da podliježu navedenom pravilu. Druga moguća varijanta je uvođenje poveznica između instanci ontologijom preslikavanja koja bi sadržavala trojke kao na primjer: *http://www.isvu.hr/nastavnici/student/0036368145 owl:sameAs http://ahyco.fer.hr/student/0036368145*. Ontologiju preslikavanja za opisan slučaj jako je teško održavati, jer bi svaki unos studenta u relacijskoj bazi morao rezultirati trojkom u ontologiji preslikavanja. Također, uvođenje ontologije preslikavanja zahtijeva da se i podaci pohranjuju u lokalno semantičko spremište podataka kako

bi se moglo izvesti zaključivanje nad definiranim poveznicama više izvora podataka. U promatranom slučaju, takav pristup nije isplativ, jer nema nužne potrebe za izgradnjom ontologije preslikavanja, a i priroda funkcionalnosti koja se ugrađuje u postojeću aplikaciju ne zahtijeva primjenu zaključivanja, pa nema ni potrebe za razvojem sloja koji bi podržavao zaključivanje nad RDF grafovima.

Pristupni sloj i pretvorba RDF podataka u objekte

Osim pristupnog sloja relacijskoj bazi podataka, koji je već dio aplikacije, dodatno je kreiran pristupni sloj semantičkim podacima. Tim slojem je omogućena komunikacija korištenjem SPROT protokola, budući da su podaci relacijskih baza podataka korištenjem D2R Servera dostupni putem SPARQL krajnjih točki. Na temelju analize zahtjeva za proširenjem funkcionalnosti postojeće aplikacije i ontologije domene izvora koji se dodatno integriraju proširen je objektni model aplikacije na prethodno predložen način (poglavlje 7.5.2). Za pretvorbu dohvaćenih semantičkih podataka u objekte korištena je ista razvojna paradigma kao ona već prisutna za pretvaranje podataka relacijske baze u objekte. Pristupni sloj i pretvorba podataka realizirana je korištenjem programskog okvira - Jena.

Prilagodba objektnog modela

Kako se radi o prilagodbi aplikacije, dodatno je potrebno povezati integrirane semantičke izvore s onim već postojećim, dakle potrebno je uvesti vezu između postojećeg objektnog modela aplikacije i podataka iz dodanih izvora podataka. Odnosno, potrebno je izjednačiti instance dobivene iz relacijske baze podataka putem SQL upita s podacima dobivenim iz semantičkih izvora podataka. Prema naputku iz predložene metodologije procesa prilagodbe, sve klase koje predstavljaju entitete čija se svojstva barem dijelom nalaze u modelima semantičkih izvora nasljeđuju klasu RDFObject. Na taj način svaka instanca te klase sadrži svojstvo URI čiji je sadržaj potrebno generirati aplikacijom na isti način na koji se on generira pri definiranju preslikavanja podataka relacijske baze u semantičke. Primjer proširene klase prikazuje slika 75 - crvenom bojom su podcrtani dodani dijelovi. Pri instanciranju objekta PredmetData te dodjeljivanjem vrijednosti prikupljenih iz relacijske baze podataka tom objektu, automatski se kreira i sadržaj svojstva URI za taj objekt. Tako generiran URI predstavlja vezu sa semantičkim izvorima, odnosno koristi ga se kao poveznicu pri postavljanju SPARQL upita. Dakle, odabirom studenta na stranici koju prikazuje slika 72 poznati su parametri za obavljanje upita

(predmet, student i akademska godina) u svrhu dohvata rezultata kontinuiranog praćenja studenta na predmetu. Primjer takvog SPARQL upita prikazuje slika 76.

```
public class PredmetData extends RDFObject implements Serializable {

   private Integer sifPred;
   private String kratPred;
   private Integer sifVU;
   private String ulaziProsjek;
   private String polazeSe;
   private String komentarPred;
   private String urlPred;
   private Double ectsBod;
   private Integer sifNajUzaOrgJed;
   private String defaultniNaziv;
   private OrgJedData orgJed;

   public PredmetData() {
   }

   public PredmetData(  Integer sifPred,
                        String kratPred,
                        String defaultniNaziv,
                        Integer sifVU,
                        String ulaziProsjek,
                        String polazeSe,
                        String komentarPred,
                        String urlPred,
                        Double ectsBod,
                        Integer sifNajUzaOrgJed ) {

      this.sifPred = sifPred;
      this.kratPred = kratPred;
      this.sifVU = sifVU;
      this.ulaziProsjek = ulaziProsjek;
      this.polazeSe = polazeSe;
      this.komentarPred = komentarPred;
      this.urlPred = urlPred;
      this.ectsBod = ectsBod;
      this.sifNajUzaOrgJed = sifNajUzaOrgJed;
      this.defaultniNaziv = defaultniNaziv;
      this.URI = GenerateURI.generate("course", String.valueOf(sifPred));

   }
}
```

Slika 75 Primjer klase koja nasljeđuje klasu RDFObject

```
ResultSet rs = ServiceAccess.makeServiceQuery(
        "PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> " +
        "PREFIX foaf: <http://xmlns.com/foaf/0.1/> " +
        "PREFIX dc: <http://purl.org/dc/elements/1.1/> " +
        "PREFIX vocab: <http://ahyco.fer.hr/vocab/> " +
        "PREFIX sisvu: <http://www.fer.hr/projekt#> " +
        "SELECT ?provjeraStudenta ?nazivProvjere ?maxBodova ?ostvarenaTezina ?osoba ?ime ?prezime ?akstupanj " +
        " WHERE { " +
        "   ?provjera           vocab:pripada_predmet   " + predmet.getURI()  +" ." +
        "   ?provjeraStudenta   vocab:pripada_provjera  ?provjera  ." +
        "   ?provjeraStudenta   vocab:pristupnik        " + student.getURI() + " ." +
        "   ?provjera           foaf:name               ?nazivProvjere ." +
        "   ?provjera           dc:description          ?opisProvjere ." +
        "   ?provjera           vocab:maxbodova         ?maxBodova ." +
        "   ?provjera           vocab:akademskaGodina   ?akgodina " +
        "   ?akgodina           vocab:godina            " + akgod + "." +
        "   ?provjeraStudenta   vocab:ostvarenaTezina   ?ostvarenaTezina ." +
        "   ?provjeraStudenta   vocab:ispravljac        ?osoba . " +
            "   SERVICE <http://161.53.18.121:2020/sparql> " +
            "    { ?osoba           foaf:firstName      ?ime . " +
            "      ?osoba           foaf:surname        ?prezime . " +
            "      OPTIONAL { " +
            "       ?osoba  sisvu:abbreviation  ?akstupanj } . " +
            "      } " +
        " }");
```

isvuDB

ahycoDB pristupna točka

isvuDB pristupna točka

Slika 76 SPARQL upit nad dva izvora podataka

Slika 76 prikazuje upit koji se izvodi istovremeno na više semantičkih izvora podataka. Parametri po kojima se vrši pretraga su podaci dohvaćeni iz relacijske baze podataka, podaci o provjerama kontinuiranog praćenja dohvaćaju su iz ahycoDB-a, a podaci o osobama koje su ispravljale te provjere iz isvuDB-a.

Naposljetku slijedi i izrada novih web stranica koje će prikazivati podatke prikupljene iz drugih izvora podataka. Prikaz podataka na temelju dohvaćenih podataka upitom koji prikazuje slika 76, prikazuje slika 77.

Slika 77 Prikaz nove funkcionalnosti

7.5.5. Ocjena primjenjivosti

Aplikacija odabrana za verifikaciju predložene metodologije procesa prilagodbe poslovna je aplikacija, pa se temelji na podacima koje nije poželjno javno objavljivati. Također, takve aplikacije nalažu da se mora podržati potpuna točnost integracije podataka, kao i zaštita podataka. Dodatno, domena aplikacije promatranog slučaja je takva da ne zahtijeva implementaciju zaključivanja ili primjena metrika sličnosti. No, kako se radi o metodama koje pripadaju inženjerstvu semantičkih web aplikacija smatra se da su već uhodane i ne očekuje se da njihova primjena u procesu prilagodbe predstavlja problem.

Proces prilagodbe bilo je u potpunosti moguće izvesti slijedeći korake predložene metodologije. Kao najzahtjevniji koraci ocijenjeni su inženjerstvo ontologije i definiranje preslikavanja podataka relacijske baze u RDF podatke. Radi se o koracima koji zahtijevaju napredno poznavanje jezika ontologija i domene nad kojom je ona izgrađena. Ostali koraci srodni su inženjerstvu klasičnih aplikacija te je stoga i lakše njihovo provođenje.

U promatranom slučaju ocjenjuje se da je značajna korist postignuta ostvarenom integracijom u odnosu na druge opcije koje su razmatrane kao moguće rješenje postavljenog problema (poglavlje 7.5.4.2). Ostvaren je vrlo fleksibilan način integracije na koji promjena potreba za skupom podataka koji se dohvaća nema nikakav utjecaj. Također, izmjena strukture relacijske baze podataka, može biti potpuno nevidljiva za aplikaciju koja takav izvor integrira uslijed korištenja medijatora. Najveća prednost je semantički opis izvora podataka kojim se eliminiraju svi problemi nejednoznačnosti prilikom integracije podataka. Kao nedostatak ciljne aplikacije ističe se teže održavanje uslijed povećanja opsega znanja kojima razvojni tim mora raspolagati, dakle nužno je uključenje eksperata ontologija. Trenutno stanje razvoja semantičkih tehnologija ne dopušta izmjenu semantičkih podataka, već samo njihovo čitanje. Tako da svi podaci koji su dohvaćeni iz drugih izvora podataka, koji nisu u vlasništvu aplikacije koja se prilagođava, služe samo za pregled, odnosno prikaz informacija. Na taj način je olakšano i kreiranje ontologije, budući da nije nužno podržati 'zatvaranje svijeta', jer se podaci ne unose u semantičkom formatu. Ostala funkcionalnost aplikacije, dakle izmjena i unos podataka, obavlja se na isti način kao i u postojećoj aplikaciji, pa performanse sustava u tom pogledu ostaju sasvim iste.

Za sve slučajeve u kojima je potrebno uvesti integraciju više heterogenih izvora podataka, samo u svrhu dohvata i ostvarivanja zajedničkog pogleda na podatke, a ne i u svrhu izmjene ili implementacije bilo kakve kompliciranije poslovne logike nad takvim podacima, ovom se procesu prilagodbe daje visoka ocjena primjenjivosti.

7.6. Poboljšavanje/uvođenje funkcionalnosti pretraživanja

Motivacija za poduzimanje ovog procesa prilagodbe je ugradnja funkcionalnosti pretraživanja u postojeću aplikaciju. Funkcionalnost pretraživanja realizirana semantičkim tehnologijama obećava bolje rezultate u odnosu na realizaciju klasičnim tehnologija. Analizom semantičkih web aplikacija pokazano je da je teško poopćiti implementaciju funkcionalnosti pretraživanja i odvojiti je od domene na koju se primjenjuje. Procesom prilagodbe koji se ovdje predlaže obuhvaćaju se svi koraci kojima se aplikacija priprema za implementaciju algoritma pretraživanja semantičkim tehnologijama dok su sami detalji implementacije algoritma izostavljeni.

Analizom semantičkih web aplikacija utvrđeno je da postoje različiti pristupi realizacija semantičkog pretraživanja, od kojih se za proces prilagodbe odabire onaj

za koji se smatra da ga je lakše integrirati u postojeći sustav. Postoje dva osnovna načina implementacije pretraživanja korištenjem semantičkih tehnologija:

- facetno pretraživanje
- pretraživanje pomoću ključnih riječi (oznaka) potpomognuto ontologijom

Facetno pretraživanje objašnjeno je u poglavlju 6.3. Teško ga je primijeniti nad postojećim aplikacijama, jer se zahtijeva značajna promjena *izgled* i *model* dijela MVC obrasca. Uvođenje podrške za facetno pretraživanje u izgled mijenja koncept prikaza web stranica aplikacije pa se procjenjuje da takva vrsta prilagodbe nije isplativa.

Pretraživanje pomoću ključnih riječi (oznaka) potpomognuto ontologijom smatra se lakše ostvarivim u već postojećem aplikacijama. Za njegovu realizaciju u postojećem aplikacijama potrebno je osigurati sljedeće:

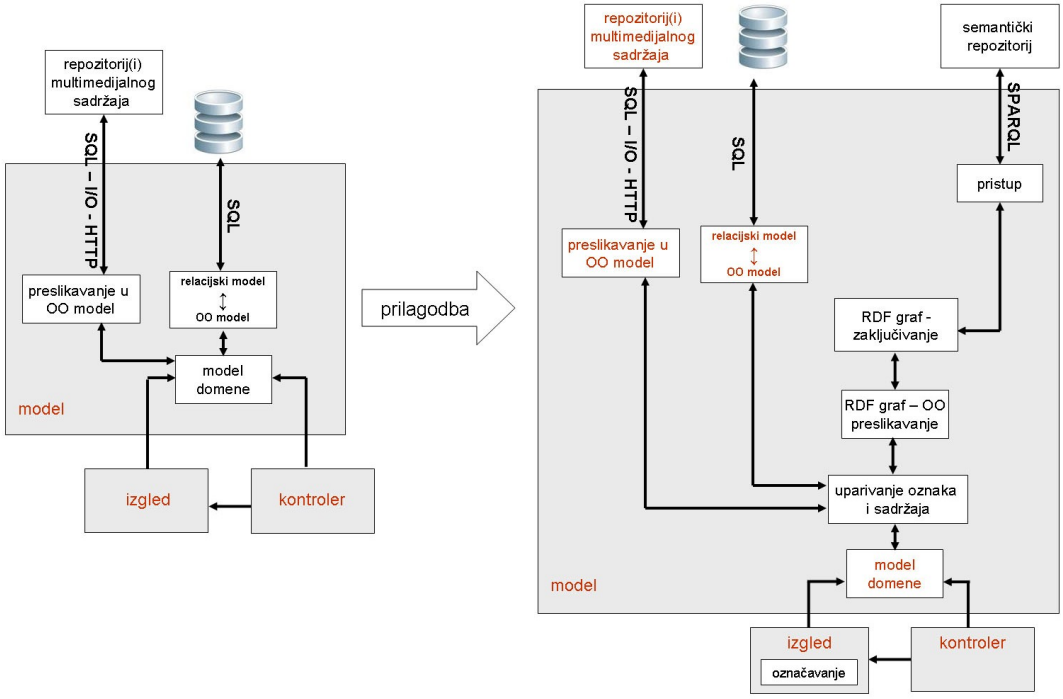
- semantički repozitorij
- mehanizam zaključivanja
- vezu između sadržaja aplikacije i semantičkih oznaka

S ciljem da se ostvari bilo kakav oblik semantičkog pretraživanja potrebno je osigurati pohranjivanje značenja sadržaja, odnosno u postojeće rješenje integrirati semantički repozitorij koji bi sadržavao semantičke opise sadržaja. Nad takvim repozitorijem tada je moguće izvoditi upite, koji bi dodatno uz pomoć mehanizma za zaključivanje davali semantički točnije/bliske odgovore od onog što klasični pretraživači mogu ponuditi. Razlikuju se dva načina označavanja sadržaja: ručno ili automatsko. Ovdje predložen proces prilagodbe vrijedi za ručno označavanje sadržaja koje obavlja korisnik.

Semantičko označavanje sadržaja podržano je u semantičkim *sustavima za upravljanje znanjem*, pa se za definiranje ciljne arhitekture ovog procesa prilagodbe preuzimaju elementi arhitekture tih sustava (slika 58). Tijekom analize semantičkih web aplikacija funkcionalnost pretraživanja je prepoznata i kao učestala funkcionalnost semantičkih web portala. No, kod semantičkih web portala uglavnom se radilo o vizualnim (facetnim) pretraživanjima ili su se oznake sadržaja pohranjivale lokalno u semantički repozitorij, što se opet podudara s arhitekturom sustava za upravljanje znanjem.

Specifičnost ovog procesa prilagodbe je pretpostavka da se češće za prilagodbu razmatraju klasične web aplikacije, koje osim relacijske baze sadrže i druge repozitorije za multimedijalan sadržaj. Tako da se upravo arhitektura takvih aplikacija uzima kao arhitektura aplikacija koje se razmatraju za prilagodbu. Slika 78

prikazuje početnu i ciljnu arhitekturu ovog procesa prilagodbe. Ciljna arhitektura osim elemenata sustava za upravljanje znanjem sadrži i relacijsku bazu podataka kojoj je zadržan klasičan SQL pristup budući da se prilagodbom način same pohrane sadržaja ne mijenja, već se samo pruža mogućnost semantičkog označavanja takvog sadržaja.



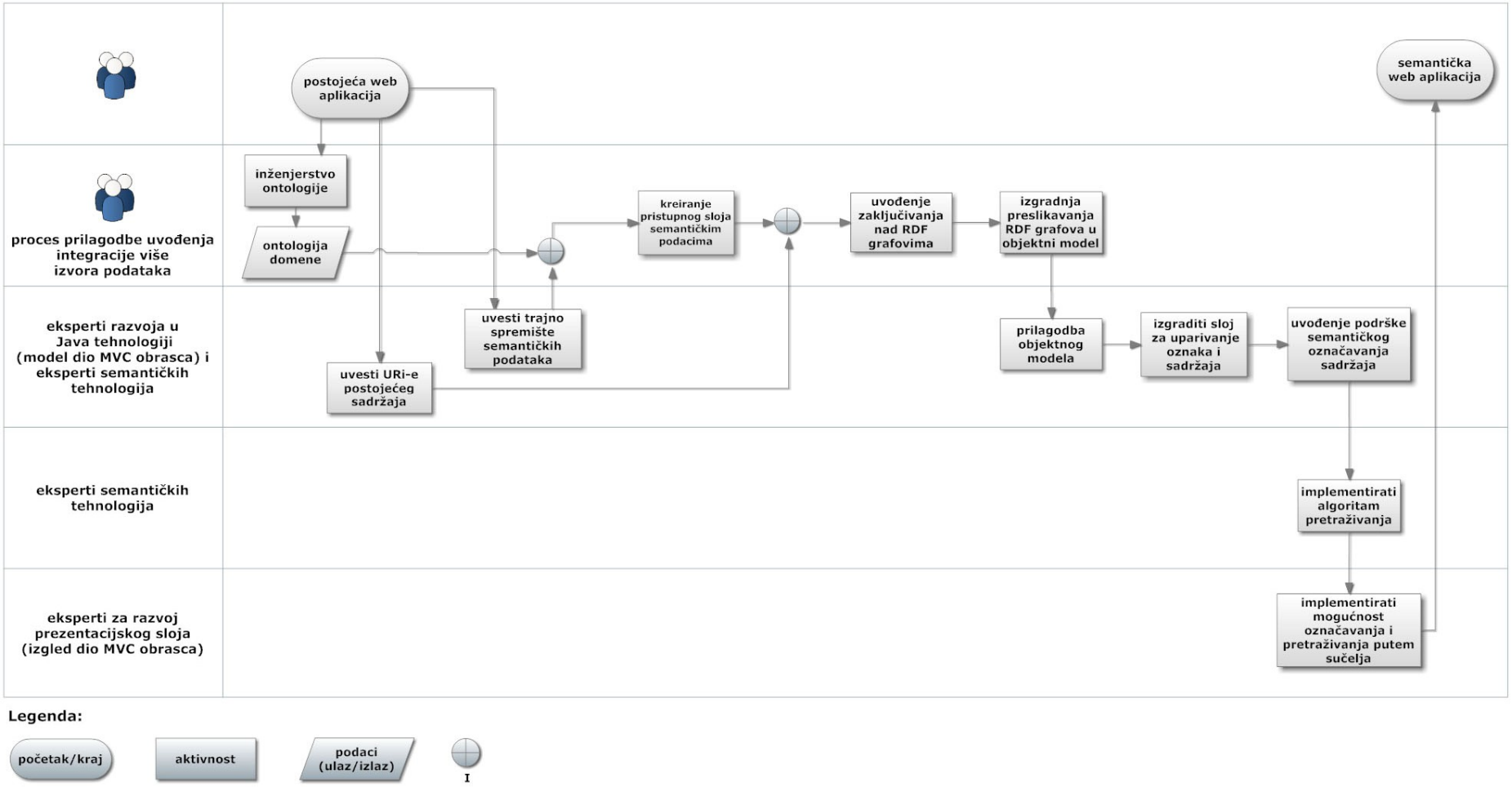
Slika 78 Polazna i ciljna arhitektura procesa prilagodbe

7.6.1. Metodologija prilagodbe

Primjena predložene metodologije prilagodbe podrazumijeva poduzimanje aktivnosti koje prikazuje tablica 12. U većoj mjeri se radi o aktivnostima koje su već poznate iz inženjerstva semantičkih web aplikacija, kao i iz procesa prilagodbe uvođenja podrške integracije više izvora podataka. Dodatne aktivnosti koje se moraju poduzeti su implementacija algoritma prema kojem će se obavljati pretraživanje te uvođenje podrške za označavanje sadržaja. Za svaku aktivnost koja se poduzima tijekom prilagodbe navedeni su njeni preduvjeti i pripadnost predloženom postupku. Također, koje bi eksperte trebalo uključiti za obavljanje pojedinih koraka procesa prilagodbe prikazuje slika 79.

Tablica 12. Metodologija prilagodbe uvođenja funkcionalnosti pretraživanja

broj aktivnosti	preduvjeti	opis aktivnosti	pripadnost aktivnosti prethodno predloženom postupku
1	nema	inženjerstvo ontologije – pronaći dostupne ontologije koje pokrivaju sadašnji model podataka, izgraditi dodatne ontologije i međusobno ih povezati	inženjerstvo semantičkih web aplikacija proces prilagodbe integracije više izvora podataka
2	nema	uvesti URI-e postojećeg sadržaja (voditi računa o razrješavanju URI-a)	proces prilagodbe integracije više izvora podataka
3	nema	uvesti trajno spremište semantičkih podataka	inženjerstvo semantičkih web aplikacija
4	1, 3	kreirati pristupni sloj semantičkim podacima	inženjerstvo semantičkih web aplikacija proces prilagodbe integracije više izvora podataka
5	2, 4	omogućiti zaključivanje / izračun sličnosti nad dobivenim RDF grafovima	inženjerstvo semantičkih web aplikacija proces prilagodbe integracije više izvora podataka
6	5	izgraditi sloj preslikavanja RDF grafova u objektni model	inženjerstvo semantičkih web aplikacija proces prilagodbe integracije više izvora podataka
7	6	dodatno prilagoditi objektni model (udruživanje objekata)	proces prilagodbe integracije više izvora podataka
8	7	izgraditi sloj za uparivanje oznaka i sadržaja (dodatna prilagodba objektnog modela)	proces prilagodbe uvođenja funkcionalnosti pretraživanja
9	8	uvođenje podrške semantičkog označavanja sadržaja	inženjerstvo semantičkih web aplikacija
10	9	implementirati algoritam pretraživanja (metrike sličnosti, kategorizacija, itd…)	proces prilagodbe uvođenja funkcionalnosti pretraživanja
11	10	implementirati mogućnost označavanja i pretraživanja putem sučelja	proces prilagodbe uvođenja funkcionalnosti pretraživanja



Slika 79 Tijek procesa prilagodbe uvođenja funkcionalnosti pretraživanja u postojeće aplikacije po zaduženjima

7.6.1.1. Promjene MVC obrasca - model

Većina aktivnosti koje je potrebno poduzeti u model dijelu MVC obrasca poznate su iz inženjerstva semantičkih web aplikacija ili iz prethodno predloženog procesa prilagodbe (poglavlje 7.5) pa se ovdje neće detaljno razmatrati već se daje samo kratak osvrt na njih.

U svrhu pohrane semantičkih opisa sadržaja, koji se koriste pri pretraživanju, u model se uvodi semantički repozitorij. Također, potrebno je izgraditi ontologiju pomoću koje se može semantički označavati sadržaj aplikacije. Budući se u postojeću aplikaciju uvodi nova vrsta trajnog spremišta - semantički repozitorij, potrebno je osigurati i novu vrstu pristupnog sloja, odnosno podršku za SPARQL upite. Kako bi se mogli uvesti semantički opisi postojećeg sadržaja potrebno je sadržaju dodijeliti identificirajuće URI-e. Dakle, URI dokumenta, teksta, slike, itd… Realizacija semantičkog označavanja obavlja se na isti način kao i kod sustava za upravljanje znanjem – implementacijom MOAT-a (poglavlje 6.4.1).

Samu izradu algoritma pretraživanja teško je sugerirati u potpunosti budući da je specifična za pojedinu problematiku aplikacije, ali se može predvidjeti da se zasniva na metrikama sličnosti i zaključivanju, koje je moguće provesti ukoliko se uspješno obave sve prethodno navedene aktivnosti.

7.6.1.2. Promjene MVC obrasca - izgled

Potrebno je u web (JSP) stranice uvesti podršku za označavanje sadržaja. Koristi li se predložena MOAT arhitektura, radi se o implementaciji MOAT klijenta (poglavlje 4.3). Također, potrebno je implementirati web sučelje putem kojeg će se pretraživanje ostvarivati.

7.6.2. Ocjena primjenjivosti

Kako bi se mogla dati kvalitetna ocjena primjenjivosti procesa prilagodbe potrebno je znati korist i trošak koju njegova provedba donosi. Studije koje bi pokazale korist predloženog semantičkog pretraživanja (korištenjem oznaka) još uvijek nisu dostupne, jer se radi o novijem pristupu. Nedavno objavljeno istraživanje [Bischoff2008] iznosi zaključke o tome koje kategorije oznaka uvesti i kako ih koristiti pri označavanju pojedine vrste multimedijalnog sadržaja. No, radi se o istraživanju koje se odnosi na cijeli web pa opet ostaje nepoznato kakav bi utjecaj predloženo pretraživanje imalo u zatvorenim sredinama, za koje su sustavi za upravljanje znanjem predloženi u ovom istraživanju i namijenjeni. Provođenje vlastite analize

zahtijeva bogatu ontologiju, velik skup podataka i korisnika što se tijekom ovog istraživanja nije moglo podržati. Zbog navedenih razloga ne može se dati odgovor na pitanje koliku korist donosi ugradnja funkcionalnosti pretraživanja. No, za razliku od koristi predloženog procesa prilagodbe, trošak je lakše ocijeniti. Prema iskustvu stečenom kod verifikacije procesa prilagodbe uvođenja integracije, kreiranje ontologije i izrada samog algoritma pretraživanja smatraju se najzahtjevnijim aktivnostima, pri čemu velik problem predstavlja razvoj prikladnog algoritma pretraživanja, budući da ne postoje rezultati istraživanja koji bi se mogli iskoristiti kao smjernice za implementaciju. Navedene aktivnosti također zahtijevaju da se u proces prilagodbe uključe eksperti ontologija.

Ostali koraci kao što su implementacija pristupnog sloja semantičkim podacima, dodjeljivanje URI-a postojećem sadržaju, implementiranje zaključivanja, prilagodba objektnog modela, kao i preslikavanja RDF grafova u OO model objašnjeni su kod procesa prilagodbe uvođenja podrške integracije više izvora podataka i već je dana ocjena njihove primjenjivosti (poglavlje 7.5.4). Tijekom izrade Sweba kao posljednja funkcionalnost implementirano je označavanje sadržaja. Dakle, radi se o situaciji koja bi se u ovom procesu prilagodbe zatekla prije obavljanja aktivnosti „implementirati algoritam pretraživanja“ pa se može dati adekvatna ocjena kompleksnosti aktivnosti. Označavanje je realizirano primjenom MOAT ontologije i arhitekture. Implementacija se sastojala od promjena web stranica na kojima je bilo potrebno uvesti mogućnost dinamičkog označavanja sadržaja kao i uvođenja podrške za pohranu oznaka u semantički repozitorij. Budući da se radi o već uhodanim rješenjima, a i postoje programski okviri (MOAT Java klijent i poslužitelj[41]) koji su rezultat ovog istraživanja i mogu se koristiti pri implementaciji, ocjenjuju se da je aktivnost lako provesti.

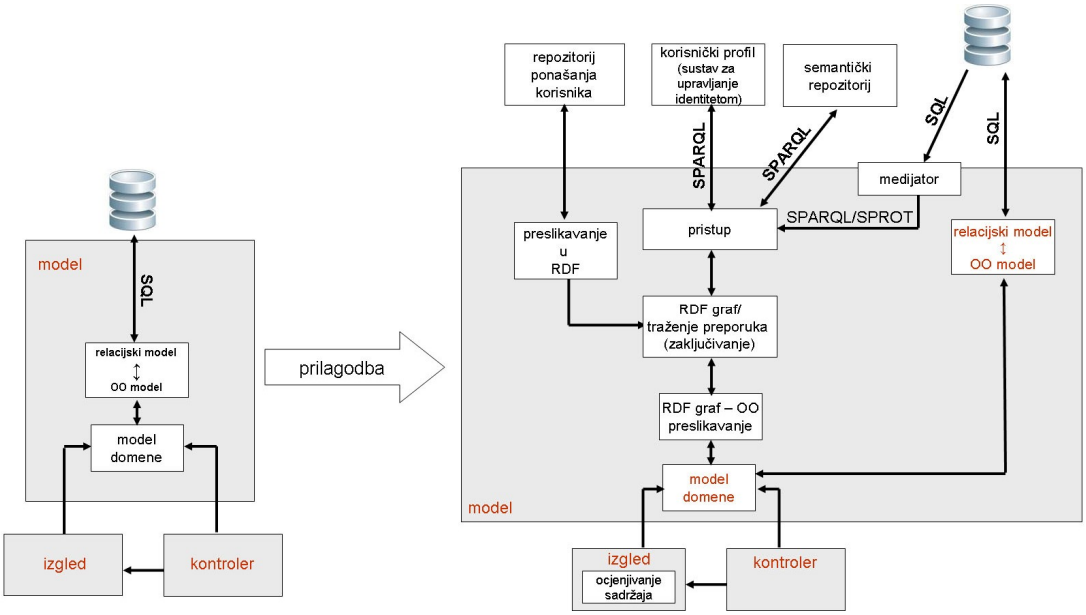
Iako se procjenjuje da trošak izmjena koje se moraju poduzeti nije velik, nemogućnost da se a priori ocijeni korist koju ovaj proces prilagodbe donosi sprečava mogućnost pružanja potpune ocjene primjenjivosti.

[41] http://moat.student.hr

7.7. Poboljšavanje/uvođenje funkcionalnosti personalizacije i preporučenog sadržaja

Prilikom poopćenja personalizacije objašnjeno je da ona podrazumijeva više od same preporuke sadržaja. Nemoguće je pokriti sve aspekte personalizacije pa je proces prilagodbe koji se ovdje predlaže usmjeren samo na problem uvođenja preporuke sadržaja kao najkompliciranije funkcionalnosti personalizacije.

Kako bi se u postojeću aplikaciju mogao uvesti izračun preporučenog sadržaja, kao jednog vida personalizacije, potrebno je u postojeći sustav integrirati elemente predložene arhitekture *semantičke web aplikacije preporučenog sadržaja* (slika 60). Slika 80 ilustrira početnu i ciljnu arhitekturu ovog procesa prilagodbe. Ciljna arhitektura sadrži sve elemente predložene semantičke web aplikacije preporučenog sadržaja s razlikom da se iz početne arhitekture procesa prilagodbe zadržava relacijska baza podataka, kojoj se omogućava pristup semantičkim tehnologijama uz zadržavanje i postojećeg pristupa (SQL) relacijskoj bazi podataka.



Slika 80 Polazna i ciljna arhitektura procesa prilagodbe

7.7.1. Metodologija prilagodbe

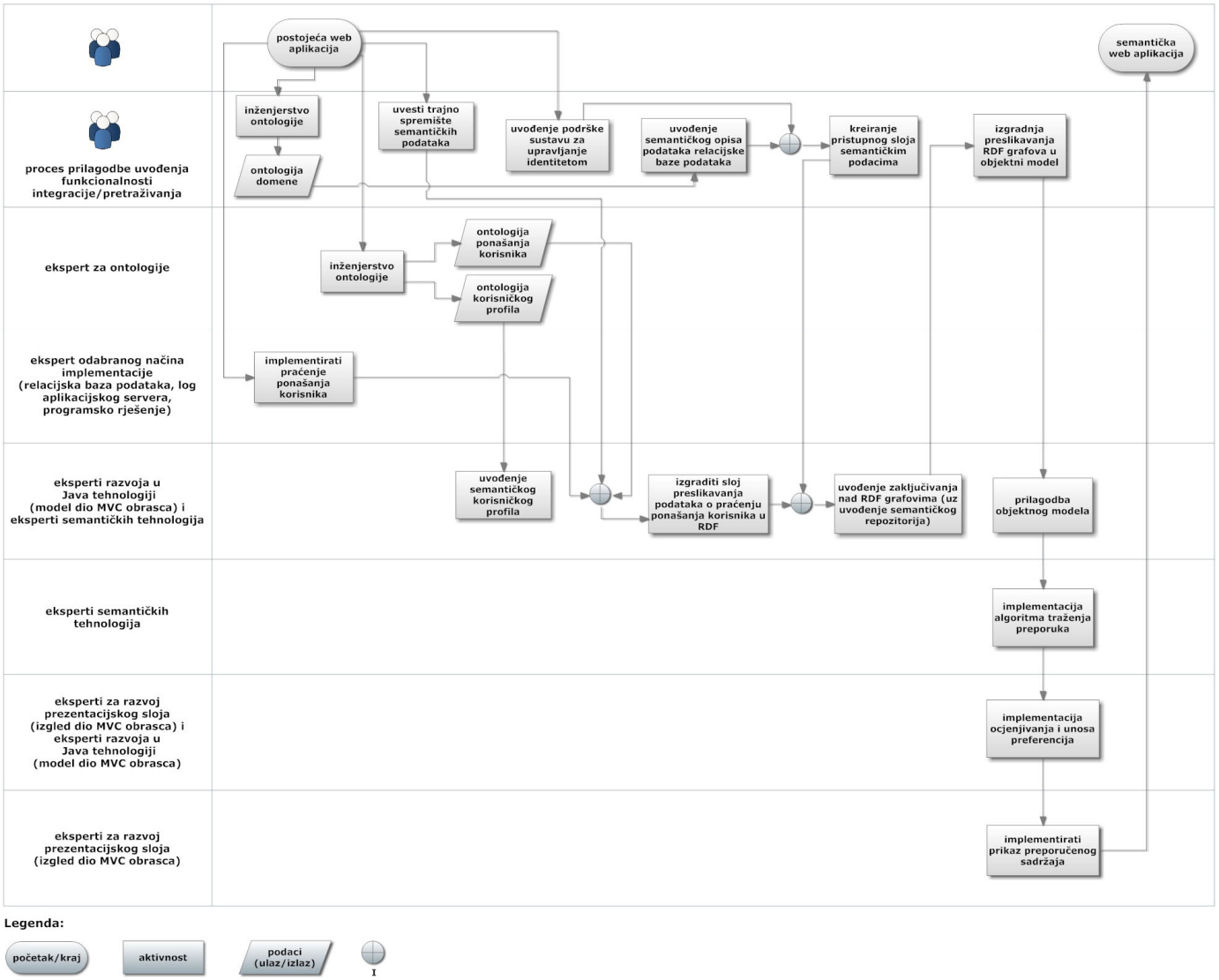
Uvođenje funkcionalnosti preporučenog sadržaja u postojeću aplikaciju zahtijeva poduzimanje aktivnosti koje su već poznate iz inženjerstva semantičkih web aplikacija, kao i iz procesa prilagodbe uvođenja podrške integracije više izvora

podataka (poglavlje 7.5). Dodatne aktivnosti koje se moraju poduzeti su obogatiti sučelje modulima za ocjenjivanje sadržaja, kao i izrada novih stranica na kojima se prikazuje preporučen sadržaj. Primjena predloženog procesa prilagodbe podrazumijeva poduzimanje aktivnosti koje prikazuje tablica 13. Za svaku aktivnost koja se poduzima tijekom prilagodbe navedeni su njeni preduvjeti i pripadnost predloženom postupku. Također, koje bi eksperte trebalo uključiti za obavljanje pojedinih koraka procesa prilagodbe prikazuje slika 81.

Tablica 13. Metodologija prilagodbe uvođenja preporučenog sadržaja

broj aktivnosti	preduvjeti	opis aktivnosti	pripadnost aktivnosti prethodno predloženom postupku
1	nema	inženjerstvo ontologije – pronaći dostupne ontologije koje pokrivaju sadašnji model podataka, izgraditi dodatne ontologije i međusobno ih povezati	inženjerstvo semantičkih web aplikacija proces prilagodbe integracije više izvora podataka
2	nema	inženjerstvo ontologije – kreirati ontologiju ponašanja korisnika	inženjerstvo semantičkih web aplikacija proces prilagodbe uvođenja personalizacije i preporučenog sadržaja
3	nema	inženjerstvo ontologije – kreirati ontologiju korisničkog profila	inženjerstvo semantičkih web aplikacija proces prilagodbe uvođenja personalizacije i preporučenog sadržaja
4	nema	omogućiti pristup zaštićenim podacima – uvođenje podrške sustavu za upravljanje identitetom	inženjerstvo semantičkih web aplikacija proces prilagodbe integracije više izvora podataka
5	nema	implementirati praćenje ponašanja korisnika	proces prilagodbe uvođenja personalizacije i preporučenog sadržaja
6	nema	uvesti trajno spremište semantičkih podataka	proces prilagodbe uvođenja funkcionalnosti pretraživanja
7	1	podatke postojeće relacijske baze podataka semantički opisati	proces prilagodbe integracije više izvora podataka

broj aktivnosti	preduvjeti	opis aktivnosti	pripadnost aktivnosti prethodno predloženom postupku
8	3	uvesti semantički korisnički profil	proces prilagodbe uvođenja personalizacije i preporučenog sadržaja
9	4, 7	kreirati pristupni sloj semantičkim podacima	inženjerstvo semantičkih web aplikacija proces prilagodbe integracije više izvora podataka
10	2, 5, 6	izgraditi sloj preslikavanja podataka o praćenju ponašanja korisnika u RDF	proces prilagodbe integracije više izvora podataka
11	9, 10	omogućiti zaključivanje (povezivanje) nad dobivenim RDF grafovima – podrazumijeva i uvođenje semantičkog repozitorija	inženjerstvo semantičkih web aplikacija
12	11	izgraditi sloj preslikavanja RDF grafova u objektni model (uz prilagodbu objektnog modela)	inženjerstvo semantičkih web aplikacija proces prilagodbe integracije više izvora podataka
13	12	prilagoditi objektni model	proces prilagodbe uvođenja funkcionalnosti pretraživanja
14	13	implementirati algoritam traženja preporuka (dodatna prilagodba objektnog modela)	proces prilagodbe uvođenja personalizacije i preporučenog sadržaja
15	14	implementirati mogućnost ocjenjivanja i unosa preferencija	proces prilagodbe uvođenja personalizacije i preporučenog sadržaja
16	15	implementirati prikaz preporučenog sadržaja	proces prilagodbe uvođenja personalizacije i preporučenog sadržaja



Slika 81 Tijek procesa prilagodbe uvođenja funkcionalnosti preporučenog sadržaja u postojeće aplikacije po zaduženjima

7.7.1.1. Promjene MVC obrasca - model

Uvođenje ontologije domene

Pretpostavka je da se web aplikacija čija se prilagodba ovdje razmatra temelji na relacijskoj bazi podataka. Znači, potrebno je semantički opisati podatke domene pohranjene u relacijskoj bazi podataka. Provođenjem inženjerstva ontologije kreira se ontologija domene kojom se ti podaci semantički opisuju. Relacijska baza podataka se zadržava, kao i pristup putem SQL-a zbog zadržavanja funkcionalnosti aplikacije i minimizacije promjena.

Uvođenje semantičkog korisničkog profila

Kako bi se moglo realizirati zaključivanje nad ontologijom domene i nad profilom korisnika, profil je također potrebno semantički opisati. Predloženo je korištenje sustava za upravljanje identitetom u svrhu podrške semantičkog opisa profila budući da taj sustav nudi rješenje i na učestali problem kod aplikacija preporučenog sadržaja, a to je privatnost korisnika. Dakle, novi zahvat koji je potrebno poduzeti u postojećoj aplikaciji je upravljanje pristupom aplikaciji, kao i održavanje korisničkog profila, povjeriti sustavu za upravljanje identitetom. Gledajući s implementacijske razine radi se uvođenju veze između dotad korištenog korisničkog imena i OpenID URI-a korisnika, kao i promjenama u modulu koji kontrolira pristup aplikaciji.

Praćenje ponašanja korisnika

Načini praćenja ponašanja korisnika problem su sam za sebe. Ovdje se predlaže ili korištenje analize web loga ili snimanje akcija koje korisnik obavlja tijekom korištenja aplikacije. Predlaže se uvođenje realizacije snimanja akcija korisnika na način predstavljen u radu [Schmidt2007].

Integriranje podataka

Kako bi se omogućilo zaključivanje u svrhu preporuke sadržaja, podatke o ponašanju korisnika, profil korisnika, kao i podatke domene potrebno je integrirati. Moguće je da je za potrebe aplikacije potrebno integrirati i druge korisniku zanimljive podatke, iz drugih izvora podataka. Tehnike integriranja kako javnih, tako i zaštićenih izvora podataka preuzimaju se iz prethodno opisanog procesa prilagodbe (poglavlje 7.5) i neće se ovdje dodatno razmatrati.

Implementacija algoritma

Promjene koje je potrebno implementirati u ovom koraku teško je predvidjeti, jer su ovisne o domeni aplikacije i cilju koji se aplikacijom želi postići.

7.7.1.2. Promjene MVC obrasca - izgled

Implementacije koje se trebaju odraditi u ovom dijelu MVC obrasca uglavnom podrazumijevaju izgradnju novih stranica, a ne toliko izmjenu postojećih.

Potrebno je stvoriti nove stranice ili promijeniti postojeće tako da sadržavaju prikaz preporučenog sadržaja. Tehničko rješenje ocjenjivanja sadržaja od strane korisnika poželjno je razviti u skladu s paradigmom razvoja bogatih internet aplikacija, jer je vrlo važno podržati dinamičko sučelje, koje korisnika neće ometati u radu, budući da povratna informacija korisnika ima veliku vrijednost za postupak preporuke sadržaja.

7.7.2. Ocjena primjenjivosti

Osnovna motivacija za uvođenje personalizacije u postojeće aplikacije je pokušaj da se utječe na povećanje zadovoljstva korisnika. Kod ocjene primjenjivosti uvođenja preporučenog sadržaja u postojeće aplikacije naišlo se na sličan problem kao i kod ocjene primjenjivosti uvođenja pretraživanja. Naime, kako bi se moglo procijeniti koliku korist donosi implementacija preporučenog sadržaja potrebno je udovoljiti sljedećim uvjetima:

- velik broj korisnika – utječe na efikasnost skupnog filtriranja sadržaja
- velik broj podataka nad kojima se traže preporuke – utječe na efikasnost klasičnog filtriranja sadržaja

Ocjena primjenjivosti predloženog načina realizacije preporučenog sadržaja bila je zamišljena u sklopu izrade semantičkog web portala Sweb. No, budući da je portal rađen za okvire Hrvatske, pokazalo se velikim problemom pronaći dovoljno velik skup izvora podataka na hrvatskom jeziku s dovoljno velikom izražajnosti samih podataka nad kojima bi se moglo obavljati sugeriranje preporučenog sadržaja. Dodatno, kako se predlaže traženje sadržaja na osnovu ocjena korisnika, poželjno je imati što veći broj korisnika. Sweb je eksperimentalan sustav, pa kao takav nije pušten u produkciju, te se procijenilo da bi broj korisnika koji bi mogao sudjelovati u evaluaciji također bio ograničen. Dakle, osim evaluacije na eksperimentalnoj razini, dodatne rezultate u takvim uvjetima teško je ostvariti.

Što se tiče procjene troška uvođenja preporučenog sadržaja, većina koraka je poznata iz prethodno predloženih procesa prilagodbe. Kompleksnost izrade

algoritma koji bi obavljao filtriranje sadržaja na temelju semantičke sličnosti ovisi isključivo o domeni aplikacije i karakteristikama izgrađene ontologije domene. Za izradu skupnog filtriranja moguće se voditi već postojećim rješenjima, jer ovisnost o domeni aplikacije nije izražena. Također, potrebne su značajnije promjene u korisničkom sučelju kako bi se podržalo ocjenjivanje sadržaja od strane korisnika, kao i dodavanje mogućnosti pregleda preporučenog sadržaja.

Iako proces prilagodbe sadrži velik broj aktivnosti, koje su vremenski zahtjevne, kako ih je većina poznata iz inženjerstva web aplikacija procjenjuje se da je provedba cijelog procesa prilagodbe ostvariva.

7.8. Uvođenje značenja prikazanog sadržaja

U trenutku provođenja ovog istraživanja još uvijek nije bio postignut konsenzus oko načina zajedničke implementacije RDFa oznaka i povezanih podataka. Postoje različita mišljenja o tome kako bi se RDFa i povezani podaci mogli međusobno koristiti [Halb2008]. No, budući da je RDFa W3C preporuka te već postoje preglednici i agenti koji su u stanju interpretirati takav sadržaj, mogu se prepoznati sljedeći razlozi za izradu aplikacija s implementiranim označavanjem u prezentacijskom sloju:

- poboljšavanje rezultata pretraživanja – mogućnost korištenja semantičkih tražilica (na primjer yahoo searchmonkey[42] ili Google[43])
- mogućnost korištenja semantičkih preglednika
- mogućnost korištenja semantičkog okruženja (na primjer korištenje SemClipa [Reif2007] koji omogućava semantički svjesno kopiranje podataka između različitih aplikacija)

Premda postoje razna programska rješenja koja nude potporu uvođenju semantike u model podataka (na primjer objektni i relacijski model), takva programska rješenja ne postoje i za prikaz te iste semantike u prezentacijskom sloju. Potreba za takvim rješenjima već je u poglavlju 6.3.1.1.

Budući da je prisutan nedostatak programskih rješenja koja se mogu koristiti kao potpora uvođenju semantike u prezentacijski sloj (izgled dio MVC obrasca), u sklopu ovog istraživanja predloženi su mogući načini uvođenja semantike u prezentacijski sloj postojeće aplikacije. Razlikuju se dva osnovna slučaja za koja se predlažu različite metodologije:

- u postojeću web aplikaciju uvodi se samo značenje prikazanog sadržaja

[42] http://developer.yahoo.com/searchmonkey/
[43] http://www.google.com/support/webmasters/bin/answer.py?hl=en&answer=99170

- web aplikacija je prethodno podvrgnuta nekom drugom procesu prilagodbe semantičkom webu te se dodatno uvodi značenje prikazanog sadržaja

Za slučaj kada se u postojeću aplikaciju samo uvodi značenje prikazanog sadržaja realiziran je automatski proces uvođenja semantike. Cjelokupni proces objašnjen je u radu [Rovan2008b], ali se također analizira i u nastavku. Za sve ostale slučajeve predlaže se prilagodba sloja preslikavanja RDF grafova u objektni model.

7.8.1. Automatski proces prilagodbe

Realizacija automatskog procesa prilagodbe temelji se na uvođenju veza između objektnog modela domene i ontologija kojima se opisuju podaci. Dinamički sadržaj aplikacije koji se prikazuje korisniku na ekranu ostvaruje se pozivom članskih funkcija JavaBean-ova (npr. `student.getFirstName()`) iz JSP stranica. JavaBean-ovi dio su objektnog modela domene. Slika 82 prikazuje JSP kôd koji sadrži pozive JavaBean funkcija te rezultat izvođenja takvog kôda prikazanog u XHTML-u.

```
<html>
...
<table>
   <tr>
      <th>JMBAG</th>
      <th>Ime</th>
      <th>Prezime</th>
   </tr>
   <%
   Iterator it = students.iterator();
   while (it.hasNext()) {
      Student student = (Student)it.next();
   %>
   <tr>
      <td><%=student.getCode()%></td>
      <td><%=student.getFirstName()%></td>
      <td><%=student.getLastName()%></td>
   </tr>
   <%}%>
</table>
...
</html>
```

```
<table>
  <tr>
      <th>JMBAG</th>
      <th>Ime</th>
      <th>Prezime</th>
  </tr>
  <tr>
      <td>0036390867</td>
      <td>Jasmina</td>
      <td>Drmić</td>
  </tr>
  <tr>
      <td>0036405741</td>
      <td>Igor</td>
      <td>Gotovac</td>
  </tr>
</table>
```

Slika 82 Prikaz JSP kôda i dinamički generiranog XHTML-a

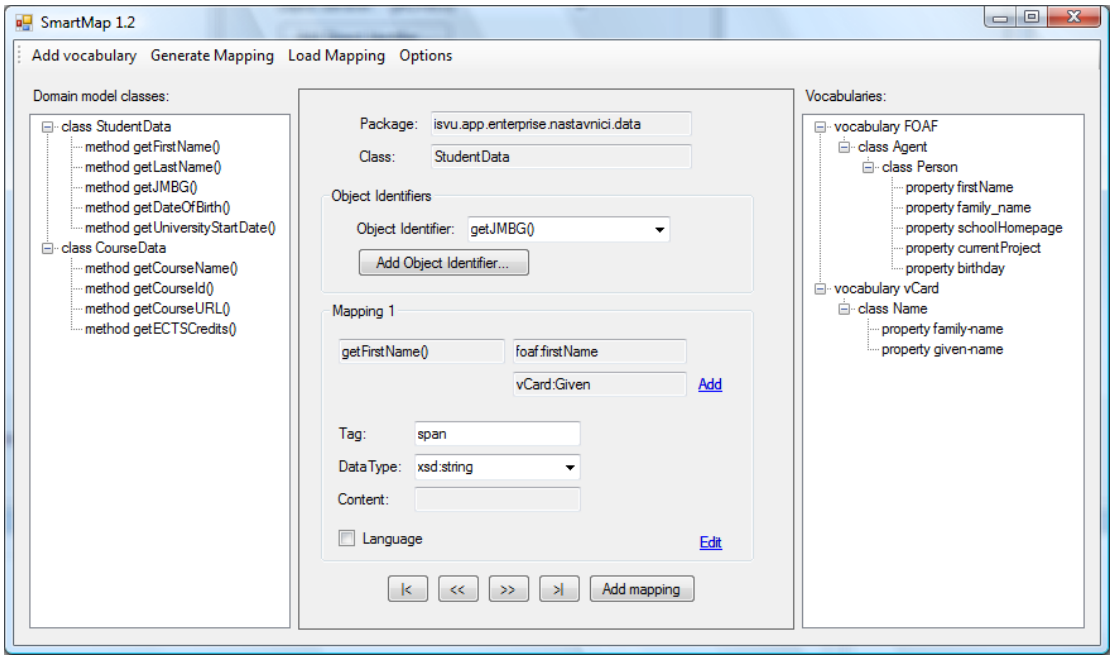
Problem koji se ovdje proučava je kako tim dinamičkim podacima za vrijeme izvođenja aplikacije dati značenje definirano odgovarajućem ontologijom. Kao tehnika označavanja odabrana je tehnologija RDFa. Cilj je postići da članske funkcije JavaBean-ova, koje se koriste za prikaz podataka u JSP-u, vraćaju uz samu vrijednost podatka i njegovo značenje. Značenje se pridodaje korištenjem RDFa

oznaka. Slika 83 prikazuje sadržaj u XHTML-u označen na takav način. Dakle, problem je kako uvesti veze između podatka i njegove definicije u ontologiji te kako tu istu vezu iskoristiti pri prikazu samog podatka.

```
<table>
  <tr>
      <th>Rbr</th>
      <th>JMBAG</th>
      <th>Ime</th>
      <th>Prezime</th>
  </tr>
  <tr>
      <td>1</td>
      <td>0036390867</td>
      <td>
          <span about="#0036390867" typeof="foaf:Person">
              <span property="foaf:firstName">Jasmina</span>
          </span>
      </td>
      <td>
          <span about="#0036390867" typeof="foaf:Person">
              <span property="foaf:surname">Drmić</span>
          </span>
      </td>
  </tr>
  <tr>
      <td>2</td>
      <td>0036405741</td>
      <td>
          <span  about="#0036405741" typeof="foaf:Person">
              <span property="foaf:firstName">Igor</span>
          </span>
      </td>
      <td>
          <span  about="#0036405741" typeof="foaf:Person">
              <span property="foaf:surname">Gotovac</span>
          </span>
      </td>
  </tr>
</table>
```

Slika 83 Semantički označen sadržaj

Kao potpora provođenja opisanog procesa uvođenja veza između objektnog modela i ontologije domene predlaže se korištenje pomoćnog alata. Izgled izgrađenog prototipa takvog alata prikazuje slika 84.



Slika 84 Izgled alata za definiranje veza između objektnog modela domene i ontologije

Veze koje se definiraju alatom predstavljaju trojke (RDF izjave). Subjekt je jednak jedinstvenom identifikatoru objekta (URI resursa), predikat odgovara članskoj funkciji, dok je objekt rezultat funkcije (vrijednost). Veze se ostvaruju odabirom odgovarajuće članske funkcije klase iz hijerarhije objektnog modela prikazanog u lijevom prozoru alata i svojstva ontologije (veze mogu biti N:N) prikazanog u desnom prozoru alata. Identifikator objekta čini jedna ili više funkcija te je moguće definirati format po kojem se identifikator (URI) izgrađuje. Npr. identifikator studenta može biti:

- student.getCode() → URI = http://www...../0036369145
- student.getFirstName() + student.getLastName() → URI = http://www...../IgorGotovac

Moguće je definirati i kojoj klasi ontologije pripada objekt (npr. objekt student odgovara klasi foaf:Person). Dodatne mogućnosti koje se mogu specificirati su:

- koja XHTML oznaka će se koristiti za generiranje RDFa značenja (npr. span, p, tr, h1)
- tip podatka
- oblik zapisa (RDFa oznaka „content“)
- definiranje jezika (ako aplikacija podržava internacionalizaciju definira se funkcija koja vraća jezik korišten u aplikaciji)

Nakon što se definiraju svi potrebni podaci za izradu objektnog modela podataka s ugrađenom podrškom za automatsko označavanje sadržaja, alat generira XML s opisom veza između objektnog modela i ontologije domene (slika 85). Također, alatom se generiraju klase koje predstavljaju proširenje objektnog modela domene. Za svaku klasu za koju je definirano preslikavanje, stvara se nova „omotač“ klasa koristeći „decorator“ obrazac. Slika 86 prikazuje klasu koja je generirana korištenjem pomoćnog alata. U generiranim klasama kreiraju se samo one funkcije koje su korištene tijekom definicije pravila preslikavanja. Sve članske funkcije generiranih klasa predstavljaju nadjačanja funkcija u roditelj klasi kako bi se njihovim pozivom zajedno s podatkom dohvatile i pripadajuće RDFa oznake.

```
....

<ontologija>
  <naziv>foaf</naziv>
  <xmlns>http:/xmlns.com/foaf/0.1/</xmlns>
  <klase>
    <klasa>
      <naziv>Person</naziv>
      <svojstva>
        <svojstvo>
          <naziv>firstName</naziv>
          <tag>span</tag>
          <property>yes</property>
          <resource>no</resource>
          <rel>no</rel>
          <href>no</href>
          <preslikavanjaObjektMetode>
            <paketi>
              <paket>
                <naziv>isvu.app.enterprise.nastavnici.data</naziv>
                <klase>
                  <klasa>
                    <naziv>StudentData</naziv>
                    <metode>
                      <metoda>
                        <naziv>getIme</naziv>
                        <language>yes</language>
                      </metoda>
                    </metode>
                  </klasa>
                </klase>
              </paket>
            </paketi>
          </preslikavanjaObjektMetode>
        </svojstvo>
      </svojstva>
    </klasa>
  </klase>

....|
```

Slika 85 Primjer generirane XML datoteke

```
public class StudentDataSemWeb extends StudentData{

    private StudentData student;
    private String paket;
    private String klasa;


    public StudentDataSemWeb(StudentData student) {
        this.student = student;
        paket = this.student.getClass().getPackage().getName();
        klasa = this.student.getClass().getSimpleName();
    }


    public String getIme() {
        return new AnnotationHelper().getAnotiraniSadrzaj(paket, klasa, "getIme", student);
    }

    public String getPrezime() {
        return new AnnotationHelper().getAnotiraniSadrzaj(paket, klasa, "getPrezime", student);

    }

    public String getJMBAG() {
        return new AnnotationHelper().getAnotiraniSadrzaj(paket, klasa, "getJMBAG", student);
    }


}
```

Slika 86 Generirana klasa za prikaz semantike

```
<html>
...
<table>
   <tr>
      <th>JMBAG</th>
      <th>Ime</th>
      <th>Prezime</th>
   </tr>
   <%
   Iterator it = students.iterator();
   while (it.hasNext()) {
      Student student = (Student)it.next();
   %>
   <tr>
      <td><%=student.getCode()%></td>
      <td><%=student.getFirstName()%></td>
      <td><%=student.getLastName()%></td>
   </tr>
   <%}%>
</table>
...
</html>
```

Slika 87 Izmijenjen izgled - dodan poziv generiranog objekta

Sve dosad objašnjeno vrijedi za opis prikazanog sadržaja, dakle trojki kojima je objekt upravo vrijednost prikazana na ekranu. No, na web stranicama postoje i značenja koja nisu tako eksplicitno zadana. Na primjer odnos između prikazanih sadržaja. Gledajući web stranicu koju prikazuje slika 88 korisnik prepozna značenje da svi navedeni studenti u tablici pripadaju grupi 1.D_RI i da su upisani u akademsku godinu 2009. Veze koje predstavljaju implicitno značenje bi također

trebalo eksplicitno navesti RDFa izjavama, inače je ono za agente semantičkog weba nedostupno.

Slika 88 Prikaz web stranice aplikacije Nastavnički portal

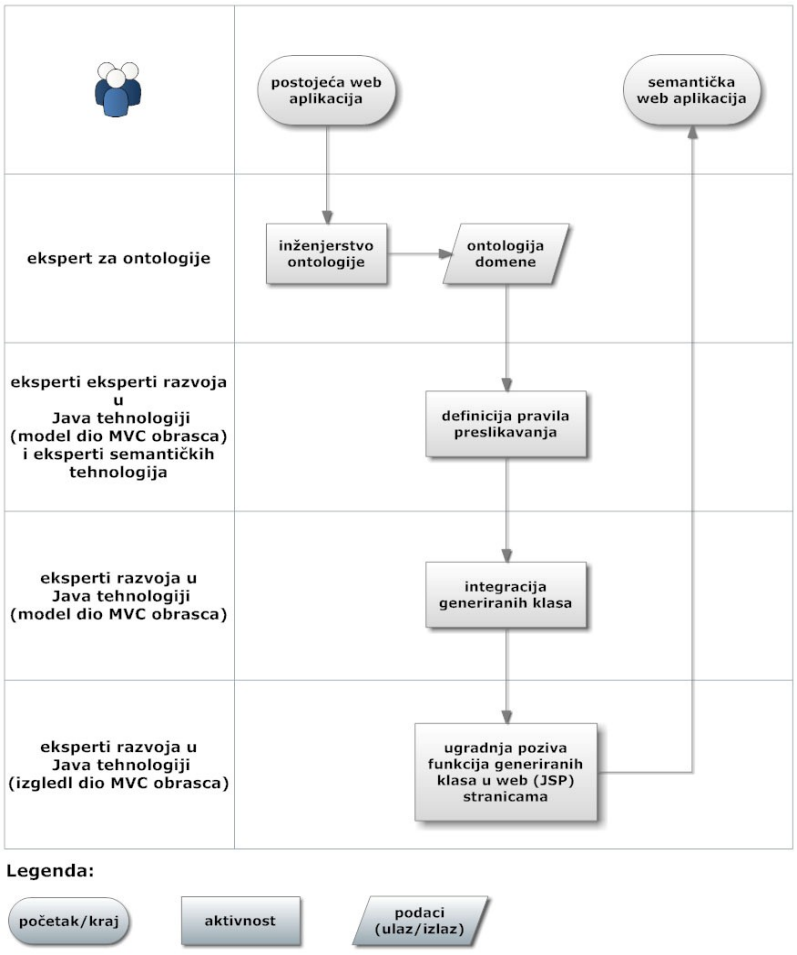
Opis implicitnog značenja ostvaruje se trojkama u kojima je objekt izjave resurs, dakle druga instanca klase korištena u aplikaciji. Kako bi se pružila podrška opisu i takvog implicitnog značenja prikazanog sadržaja, predlaže se proširenje navedenog pristupa u kojem bi se dodatno mogle definirati i veze (predikati) između klasa. Ukoliko je zamišljeno da aplikacija koja se izgrađuje je u skladu s konceptom povezanih podataka, ovaj dio je nepotrebno uvoditi. Naime, prema načelima povezanih podataka svaki podatak mora biti moguće razriješiti, što znači da je dovoljno za svaki podatak prikazan na ekranu definirati njegov URI (što je postignuto ovim načinom prilagodbe). Pristupom tom URI-u putem semantičkih web agenata ili preglednika semantičkog weba moguće je dohvatiti njegova dodatna svojstva i poveznice s drugim pojmovima semantičkog weba.

7.8.2. Metodologija prilagodbe

Uvođenje značenja prikazanog sadržaja u postojeću aplikaciju zahtijeva poduzimanje aktivnosti koje su specifične isključivo ovom procesu prilagodbe, osim samog kreiranja ontologije domene. Primjena predloženog procesa prilagodbe podrazumijeva poduzimanje aktivnosti koje sadržava tablica 14. Također, koje bi eksperte trebalo uključiti za obavljanje pojedinih koraka procesa prilagodbe prikazuje slika 89.

Tablica 14 Metodologija prilagodbe uvođenja značenja prikazanog sadržaja

broj aktivnosti	preduvjeti	opis aktivnosti	pripadnost aktivnosti prethodno predloženom postupku
1	nema	inženjerstvo ontologije – pronaći dostupne ontologije koje pokrivaju sadašnji model podataka, izgraditi dodatne ontologije i međusobno ih povezati	inženjerstvo semantičkih web aplikacija proces prilagodbe integracije više izvora podataka
2	1	definirati pravila preslikavanja	proces prilagodbe uvođenja značenja sadržaja
3	2	integrirati generirane klase (korištenjem API-a)	proces prilagodbe uvođenja značenja sadržaja
4	3	ugraditi pozive funkcija generiranih klasa u web (JSP) stranicama	proces prilagodbe uvođenja značenja sadržaja



Slika 89 Proces prilagodbe ugradnje značenja prikazanog sadržaja

7.8.2.1. Promjene MVC obrasca - model

Uz sam prototip pomoćnog alata, realiziran je i API kojim se generirane klase i XML datoteke integriraju u postojeću aplikaciju. Dakle, promjene koje je potrebno obaviti u *model* dijelu postojeće aplikacije podrazumijevaju uključivanje generiranih klasa te specifikacija parametara potrebnih za njihovu uspješnu primjenu korištenjem realiziranog API-a.

7.8.2.2. Promjene MVC obrasca - izgled

JSP stranice koje su dio *izgled* dijela obrasca potrebno je promijeniti na način da se na mjesta gdje se u njima pozivaju funkcije JavaBean-ova koje vraćaju sadržaj koji se prikazuje na ekranu, umetne generirani „omotač“ tih JavaBean-ova (slika 87). Postoje i slučajevi u kojima postoji veza između objekata, dakle kada je u svrhu davanja značenja prikazanog sadržaja potrebno izgraditi trojku kojoj je objekt resurs. Budući da se u tom slučaju ne uvodi oznaka sadržaja koji se eksplicitno prikazuje na ekranu pa se i ne koristi nadjačana funkcija, tada je potrebno eksplicitno uvesti poziv funkcije koja generira takve RDFa oznake.

7.8.3. Prilagodba sloja preslikavanja RDF grafova u objektni model

U slučaju kada je postojeća web aplikacija već podvrgnuta nekim od prethodno definiranih procesa prilagodbe, uvođenje opisa prikazanog sadržaja može se ostvariti prilagodbom sloja preslikavanja RDF grafova u objektni model te se smatra da je u tom slučaju primjena predloženog automatskog procesa prilagodbe neisplativa. Kao rezultat prethodno predloženih procesa prilagodbe objekti objektnog modela aplikacije već imaju ugrađeno svojstvo identificirajućeg URI-a pa se ne treba uvoditi nikakav dodatan mehanizam kojim bi se URI generirao. Promjene koje se dodatno moraju uvesti je da se slojem preslikavanja RDF grafova u objekte osigura da instancirani objekti ne sadrže samo vrijednosti podataka, nego i metapodatke o njihovom značenju. Takav pristup načelno je sličan ovdje predloženom automatskom označavanju sadržaja. Dakle, uz klasu koja predstavlja entitete iz semantičkog modela kreira se i „omotač klasa“. Prilikom preslikavanja RDF grafova u objektni model meta podaci su već poznati, oni i čine RDF graf, pa ih je potrebno prenijeti i u omotač klasu. Ovaj pristup nije realiziran u okviru ovog istraživanja i ostavlja se za budući rad.

7.8.4. Ocjena primjenjivosti

Verifikacija predloženog automatskog procesa prilagodbe obavljena je prilagodbom postojeće aplikacije „Nastavnički portal“, kao i aplikacije za praćenje dnevnika rada razvijane programskim okvirom Tapestry.

Predloženi postupak pokazao se primjenjiv u oba navedena slučaja. Najveći problem davanja ocjene primjenjivosti proizlazi iz nedefiniranosti načina na koji bi se povezani podaci i RDFa trebali koristiti u kombinaciji. Kako je trenutno nemoguće pretpostaviti točan način na koji će se RDFa koristiti od strane preglednika semantičkog weba, pa i semantičkih RDFa agenata, teško je odrediti korist koju primjena RDFa oznaka u postojeću web aplikaciju donosi.

Neki od tehničkih problema koji su prepoznati prilikom primjene ovog procesa prilagodbe je da ukoliko web stranice nisu pisane po XHTML sintaksi tada obogaćivanje stranica RDFa tehnologijom nije moguće. No, postoje alati koji pomažu pri preoblikovanju dokumenata pisanih HTML sintaksom u XHTML sintaksu, pa je u tom slučaju također proces moguće provesti, ali uz nešto više truda. Prisutni su i problemi u slučaju korištenja Ajaxa i Javscripta pri izradi JSP stranica, koji su prethodno objašnjeni kod elaboriranja RDFa tehnologije u poglavlju 4.4.4.

S pretpostavkom da će se RDFa koristiti samo za opis prikazanog sadržaja, bez uvođenja značenja između prikazanih sadržaja, te da će se sva dodatna semantika dohvaćati razrješavanjem URI-a, ovaj proces prilagodbe ocjenjuje se visokom ocjenom primjenjivosti.

7.9. Osvrt na provedivost cjelokupnog procesa prilagodbe

U ovom poglavlju predložen je cjelokupni proces prilagodbe postojećih web aplikacija semantičkom webu. Prepoznato je više mogućih razloga za poduzimanje prilagodbe. Za svaki razlog predložena je metodologija koja propisuje slijed aktivnosti koje je potrebno obaviti u svrhu prilagodbe. Cjelokupni tijek aktivnosti koje se poduzimaju za prilagodbu određene aplikacije nastaje slijednim poduzimanjem predloženih metodologija prema definiranim pravilima.

Pregledom prethodnih istraživanja u području prilagodbe aplikacija semantičkom webu utvrđeno je da je značajniji napredak ostvaren na području preslikavanja relacijskih baza podataka u semantičke modele, te na području uvođenja semantike u sam prikaz web aplikacija. Također, problem integracije podataka je područje koje se istražuje već dugi niz godina, pa se opravdano očekivalo da će rezultati

prilagodbe u svrhu uvođenja integracije više izvora podataka biti lakše ostvarivi u usporedbi s ostalim razlozima prilagodbe. Tehnologije i alati nastali kao rezultat istraživanja na spomenutom području znatno olakšavaju rješavanje problema kod provedbe prilagodbe. Tablica 15 prikazuje za svaku aktivnost koje bi semantičke tehnologije trebalo poznavati i koristiti, te koji su alati korišteni prilikom verifikacije aktivnosti predloženog procesa prilagodbe.

Tablica 15 Pregled alata korištenih pri verifikaciji metodologije prilagodbe uvođenja integracije više izvora podataka

broj aktivnosti	opis aktivnosti	dostupni alati	semantičke tehnologije
1	inženjerstvo ontologije – pronaći dostupne ontologije koje pokrivaju sadašnji model podataka, izgraditi dodatne ontologije i međusobno ih povezati	Protégé	RDF, RDFS, OWL
2	omogućiti pristup zaštićenim podacima – uvođenje podrške sustavu za upravljanje identitetom	sustav za upravljanje identitetom	RDF, RDFS, SPROT
3	podatke postojeće relacijske baze podataka semantički opisati	D2RQ, D2R	RDF, RDFS, OWL, SPARQL
4	izraditi "ontologiju preslikavanja" u svrhu povezivanja RDF grafova		RDF, RDFS, OWL
5	kreirati pristupni sloj semantičkim podacima	Jena	RDF, SPARQL, SPROT
6	omogućiti zaključivanje (povezivanje) nad dobivenim RDF grafovima – podrazumijeva i uvođenje semantičkog repozitorija	Jena – ugrađeni mehanizam za zaključivanje	RDF, RDFS, OWL
7	izgraditi sloj preslikavanja RDF grafova u objektni model (uz prilagodbu objektnog modela)	Jena	RDF, RDFS, OWL, SPARQL
8	dodatno prilagoditi objektni model (udruživanje objekata iz više izvora)		RDF, povezani podaci
9	izgraditi/izmijeniti sučelje		

Premda je uvođenje značenja prikazanog sadržaja u postojeće aplikacije također bilo predmet više istraživanja, promjene koje su nastale u pravcima istraživanja semantičkog weba, promijenile su aktualnost korištenih tehnologija pa i sam koncept označavanja prikaza sadržaja. Za realizaciju ovog razloga prilagodbe razvijen je prototip alata koji se koristi kao potpora provedbi prilagodbe, te koji se temelji na novijoj tehnologiji - RDFa.

Osim integracije više izvora podataka i uvođenja značenja prikazanog sadržaja, kao mogući razlozi prilagodbe prepoznati su i uvođenje semantičkog pretraživanja, kao i uvođenje personalizacije/preporučenog sadržaja u postojeću aplikaciju.

Problem semantičkog pretraživanja je također problem koji se duže vremena proučava. No, i na ovaj problem promjena pravaca istraživanja semantičkog weba ima znatan utjecaj. Tek odnedavno su dostupne tehnologije i alati koji su značajni za njegovu realizaciju, kao što su tehnologije označavanja i alati za izračun semantičkih sličnosti. Tablica 16 prikazuje sve korake prilagodbe uvođenja funkcionalnosti pretraživanje, te pregled korištenih alata pri verifikaciji predložene metodologije.

Tablica 16 Pregled alata korištenih pri verifikaciji metodologije prilagodbe uvođenja funkcionalnosti pretraživanja

broj aktivnosti	opis aktivnosti	dostupni alati	semantičke tehnologije
1	inženjerstvo ontologije – pronaći dostupne ontologije koje pokrivaju sadašnji model podataka, izgraditi dodatne ontologije i međusobno ih povezati	Protégé	RDF, RDFS, OWL
2	uvesti URI-e postojećeg sadržaja (voditi računa o razrješavanju URI-a)		povezani podaci
3	uvesti trajno spremište semantičkih podataka	Jena + MSSQL	RDF, RDFS, OWL, SPARQL
4	kreirati pristupni sloj semantičkim podacima	Jena	RDF, SPARQL, SPROT
5	omogućiti zaključivanje / izračun sličnosti nad dobivenim RDF grafovima	Jena – ugrađeni mehanizam za zaključivanje	RDF, RDFS, OWL
6	izgraditi sloj preslikavanja RDF grafova u objektni model	Jena	RDF
7	dodatno prilagoditi objektni model (udruživanje objekata)		RDF, povezani podaci

8	izgraditi sloj za uparivanje oznaka i sadržaja (dodatna prilagodba objektnog modela)		RDF
9	uvođenje podrške semantičkog označavanja sadržaja	MOAT	RDF, RDFS, SPARQL
10	implementirati algoritam pretraživanja (metrike sličnosti, kategorizacija, itd…)	Silk	RDF, RDFS, OWL, povezani podaci, SPARQL
11	implementirati mogućnost označavanja i pretraživanja putem sučelja		

Personalizacija, kao i traženje preporučenog sadržaja funkcionalnost je koja je u posljednje vrijeme nezaobilazna u društvenim web aplikacijama, pa i u web portalima. Istraživanja na tom području su zanemariva, pa je teško temeljiti rad na već postojećim rezultatima, stoga se ova prilagodba od svih predloženih smatra najteže provedivom. Također, za ovu prilagodbu potrebno je provesti najveći broj aktivnosti. Slijed aktivnosti, te koje alate je moguće koristiti pri kojoj aktivnosti prikazuje tablica 17.

Tablica 17 Pregled alata korištenih pri verifikaciji metodologije prilagodbe uvođenja funkcionalnosti preporučenog sadržaja

broj aktivnosti	opis aktivnosti	dostupni alati	semantičke tehnologije
1	inženjerstvo ontologije – pronaći dostupne ontologije koje pokrivaju sadašnji model podataka, izgraditi dodatne ontologije i međusobno ih povezati	Protégé	RDF, RDFS, OWL
2	inženjerstvo ontologije – kreirati ontologiju ponašanja korisnika	Protégé	RDF, RDFS, OWL
3	inženjerstvo ontologije – kreirati ontologiju korisničkog profila	Protégé	RDF, RDFS, OWL
4	omogućiti pristup zaštićenim podacima – uvođenje podrške sustavu za upravljanje identitetom	Sustav za upravljanje identitetom	RDF,RDFS, SPROT
5	implementirati praćenje ponašanja korisnika		
6	uvesti trajno spremište semantičkih podataka	Jena + MSSQL	RDF, RDFS, OWL, SPARQL

7	podatke postojeće relacijske baze podataka semantički opisati	D2RQ, D2R	RDF, RDFS, OWL, SPARQL
8	uvesti semantički korisnički profil	sustav za upravljanje identitetom	RDF, RDFS, SPROT
9	kreirati pristupni sloj semantičkim podacima	Jena	RDF, SPARQL, SPROT
10	izgraditi sloj preslikavanja podataka o praćenju ponašanja korisnika u RDF	Jena	RDF, RDFS, OWL
11	omogućiti zaključivanje (povezivanje) nad dobivenim RDF grafovima – podrazumijeva i uvođenje semantičkog repozitorija	Jena – ugrađeni mehanizam za zaključivanje	RDF, RDFS, OWL
12	izgraditi sloj preslikavanja RDF grafova u objektni model (uz prilagodbu objektnog modela)	Jena	RDF
13	prilagoditi objektni model		RDF
14	implementirati algoritam traženja preporuka (dodatna prilagodba objektnog modela)	Silk, Jena, mehanizam za zaključivanje	RDF, RDFS, OWL
15	implementirati mogućnost ocjenjivanja i unosa preferencija		
16	implementirati prikaz preporučenog sadržaja		

Ocjena primjenjivosti procesa prilagodbe, kao procesa koji dodaje vrijednost, vrlo je složena. Najveći problem ocjenjivanja primjenjivosti predstavlja nemogućnost procjene koristi koja se ostvaruje poduzetom prilagodbom. Zbog toga se ocjene dane u ovom poglavlju temelje na procjeni troška provedbe prilagodbe. Izuzetak je uvođenje integracije više izvora podataka kod kojeg se ipak mogla dati komparacija s ostalim klasičnim pristupima ostvarenja integracije te na taj način procijeniti korist poduzete prilagodbe. Dok god semantički web ne zaživi kao okruženje, što bi rezultiralo većim brojem semantički opisanih podataka, kao i korisnika na čijim dojmovima se može temeljiti evaluacija korisnosti aplikacija, teško je donijeti kvalitetne procjene isplativosti prilagođavanja postojećih web aplikacija tom novom okruženju.

8.Zaključak

U ovom istraživanju proveden je pregled područja razvoja weba, kao i razvoja web aplikacija, koji je bio posebno usmjeren na web aplikacije pisane jezikom Java. Provedena analiza je pokazala da se radi o promjenjivom području u kojem se vrlo brzo izmjenjuju aktualne tehnologije i standardi.

Provedena je i opsežna analiza razvoja semantičkog weba, kao i tehnologija i metodologija njihove primjene koje su nastale kao rezultat istraživanja u tom području. Razvoj semantičkog weba traje duže nego što se to inicijalno očekivalo. U ovom istraživanju kao jedan od uzroka sporog razvoja rješenja semantičkog weba prepoznata je promjenjivost standarda i aktualnosti tehnologija samog weba, koje utječu na to da se i standardi semantičkog weba moraju stalno usklađivati s njima. Također, na temelju provedene analize smatra se da su dodatan utjecaj na nedostatak rješenja semantičkog weba ostavile i promjene u pravcima istraživanja u samom području semantičkog weba.

Budući da je semantički web promjenjivo okruženje, sva istraživanja u području inženjerstva semantičkih web aplikacija brzo zastarijevaju. U ovom istraživanju provedena je analiza dostupnih rješenja semantičkog weba kako bi se mogli postaviti teorijski okviri razvoja semantičkih web aplikacija. Nastajalo se ostvariti da ti teorijski okviri slijede današnje standarde razvoja web aplikacija, te da su u skladu s načelima novih pravaca istraživanja unutar područja semantičkog weba. Rezultat je provedene analize kategorizacija semantičkih web aplikacija te prijedlog arhitekture za svaku od prepoznatih kategorija. Elementi arhitekture odabrani su u skladu s provedenom analizom tehnologija semantičkog weba. Poteškoću pri prijedlogu arhitekture predstavljala je činjenica da su novi pravci istraživanja unutar semantičkog weba novijeg datuma, pa još nije sasvim jasna slika primjene svih tehnologija koje su odobrene kao web standard, što ostavlja prostora za budući rad.

Tijekom analize učestalih funkcionalnosti semantičkih web aplikacija otkriven je problem pristupa zaštićenim podacima, koji je potreban u velikom broju slučaja uporaba semantičkih web aplikacija. Kao rješenje na uočen problem, predložen je sustav za upravljanje identitetom u okruženju semantičkog weba.

U okviru istraživanja realizirani su prototipovi prema predloženim arhitekturama kategorija semantičkog weba i prototip sustava za upravljanje identitetom.

Područje semantičkog weba interdisciplinarno je područje i realizacija rješenja semantičkog weba zahtijeva znanja iz više područja. Interdisciplinarnost je posebno

izražena kod izrade funkcionalnosti semantičkog pretraživanja, kao i kod funkcionalnosti izračuna preporučenog sadržaja. Budući da bi potpuna realizacija navedenih funkcionalnosti prelazila okvire ovog istraživanja, jer traži napredna znanja i područja pretraživanje informacija rad na tom području ostavlja se za buduća istraživanja.

Osnovni je cilj ovog istraživanja bio predložiti proces prilagodbe postojećih web aplikacija semantičkom webu. Tijekom analize semantičkih web aplikacija otkriveni su razlozi prilagodbe. Svaki razlog prilagodbe predstavlja uvođenje neke od prepoznatih funkcionalnosti karakterističnih za semantičke web aplikacije u postojeću aplikaciju. Za svaki razlog prilagodbe predložena je metodologija prilagodbe. Budući da je moguće da postoji više razloga za prilagodbu određene aplikacije, u ovom je istraživanju predložen i cjelokupni proces prilagodbe koji propisuje na koji način se povezuju i provode predložene metodologije u cijelosti.

Već spomenut problem interdisciplinarnosti područja semantičkog weba, te dodatno i malen broj dostupnih semantičkih izvora podataka onemogućio je da se provede kvalitetno ocjenjivanje primjenjivosti metodologije prilagodbe uvođenja funkcionalnosti pretraživanja, kao i uvođenja funkcionalnosti personalizacije i preporučenog sadržaja te se procjenjuje da će rad u tom području biti moguće u potpunosti ostvariti tek s pojavom većeg broja aplikacija i korisnika semantičkog weba. Ocjena primjenjivosti ostalih metodologija prilagodbe provedena je u potpunosti te se predložene metodologije smatraju u potpunosti ostvarivima.

Literatura

[Abel2007] Abel, F., Frank, M., Henze, N., Krause, D., Plappert, D., Siehndel, P.: ‘GroupMe!-Where Semantic Web meets Web 2.0’, Lecture notes in computer science, 2007, 4825, pp. 871-878

[Adida2008] Adida, B.: ‘hGRDDL: Bridging microformats and RDFa’, Web Semantics: Science, Services and Agents on the World Wide Web, 2008, 6, (1), pp. 54-60

[Adida2009a] Adida, B., Birbeck, M.: 'RDFa Primer 1.0', http://www.w3.org/TR/xhtml-rdfa-primer/, [11.05.2009.]

[Adida2009b] Adida, B., Birbeck, M., McCarron, S., Pemberton, S.: 'RDFa in XHTML: Syntax and Processing', http://www.w3.org/TR/rdfa-syntax/, [11.05.2009.]

[Alavi2001] Alavi, M., Leidner, D.: ‘Review: Knowledge management and knowledge management systems: Conceptual foundations and research issues’, MIS Quarterly, 2001, pp. 107-136

[Alkhateeb2008] Alkhateeb, F.: ‘Querying RDF (S) with regular expressions’, Université Joseph Fourier, 2008.

[Amardeilh2006] Amardeilh, F., Carloni, O., Noël, L.: ‘PressIndex: a Semantic Web Press Clipping Application’. In: Proceedings of the International Semantic Web CHallenge, Athens, Georgia, USA, 2006

[Anand2005] Anand, S., Mobasher, B.: ‘Intelligent techniques for web personalization’, Lecture notes in computer science, 2005, 3169, pp. 1-36

[Ankolekar2006] Ankolekar, A., Vrandecic, D.: ‘Personalizing web surfing with semantically enriched personal profiles’. Semantic Web Personalization Workshop, Budva, Montenegro, 2006

[Antoniou2008] Antoniou, G., and Van Harmelen, F.: ‘A Semantic Web Primer', (The MIT Press, 2008.)

[Anyanwu2007] Anyanwu, K., Maduko, A., Sheth, A.: ‘SPARQ2L: towards support for subgraph extraction queries in rdf databases’. In: Proceedings of 16th international conference on World Wide Web, Banff, Alberta, Canada, 2007, pp. 797-806

[Aroyo2007] Aroyo, L., Stash, N., Wang, Y., Gorgels, P., Rutledge, L.: ‘Chip demonstrator: Semantics-driven recommendations and museum tour generation’, Lecture notes in computer science, 2007, 4825, pp. 879–886

[Astrova2004] Astrova, I.: ‘Reverse engineering of relational databases to ontologies’, Lecture notes in computer science, 2004, pp. 327-341

[Atwood2009] Atwood, M., Conlan, R.: ' OAuth Core 1.0', http://OAuth.net/core/1.0/, [10.05.2009.]

[Auer2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data', Lecture notes in computer science, 2007, 4825, pp. 722

[Baresi2000] Baresi, L., Garzotto, F., Paolini, P.: 'From Web Sites to Web Applications: NewnIssues for Conceptual Modeling'. In: Proceedings of Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling: Conceptual Modeling for E-Business and the Web, Salt Lake City, Utah, USA, 2000, pp. 89-100

[Barrasa2004] Barrasa, J., Corcho, O., Gomez-Perez, A.: 'R2O, an extensible and semantically based database-to-ontology mapping language'. 2nd Workshop on Semantic Web and Databases, Toronto, Canada, 2004

[Battle2006] Battle, R., Benson, E.: 'Bridging the semantic Web and Web 2.0 with Representational State Transfer (REST)', Web Semantics: Science, Services and Agents on the World Wide Web, 2008, 6, (1), pp. 61-69

[Battista2007] Battista, A., Villanueva-Rosales, N., Palenychka, M., Dumontier, M.: 'SMART: A Web-Based, Ontology-Driven, Semantic Web Query Answering Application'. International Semantic Web Conference, Busan, Korea, 2007

[Baumgartner2005] Baumgartner, R., Henze, N., Herzog, M.: 'The personal publication reader: Illustrating web data extraction, personalization and reasoning for the semantic web'. In: Proceedings of, 2005, pp. 515–530

[Bellekens2007] Bellekens, P., Aroyo, L., Houben, G., Kaptein, A., Van Der Sluijs, K.: 'Semantics-Based Framework for Personalized Access to TV Content: The iFanzy Use Case', Lecture notes in computer science, 2007, 4825, pp. 887-894

[Berners-Lee2001] Berners-Lee, T., Hendler, J., and Lassila, O.: 'The semantic web', Scientific American, 2001, 284, (5), p. 34-43

[Berners-Lee2009a] Berners-Lee, T.: 'Cool URIs don't change', http://www.w3.org/Provider/Style/URI, [24.5.2009.]

[Berners-Lee2009b] Berners-Lee, T.: 'Linked data', http://www.w3.org/DesignIssues/LinkedData.html, [20.05.2009.]

[Bhagdev2007] Bhagdev, R., Butters, J., Chakravarthy, A., Chapman, S., Dadzie, A., Greenwood, M., Iria, J., Ciravegna, F.: 'Doris: Managing document-based knowledge in large organisations via semantic web technologies'. 6th International Semantic Web Conference Busan, Korea, 2007

[Bischoff2008] Bischoff, K., Firan, C.S., Nejdl, W., Paiu, R.: ‘Can all tags be used for search?’. Proceeding of the 17th ACM conference on Information and knowledge management, Napa Valley, California, USA, 2008

[Bizer2007] Bizer, C., Cyganiak, R.: ‘D2RQ-Lessons Learned’. W3C Workshop on RDF Access to Relational Databases, Cambridge, MA, USA, 2007

[Bizer2009a] Bizer, C., Cyganiak, R., Heath, T.: 'How to publish linked data on the web', http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/, [20.05.2009.]

[Bizer2009b] Bizer, C., Cyganiak, R.: ‘Quality-driven information filtering using the WIQA policy framework’, Web Semantics: Science, Services and Agents on the World Wide Web, 2009, 7, (1), pp. 1-10

[Bizer2009b] Bizer, C., Schultz, A.: 'The Berlin SPARQL Benchmark', http://www4.wiwiss.fu-berlin.de/bizer/BerlinSPARQL Benchmark/results/index.html, [06.04.2009.]

[Boehm2008] Boehm, S., Koolwaaij, J., Luther, M., Souville, B., Wagner, M., Wibbels, M.: ‘Introducing IYOUIT’. In: Proceedings of International Semantic Web Conference, Karlsruhe, Germany, 2008, pp. 804-817

[Boley2007] Boley, H., Kifer, M., Patranjan, P.,Polleres, A.: ‘Rule interchange on the web’, Lecture notes in computer science, 2007, 4636, pp. 269-309

[Boiko2005] Boiko, B.: ‘Content management bible 2nd edition’ (Wiley Publishing, Indianapolis, USA, 2005. 2005)

[Brickley2008] Brickley, D., Miller, L.: 'FOAF vocabulary specification 0.91', http://xmlns.com/foaf/spec/, [20.12.2008.]

[Brickley2009] Brickley, D., Guha, R.: 'Rdf vocabulary description language 1.0: Rdfschema', www.w3.org/TR/rdf-schema/, [19.04.2009.]

[Broekstraw2002] Broekstra, J., Kampman, A., van Harmelen, F.: ‘Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema’. In: Proceedings of First International Semantic Web Conference on The Semantic Web, Sardinia, Italy, 2002, pp. 54 – 68

[Brooks1987] Brooks, F.: ‘No silver bullet: Essence and accidents of Software engineering’, IEEE computer, 1987, 20, (4), pp. 10-19

[Brown2001] Brown, S., Li, S., Burdick, R., Sullivan, B., Nelson, M., Wilkinson, S., Johnson, R., Taylor, G., Falkner, J., and Galbraith, B.: ‘Professional JSP’ (Wrox Press Ltd. Birmingham, UK, UK, 2001. 2001)

[Byrne2009] Byrne, K.: 'Relational Database to RDF Translation in the Cultural Heritage Domain',http://homepages.inf.ed.ac.uk/s0233752/docs/rdb2rdfForCH.pdf, [1.9.2009.]

[Cardoso2008] Cardoso, J., Hepp, M., Lytras, M.: 'The Semantic Web Real-world Applications from Industry' (Springer, 2008. 2008)

[Carrascal2009] Carrascal, F., Rodrigo, L., Contreras, J., Carbone, F.: 'Cantabria Cultural Heritage Semantic Portal', http://ceur-ws.org/Vol-295/paper02.pdf, [6.7.2009.]

[Casanovas2005] Casanovas, P., Poblet, M., Casellas, N., Contreras, J., Benjamins, V., Blazquez, M.: 'Supporting newly-appointed judges: a legal knowledge management case study', Journal of Knowledge Management, 2005, 9, (5), pp. 7 – 27

[Celma2005] Celma, O., Ramirez, M., Herrera, P.: 'Getting music recommendations and filtering news feeds from FOAF descriptions'. 1st Workshop on Scripting for the Semantic Web co-located with the 2nd European Semantic Web Conference, Heraklion, Greece, 2005

[Cheng2008] Cheng, F., Yesha, Y., Joshi, A., Finin, T.: 'First results of the Ontology alignment evaluation initiative 2008'. 4th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2008), Karlsruhe, Germany, October 26 - 30, 2008.

[Clark2009] Clark, K.: 'SPARQL protocol for RDF', www.w3.org/TR/rdf-sparql-protocol/, [17.04.2009.]

[Clarke2009] Clarke, C.: 'A Resource List Management Tool for Undergraduate Students Based on Linked Open Data Principles'. 6th European Semantic Web Conference, Heraklion, Greece, 2009

[Conallen1999] Conallen, J.: 'Modeling web application architectures with UML', Commun. ACM, 1999, 42, (10), pp. 63-70

[Connolly2009] Connolly, D.: 'Gleaning resource descriptions from dialects of languages (GRDDL)', [4.5.2009.]

[Contreras2004] Contreras, J., Benjamins, V., Blázquez, M., Losada, S., Salla, R., Sevilla, J., Navarro, D., Casillas, J., Mompó, A., Paton, D.: 'A semantic portal for the international affairs sector', Lecture notes in computer science, 2004, pp. 203-215

[Corcho2006] Corcho, O., López-Cima, A., Gómez-Pérez, A.: 'A platform for the development of semantic web portals'. In: Proceedings of International Conference on Web Engineering, 2006, pp. 145-152

[Corno2008] Corno, W., Corcoglioniti, F., Celino, I., Valle, E.D.: 'Optimizing SPARQL querying over pre-existing non-RDF data sources via an object-oriented approach'. 4th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2008), Karlsruhe, Germany, October 26 - 30, 2008.

[Davis1989] Davis, F.: 'Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology', MIS Quarterly, 1989, (Sept.), pp. 319-339

[Davis2009] Davis, I.: 'RDF in HTML (eRDF)', http://research.talis.com/2005/erdf/wiki/Main/RdfInHtml, [22.5.2009.]

[Demeyer2003] Demeyer, S., Ducasse, S., Nierstrasz, O.: 'Object-oriented reengineering patterns' (Morgan Kaufmann, 2003.)

[Dietzold2006] Dietzold, S., Auer, S.: 'Access control on RDF triple stores from a semantic wiki perspective'. 2nd Workshop on Scripting for the Semantic Web at ESWC2006, Budva, Montenegro, 2006

[Dill2003] Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., Rajagopalan, S., Tomkins, A., Tomlin, J.: 'SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation'. In: Proceedings of, 2003, pp. 178-186

[Doyle2005] Doyle, B., and Lopes, C.: 'Survey of Technologies for Web Application Development', ACM Journal Name, 2005, 2, (3), pp. 1-43

[Erling2008] Erling, O., Mikhailov, I.: 'Towards Web-Scale RDF'. 4th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS2008), Karlsruhe, Germany, October 26 - 30, 2008.

[Falkner2004] Falkner, J., and Jones, K.: 'Servlets and JavaServer pages: the J2EE technology Web tier' (Addison-Wesley, 2004. 2004)

[Feng2006] Feng, Q., Mookerjee, V., Sethi, S.: 'Optimal policies for the sizing and timing of software maintenance projects', European Journal of Operational Research, 2006, 173, (3), pp. 1047-1066

[Fertalj2006] Fertalj, K., Hlupic, N., Rovan, L.: 'Why (not) ORM?'. In: Proceedings of 28th International Conference on Information Technology Interfaces, Cavtat, Croatia, 2006, pp. 683-688

[Fitzpatrick2008] Fitzpatrick, B.: 'OpenID Authentication 2.0', http://openid.net/specs/openid-authentication-2_0.html, [20.12.2008.]

[Gao2009] Gao, Z., Qu, Y., Zhai, Y., Deng, J.: 'Dynamic View: Distribution, Evolution and Visualization of Research Areas in Computer Science', Lecture notes in computer science, 2005, 3729, pp. 1054

[Gasevic2004] Gasevic, D., Djuric, D., Devedzic, V., Damjanovi, V.: 'Converting UML to OWL ontologies'. In: Proceedings of 13th International WWW Conference, New York, USA, 2004, pp. 488-489

[Ginige2001] Ginige, A., and Murugesan, S.: 'Web engineering: an introduction', Multimedia, IEEE, 2001, 8, (1), pp. 14-18

[Glaser2007] Glaser, H., Millard, I.: 'Rkb explorer: Application and infrastructure'. Proceedings of Semantic Web Challenge, 2007

[Golbeck2006] Golbeck, J., Katz, Y., Krech, D., Mannes, A., Wang, T., Hendler, J.: 'PaperPuppy: Sniffing the Trail of Semantic Web Publications'. Semantic Web Challenge at ISWC-2006, 2006

[Grzonkowski2005] Grzonkowski, S., Gzella, A., Krawczyk, H., Kruk, S., Moyano, F., Woroniecki, T.: 'D-FOAF-Security Aspects in Distributed User Managment System'. IEEE International Conference on Technologies for Homeland Security and Safety (TEHOSS 2005), Gdansk, Poland, 2005

[Gueret2008] Gueret, C., Oren, E., Schlobach, S., Schut, M.: 'An Evolutionary perspective on approximate RDF query answering'. In: Proceedings of 7th International Semantic Web Conference (ISWC), Karlsruhe, Germany, 2008, pp. 215-228

[Guo2005] Guo, Y., Pan, Z., Heflin, J.: 'LUBM: A benchmark for OWL knowledge Base systems', Web Semantics: Science, Services and Agents on the World Wide Web, 2005, 3, (2-3), pp. 158-182

[Haase2004] Haase, P., Broekstra, J., Eberhart, A., Volz, R.: 'A comparison of RDF Query languages'. In: Proceedings of Third International Semantic Web Conference (ISWC), Hiroshima, Japan, 2004, pp. 502-517

[Halb2008] Halb W., Raimond Y., Hausenblas M.: .Building Linked Data For Both Humans and Machines'. In: WWW 2008 Workshop: Linked Data on the Web (LDOW2008), Beijing, China, 2008.

[Hart2008] Hart, J., Ridley, C., Taher, F., Sas, C., Dix, A.: 'Exploring the Facebook experience: a new approach to usability'. In: Proceedings of 5th Nordic conference on Human-computer interaction: building bridges, Lund, Sweden, 2008, pp. 471-474

[Harth2004] Harth, A.: 'SECO: mediation services for semantic Web data', IEEE Intelligent Systems, 2004, 19, (3), pp. 66-71

[Harth2005] Harth, A., Decker, S.: 'Optimized index structures for querying RDF from the web'. In: Proceedings of 3rd Latin American Web Congress, Buenos Aires, Argentina, 2005, pp. 71–80

[Hartig2009] Hartig, O., Muhleisen, H., Freyta, J.-C.: 'Linked Data for Building a Map of Researchers', http://www.semanticscripting.org/SFSW2009/challengesubmissions/submission2.pdf, [10.7.2009.]

[Hausenblas2007] Hausenblas, M., Slany, W., Ayers, D.: 'A performance and Scalability metric for virtual RDF graphs'. 3rd Workshop on Scripting for the Semantic Web, Innsbruck, Austria, 2007

[Hausenblas2008] Hausenblas, M., Halb, W., Raimond, Y.: 'Scripting User Contributed Interlinking'. 4th Workshop on Scripting for the Semantic Web (SFSW08), Tenerife, Spain, 1. - 6. June, 2008

[Hayes2004] Hayes, J., Gutierrez, C.: 'Bipartite Graphs as Intermediate Model for RDF'. In: Proceedings of Third International Semantic Web Conference (ISWC), Hiroshima, Japan, 2004, pp. 47-61

[Hassanzadeh2009] Hassanzadeh, O., Consens, M.: 'Linked Movie Data Base'. 2nd Linked Data on the Web Workshop (LDOW2009), Madrid, Spain, April 20th, 2009

[Heath2007] Heath, T., Motta, E.: 'Revyu. com: a Reviewing and Rating Site for the Web of Data', Lecture notes in computer science, 2007, 4825, pp. 895 – 902

[Heitmann2007a] Heitmann, B.: 'Transitioning web application frameworks towards the semantic web', Master's Thesis, Universität Karlsruhe, 2007

[Heitmann2007b] Heitmann, B., Oren, E.: 'Leveraging existing web frameworks for a SIOC explorer to browse online social communities'. ESWC Workshop on Scripting for the Semantic Web, 2007

[Herrera2006] Herrera, P., Bello, J., Widmer, G., Sandler, M., Celma, O., Vignoli, F., Pampalk, E., Cano, P., Pauws, S., Serra, X.: 'Simac: Semantic interaction with music audio contents'. 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies, 2006

[Hogenboom2008] Hogenboom, A., Milea, V., Frasincar, F., Kaymak, U.: 'Genetic Algorithms for RDF Query Path Optimization'. The First International Workshop on Nature Inspired Reasoning for the Semantic Web (NatuReS), Karlsruhe, Germany, October 26 – 30, 2008.

[Houben2003] Houben, G., Barna, P., Frasincar, F., Vdovjak, R.: 'HERA: Development of semantic web information systems', Lecture notes in computer science, 2003, pp. 529-538

[Horrocks2003] Horrocks, I., Patel-Schneider, P., Van Harmelen, F.: 'From SHIQ and RDF to OWL: The making of a web ontology language', Web Semantics: Science, Services and Agents on the World Wide Web, 2003, 1, (1), pp. 7-26

[Huhns1997] Huhns, M.N., Singh, M.P.: Ontologies for Agents. IEEE internet computing (1997)

[Hyvönen2005] Hyvönen, E., Mäkelä, E., Salminen, M., Valo, A., Viljanen, K., Saarela, S., Junnila, M., Kettula, S.: 'MuseumFinland—Finnish museums on the semantic web', Web Semantics: Science, Services and Agents on the World Wide Web, 2005, 3, (2-3), pp. 224-241

[Hyvonen2008] Hyvonen, E., Makela, E., Kauppinen, T., Alm, O., Kurki, J., Ruotsalo, T., Seppala, K., Takala, J., Puputti, K., Kuittinen, H.: 'Culture Sampo–A Collective Memory of Finnish Cultural Heritage on the Semantic Web 2.0', http://www.cs.vu.nl/~pmika/swc-2008/CultureSampo-hyvonen-et-al-culturesampo-swc-2008.final.pdf, [7.7.2009.]

[Jacyntho2002] Jacyntho, M., Schwabe, D., Rossi, G.: 'A Software Architecture for Structuring Complex Web Applications', Journal of Web Engineering, 2002, 1, (1), pp. 37-60

[Jain2006] Jain, A., Farkas, C.: 'Secure resource description framework: an access control model'. In: Proceedings of Eleventh ACM symposium on Access control models and technologies, 2006, pp. 121-129

[Jarrar2002] Jarrar, M., Meersman, R.: 'Formal ontology engineering in the dogma approach', Lecture notes in computer science, 2002, pp. 1238-1254

[Jarrar2003] Jarrar, M., Demey, J., Meersman, R.: 'On using conceptual data modeling for ontology engineering', Lecture notes in computer science, 2003, 2800, pp. 185-207

[Jazayeri2007] Jazayeri, M.: 'Some Trends in Web Application Development'. In: Proceedings of International Conference on Software Engineering, 2007 Future of Software Engineering, Minneapolis, Minnesota, USA, 2007, pp. 199-213

[Jena2010] Jena: 'A Semantic Web Framework for Java', http://jena.sourceforge.net, [26.01.2010.]

[Johnson1988] Johnson, R., Foote, B.: 'Designing Reusable Classes', Journal of Object-Oriented Programming, 1988, 1, (2), pp. 22-35

[Joo2009] Joo, J., Lee, S.: 'Adoption of the Semantic Web for overcoming technical limitations of knowledge management systems', Expert Systems With Applications, 2009, 36, (3P2), pp. 7318-7327

[Karvounarakis2002] Karvounarakis, G., Alexaki, S., Christophides, V.,s Plexousakis, D.,Scholl, M.: 'RQL: a declarative query language for RDF'. In: Proceedings of 11th international conference on World Wide Web, Honolulu, Hawaii, USA, 2002, pp. 592-603

[Keller2004] Keller, R., Berrios, D., Carvalho, R., Hall, D., Rich, S., Sturken, I., Swanson, K., Wolfe, S.: 'SemanticOrganizer: A customizable semantic repository for distributed NASA project teams', Lecture notes in computer science, 2004, pp. 767-781

[Kifer2005] Kifer, M., De Bruijn, J., Boley, H., Fensel, D.: 'A realistic architecture for the semantic web', Lecture notes in computer science, 2005, 3791, pp. 17-29

[Kim2007] Kim, H., Yang, S., Song, S., Breslin, J., Kim, H.: 'Tag mediated society with scot ontology'. 6th Internatinal Semantic Web Conference, Busan, Korea, 2007

[Kim2008] Kim, H., Passant, A., Breslin, J., Scerri, S., Decker, S.: 'Review and alignment of tag ontologies for semantically-linked data in collaborative tagging spaces'. In: Proceedings of, 2008, pp. 315-322

[Klischewski2006] Klischewski, R.: 'Migrating small governments' websites to the semantic web'. The Semantic Web meets eGovernment, , AAAI 2006, Stanford, CA, USA, 2006

[Klyne2009] Klyne, G., Carroll, J., McBride, B.: 'Resource description framework (RDF): Concepts and abstract syntax', www.w3.org/TR/rdf-concepts/ [17.04.2009.]

[Kochut2007] Kochut, K.J., Janik, M.: 'SPARQLeR: Extended Sparql for Semantic Association Discovery'. In: Proceedings of Proceedings of the 4th European conference on The Semantic Web: Research and Applications, Innsbruck, Austria, 2007, pp. 145-159

[Koivunen2005] Koivunen, M.-R. Annotea and semantic web supported collaboration. In: Proceedings of the User Aspects of the Semantic Web (UserSWeb) Workshop at the Second European Semantic Web Conference, ESWC 2005, Heraklion, Greece, pp. 5-17

[Kraines2007] Kraines, S., Guo, W., Kemper, B., Nakamura, Y.: 'EKOSS: A knowledge-user centered approach to knowledge sharing, discovery, and integration on the Semantic Web', Journal of information processing and management, 2007, 50, (6), pp. 322

[Krötzsch2006] Krötzsch, M., Vrandecic, D., Völkel, M.: 'Semantic mediawiki', Lecture Notes in Computer Science, 2006, 4273, pp. 934-942

[Kruk2004] Kruk, S.: 'Foaf-realm-control your friends' access to the resource'. In: Proceedings of 1st Workshop on Friend of a Friend, Social Networking And the (Semantic) Web (FOAF Galway), Galway, Ireland, 2004, pp. 1-9

[Lederer2000] Lederer, A.L., Maupin, D.J., Sena, M.P., Zhuang, Y.L.: ‘The Technology acceptance model and the World Wide Web’, Decis. Support Syst., 2000, 29, (3), pp. 269-282

[Lee2008] Lee, H., Jeun, I., Chun, K., Song, J.: ‘A New Anti-phishing Method in OpenID’. In: Proceedings of Second International Conference on Emerging Security Information, Systems and Technologies, SECURWARE '08., 2008, pp. 243-247

[Linked2009] 'Linked Data - Connect Distributed Data across the Web', http://www.linkeddata.org, [26.10.2009.]

[Liu2005] Liu, B., Hu, B.: ‘An evaluation of RDF storage systems for large data applications’. In: Proceedings of First International Conference on Semantics, Knowledge and Grid, 2005, pp. 59-61

[Maier2004] Maier, R.: ‘Knowledge management systems: Information and communication technologies for knowledge management’ (Springer-Verlag New York Inc, 2004.)

[Manjunath2008] Manjunath, G., Sayers, C., Reynolds, D., KS, V., Mohalik, S., Badrinath, R., Recker, J., Mesarina, M.: 'Semantic Views for Controlled Access to the Semantic Web', http://www.hpl.hp.com/techreports/2008/HPL-2008-15.html?mtxs=rss-hpl-tr, [26.10.2009.]

[McGuinness2009] McGuinness, D., Van Harmelen, F.: 'Owl web ontology language overview', http://www.w3.org/TR/owl-features/, [17.04.2009.]

[Mesbah2008] Mesbah, A., Deursen, A.v.: ‘A component- and push-based architectural style for ajax applications’, J. Syst. Softw., 2008, 81, (12), pp. 2194-2209

[Merriam2009] Merriam-Webster, I.: 'Merriam-Webster Online', http://www.merriam-webster.com/, [28.8.2009.]

[Microformat2009] 'Microformats', http://microformats.org/, [4.5.2009.]

[Mika2005] Mika, P.: ‘Flink: Semantic web technology for the extraction and analysis of social networks’, Web Semantics: Science, Services and Agents on the World Wide Web, 2005, 3, (2-3), pp. 211-223

[Motik2009] Motik, B., Grau, B., Wu, Z., Fokoue, A., Lutz, C.: 'OWL 2 Web Ontology Language: Profiles. W3C Working Draft, 11 April 2008', http://www.w3.org/TR/2008/WD-owl2-profiles-20081202/, [17.04.2009.]

[Mulvenna2000] Mulvenna, M., Anand, S., Buchner, A.: ‘Personalization on the net using Web mining: introduction’, Communications of the ACM, 2000, 43, (8), pp. 122-125

[Murugesan2007] Murugesan, S.: ‘Web Application Development: Challenges and the Role of Web Engineering’: ‘Web engineering: modelling and implementing web applications’ (Springer, 2007), pp. 7-32

[Mycroft2008] Mycroft, A.: ‘Controlling Control Flow in Web Applications’, Electronic Notes in Theoretical Computer Science (ENTCS), 2008, 200, (3), pp. 119-131

[Notoatmodjo2007] Notoatmodjo, G.: ‘Exploring the ‘Weakest Link’: A Study of Personal Password Security’ (The University of Auckland, New Zealand, 2007. 2007)

[Nowack2006] Nowack, B.: ‘CONFOTO: Browsing and annotating conference photos on the Semantic web’, Web Semantics: Science, Services and Agents on the World Wide Web, 2006, 4, (4), pp. 263-266

[Nowack2008] Nowack, B.: 'paggr - Smart Data Portals', http://www.cs.vu.nl/~pmika/swc-2008/paggr-bnowack_paggr_2008_10_01.pdf, [8.7.2009.]

[Nowack2009] Nowack, B.: 'SPARQLBot: The Semantic Web Command Line (Scripting Challenge Submission)', http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-368/paper7.pdf, [10.7.2009.]

[OpenID2009] 'OpenID', http://openid.net/, [20.12.2009.]

[Oren2006] Oren, E., Delbru, R., Decker, S.: ‘Extending faceted navigation for RDF data’, Lecture notes in computer science, 2006, 4273, pp. 559-572

[Oren2007] Oren, E., Delbru, R., Gerke, S., Haller, A., Decker, S.: ‘ActiveRDF: Object-oriented semantic web programming’. In: Proceedings of, 2007, pp. 817-824

[Oren2007b] Oren, E., Haller, A., Hauswirth, M., Heitmann, B., Decker, S., Mesnage, C.: ‘A flexible integration framework for Semantic Web 2.0 applications’, IEEE Software, 2007, 24, (5), pp. 64-71

[Oren2008] Oren, E.: ‘Algorithms and Components for Application Development on the Semantic Web’, National University of Ireland, 2008

[O’Reilly2009] O’Reilly, T.: 'What is Web 2.0: Design patterns and business models for the next generation of software', http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, [14.06.2009.]

[Pashalidis2003] Pashalidis, A., Mitchell, C.: ‘A taxonomy of single sign-on systems’, Lecture notes in computer science, 2003, pp. 249-264

[Passant2008b] Passant, A., Laublet, P.: ‘Meaning Of A Tag: A collaborative approach to bridge the gap between tagging and Linked Data’. WWW 2008 Workshop Linked Data on the Web (LDOW2008), Beijing, China, 2008

[Passant2009a] Passant, A.: 'A user-friendly interface to browse and find DOAP project with doap: store', http://ceur-ws.org/Vol-248/challenge1.pdf, [8.7.2009.]

[Passant2009b] Passant, A.: 'FOAFMap: Web2. 0 meets the Semantic Web', http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-181/challenge1.pdf, [8.7.2009.]

[Passant2009c] Passant, A.: 'Linked Data tagging with LODr', http://www.cs.vu.nl/~pmika/swc-2008/LODr-iswc-challenge-apassant-final-2.pdf, [7.7.2009.]

[Paulson2005] Paulson, L.D.: ‘Building rich web applications with Ajax’, Computer, 2005, 38, (10), pp. 14-17

[Perez2006] Perez, J., Arenas, M., Gutierrez, C.: ‘Semantics and Complexity of SPARQL’, Lecture notes in computer science, 2006, 4273, pp. 30 – 43

[Perry1992] Perry, D., Wolf, A.: ‘Foundations for the study of software architecture’, ACM SIGSOFT Software Engineering Notes, 1992, 17, (4), pp. 40-52

[Powers2003] Powers, S.: ‘Practical rdf’ (O'Reilly & Associates, Inc. Sebastopol, CA, USA, 2003.)

[Pressman2000] Pressman, R.S.: ‘What a tangled Web we weave [Web engineering]’, Software, IEEE, 2000, 17, (1), pp. 18-21

[Prud’Hommeaux2009a] Prud’hommeaux, E.: 'SPASQL: SPARQL Support In MySQL', http://xtech06.usefulinc.com/schedule/paper/156, [16.04.2009.]

[Prud’Hommeaux2009b] Prud’Hommeaux, E., Seaborne, A.: 'SPARQL query language for RDF', http://www.w3.org/TR/rdf-sparql-query/, [17.04.2009.]

[Raimond2008] Raimond, Y., Sutton, C., Sandler, M.: ‘Automatic interlinking of music datasets on the semantic web’. Linked Data on the Web Workshop (LDOW2008), Beijing, China, 2008

[Ramagem2004] Ramagem, D., Margerin, B., Kendall, J.: ‘AnnoTerra: building an integrated earth science resource using semantic Web technologies’, IEEE Intelligent Systems, 2004, 19, (3), pp. 48-57

[Raskin2005] Raskin, R., Pan, M.: ‘Knowledge representation in the semantic web for Earth and environmental terminology (SWEET)’, Computers and Geosciences, 2005, 31, (9), pp. 1119-1125

[Reddivari2007] Reddivari, P., Finin, T., Joshi, A.: 'Policy based access control for a rdf store'. In: Proceedings of IJCAI-07 Workshop on Semantic Web for Collaborative Knowledge Acquisition, 2007, pp. 78–83

[Reif2007] Reif, G., Laube, G., Moller, K., Gall, H.: 'SemClip-Overcoming the Semantic Gap Between Desktop Applications', 5th Semantic Web Challenge at the ISWC2007, Busan, South Korea, 2007

[Reif2005] Reif G., WEESA – Web Engineering for Semantic Web Applications. Ph.D. thesis, Technische Universit¨at Wien, 2005.

[Resncick1997] Resnick, P., Varian, H.: 'Recommender Systems', Communications of the ACM, 1997, 40, (3), pp. 56-58

[Rovan2006] Rovan, L., Baranovic, M.: 'Migrating web-based applications into semantic web'. In: Proceedings of 28th International Conference on Information Technology Interfaces, Cavtat, Croatia, 2006, pp. 159-164

[Rovan2008a] Rovan, L: Realizing Semantic Web Portal Using Available Semantic Web Technologies and Tools. In: Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC2008), 2008, Oct 28, Karlsruhe, Germany.

[Rovan2008b] Rovan, L., Nizetic, I.: 'Extending Java Web Applications for Semantic Web'. In: Proceedings of 30th International Conference on Information Technology Interfaces, Cavtat, Croatia, 2008, pp. 289-294

[Sahoo2009] Sahoo, S., Halb, W., Hellmann, S., Idehen, K., Thibodeau Jr, T., Auer, S., Sequeda, J., Ezzat, A.: 'A Survey of Current Approaches for Mapping of Relational Databases to RDF', W3C Incubator Group Report January, 2009

[Samar1999] Samar, V.: 'Single sign-on using cookies for Web applications'. In: Proceedings of IEEE 8th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, (WET ICE'99), 1999, pp. 158-163

[Sauermann2009] Sauermann, L., Cyganiak, R.: 'Cool URIs for the Semantic Web',http://www.w3.org/TR/cooluris/, [24.5.2009.]

[Scharl2009] Scharl, A., Weichselbraun, A., Hubmann-Haidvogel, A., Stern, H., Wohlgenannt, G., Zibold, D.: 'Media Watch on Climate Change: Building and Visualizing Contextualized Information Spaces', http://www.cs.vu.nl/~pmika/swc-2007/ MediaWatch.pdf, [6.7.2009.]

[Schmidt2008] Schmidt, M., Hornung, T., Kuchlin, N., Lausen, G., Pinkel, C.: 'An Experimental Comparison of RDF Data Management

Approaches in a SPARQL Benchmark Scenario'. In: Proceedings of 7th International Semantic Web Conference (ISWC), Karlsruhe, Germany, 2008, pp. 82-97

[Schmidt2007] Schmidt, K., Stojanovic, L., Stojanovic, N., Thomas, S.: 'On enriching ajax with semantics: The web personalization use case', Lecture notes in computer science, 2007, 4519, pp. 686-700

[Schneidewind1987] Schneidewind, N.: 'The state of software maintenance', IEEE Transactions on Software Engineering, 1987, pp. 303-310

[Schreiber2006] Schreiber, G., Amin, A., Van Assem, M., de Boer, V., Hardman, L., Hildebrand, M., Hollink, L., Huang, Z., van Kersen, J., de Niet, M.: 'Multimedian e-culture demonstrator', Lecture notes in computer science, 2006, 4273, pp. 951

[Seaborne2009a] Seaborne, A.: 'Rdql-a query language for rdf', http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/, [30.03.2009.]

[Seaborne2009b] Seaborne, A., Manjunath, G.: 'SPARQL/Update A language for updating RDF graphs', ttp://jena.hpl.hp.com/~afs/SPARQL-Update.html, [03.04.2009]

[SWCSUC2009] 'Semantic Web Case Studies and Use Cases ', http://www.w3.org/2001/sw/sweo/public/UseCases/, [25.05.2009.]

[Servant2008] Servant, F.: 'Linking Enterprise Data'. Linked Data on the Web (LDOW2008), Beijing, China, 2008.

[Shadbolt2004] Shadbolt, N., Gibbins, N., Glaser, H., Harris, S., Schraefel, M.: 'CSAKTive Space, or how we learned to stop worrying and love the semantic Web', IEEE Intelligent Systems, 2004, 19, (3), pp. 41-47

[Shadbolt2006] Shadbolt, N., Hall, W., Berners-Lee, T.: 'The semantic web revisited', IEEE Intelligent Systems, 2006, 21, (3), pp. 96-101

[Singh2002] Singh, I., Stearns, B., Johnson, M. and the Enterprise Team: 'Designing Enterprise Applications with the J2EE Platform' (Addison-Wesley Professional, 2002.)

[Sintek2002] Sintek, M., Decker, S.: 'TRIPLE - A Query, Inference, and Transformation Language for the Semantic Web'. In: Proceedings of First International Semantic Web Conference on The Semantic Web, Sardinia, Italy, 2002, pp. 364 – 378

[Smith2004] Smith, M.A.: 'Portals: toward an application framework for interoperability', Communications of the ACM, 2004, 47, (10), pp. 93-97

[Spyns2002] Spyns, P., Meersman, R., Jarrar, M.: 'Data modelling versus Ontology engineering', ACM SIGMOD Record, 2002, 31, (4), pp. 12-17

[Staab2001] Staab, S., Maedche, A. and Handschuh, S., An Annotation Framework for the Semantic Web. Proc. 1 Int. Workshop on MultiMedia Annotaion, Tokyo, 2001

[Staab2004] Staab, S., Studer, R.: 'Handbook on ontologies' (Springer, 2004.)

[Stegers2006] Stegers, R., Fekkes, P., Stuckenschmidt, H.: 'MusiDB A personalized search engine for music', Web Semantics: Science, Services and Agents on the World Wide Web, 2006, 4, (4), pp. 267-275

[Stocker2008] Stocker, M., Seaborne, A., Bernstein, A., Kiefer, C., Reynolds, D.: 'SPARQL basic graph pattern optimization using selectivity estimation'. In: Proceedings of 17th international conference on World Wide Web Beijing, China, 2008, pp. 595-604

[Stojanovic2002] Stojanovic, L., Stojanovic, N., Volz, R.: 'Migrating data-intensive web sites into the semantic web'. In: Proceedings of ACM Symposium on Applied Computing SAC-02, Madrid, Spain, 2002, pp. 1100-1107

[Studer1998] Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: Principles and methods. Data & Knowledge Engineering 25 (1998) 1-2, 161-197.

[Sumsiripong2007] Sumsiripong, E., Maleewong, K., Ungrangsi, R., Anutariya, C.: 'WWWatch: Watching the World in Different Views with Semantic Web'. International Semantic Web Conference, Busan, Korea, 2007

[Suriadi2009] Suriadi, S., Foo, E., Jøsang, A.: 'A user-centric federated single sign-on system', Journal of Network and Computer Applications, 2009, 32, (2), pp. 388-401

[Tanasescu2006] Tanasescu, V., Gugliotta, A., Domingue, J., Villaras, L., Davies, R., Rowlatt, M., Richardson, M., Stincic, S.: 'Spatial Integration of Semantic Web Services: the e-Merges Approach'. International Semantic Web Conference, Athens, GA, USA, 2006

[Tazzoli2004] Tazzoli, R., Castagna, P., Campanini, S.: 'Towards a semantic wiki wiki web'. International Semantic Web Conference, Hiroshima, Japan, 2004

[Tirmizi2008] Tirmizi, S., Sequeda, J., Miranker, D.: 'Translating SQL applications to the semantic web', Lecture notes in computer science, 2008, 5181, pp. 450-464

[Uschold1998] Uschold, M., Healy, M., Williamson, K., Clark, P., Woods, S.: 'Ontology reuse and application', Formal Ontology in Information Systems, 1998, pp. 179–192

[Volkel2006] Volkel, M., Sure, Y.: 'RDFReactor–from ontologies to programmatic data access'. Jena User Conference, HP Bristol, Great Britain, 2006

[Volz2009] Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: ‘Silk–A Link Discovery Framework for the Web of Data’. 2nd Linked Data on the Web Workshop (LDOW2009), Madrid, Spain, 20th April, 2009

[Vossen2007] Vossen, G., Hagemann, S.: ‘Unleashing web 2.0: from concepts to creativity’, Ubiquity, 2007, 8, (50)

[Vranešić2009] Vranešić, H., Rovan, L.: ‘Ontology-based Data Warehouse Development Process’. In: Proceedings of 31st International Conference on Information Technology Interfaces, Cavtat, Croatia, 2009, pp. 205-210

[Wang2006] Wang, C., Lu, J., Zhang, G.: ‘Integration of ontology data through learning instance matching’. In: Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence 2006, pp. 536-539

[Williams2009] Williams, G.: 'MT-Redland: An RDF Storage Backend for Movable Type', http://www.semanticscripting.org/SFSW2005/papers/Williams-MT-Redland.pdf, [8.7.2009.]

[Wiederhold2006] Wiederhold, G.: ‘What is your software worth?’, Communications of the ACM, 2006, 49, (9), pp. 65-75

[Wu2006a] Wu, H., Cheng, G., Qu, Y.: ‘Falcon-S: An ontology-based approach to searching objects and images in the Soccer domain’, Supplemental Proceedings of ISWC, 2006

[Wu2000] Wu, A., Wang, H., Wilkins, D.: ‘Performance Comparison of Alternative Solutions For Web-To-Database Applications–’. In: Proceedings of Southern Conference on Computing, Hattiesburg, Mississippi, USA, 2000, pp. 26-28

[Wu2006b] Wu, Z., Chen, H., Wang, H., Wang, Y., Mao, Y., Tang, J., Zhou, C.: ‘Dartgrid: a semantic web toolkit for integrating heterogeneous relational databases’. Semantic Web Challenge at 4th International Semantic Web Conference (ISWC 2006), Athens, USA, 2006

[Zhang2009] Zhang, H., Fu, L., Wang, H., Zhu, H., Wang, Y., Yu, Y.: 'EachWiki: Suggest to Be an Easy-To-Edit Wiki Interface for Everyone', http://ceur-ws.org/Vol-295/paper05.pdf, [6.7.2009.]

[Ziemer2009] Ziemer, S.: 'Web Engineering', http://csgsc.idi.ntnu.no/2004/data/svenz/WE_csgsc.pdf, [1.7.2009.]

Popis kratica

Ajax	*Asynchronus JavaScript and XML*
API	*Application programming interface*
CGI	*Common Gateway Interface*
CSS	*Cascading Style Sheets*
CURIE	*Compact URI*
DAML	*DARPA Agent Markup Language*
DOI	*Degree of interest*
DOI	*Digital Object Identifier*
DOM	*Document Object Model*
EJB	*Enterprise JavaBean*
E-R	*Entity Relationship*
FOAF	*Friend-of-a-friend*
GRDDL	*Gleaning Resource Descriptions from Dialects of Languages*
HTML	*HyperText Markup Language*
HTTP	*Hypertext Transfer Protocol*
ISBN	*International Standard Book Number*
IRI	*International Resource Identifier*
JEE	*Java Enterprise Edition*
JSF	*JavaServer Faces*
JSP	*Java Server Pages*
MIT	*Masachussets Institute of Technology*
MVC	*Model-View-Controller*
MOAT	*Meaning of a Tag*
OEM	*Object Exchange Model*
ORM	*Object – Relational Mapping*
ORM	*Object Role Modeling*
OWL	*Ontology Web Language*
RDF	*Resource Description Framework*
RDFS	*RDF Schema*
RDQL	*RDF Data Query Language*
REST	*Representational State Transfer*
RIA	*Rich Internet Applications*
RIF	*Rule Interchange Format*
RSS	*RDF Site Summary*
RQL	*RDF query language*
SKOS	*Simple-knowledge-organisation-system-vocabulary*
SOAP	*Simple Object Access Protocol*
SPARQL	*SPARQL Query Language for RDF*
SPARUL	*SPARQL/Update*
SPROT	*SPARQL Protocol*
SQL	*Structured Query Language*
SWRC	*Semantic Web for Research Communities*
XML	*Extensible Markup Language*
XPath	*XML Path Language*
XSLT	*Extensible Stylesheet Language Transformations*
UML	*Unified Modelling Language*
URI	*Uniform Resource Identifier*
URL	*Uniform Resource Locator*
URN	*Uniform Resource Name*
W3C	*World Wide Web Consortium*
WSDL	*Web Services Description Language*

Sažetak

Semantički web je zamišljen kao rezultat evolucije sadašnjeg weba. Danas je web izrazito dinamičko okruženje i platforma za sofisticirane aplikacije visoke razine složenosti. U disertaciji je predložen mogući način prilagodbe postojećih Java web aplikacija temeljenih na obrascu model-izgled-kontroler semantičkom webu. Ideje i načela na kojima se temelji web i semantički web sasvim su različita, pa samim time i realizacije u takvim okruženjima imaju međusobna odstupanja. U disertaciji je dan pregled područja weba, razvoja Java web aplikacija, te područja semantičkog weba. Također, obavljena je opsežna analiza semantičkih web aplikacija na temelju koje su definirane osnovne funkcionalnosti i kategorije semantičkih web aplikacija. Za svaku prepoznatu kategoriju semantičkih web aplikacija predložena je arhitektura. Elementi arhitekture odabrani su u skladu s provedenom analizom tehnologija semantičkog weba. Uočene su osnovne sličnosti, odnosno razlike u razvojnim procesima web aplikacija i semantičkih web aplikacija. Koristeći rezultate provedene analize i obavljenog pregleda područja predložen je proces prilagodbe postojećih web aplikacija semantičkom webu. Proces prilagodbe popraćen je metodologijom koja propisuje slijed aktivnosti koje se moraju obaviti u svrhu provođenja prilagodbe. Za predloženi proces prilagodbe dana je ocjena njegove primjenjivosti na temelju verifikacije na realnim primjerima.

Ključne riječi

- semantički web
- web aplikacija
- MVC
- Java
- prilagodba
- metodologija
- arhitektura
- upravljanje znanjem
- integracija
- označavanje sadržaja

Adjustment of Java web applications based on Model-View-Controller pattern for Semantic Web

Abstract

Today, the Web is a dynamic environment and a platform for sophisticated high-level business applications. The Semantic Web is an extension of the current Web in which information is given well-defined meaning, better enabling computers and people to work in cooperation. Although Semantic Web offers more to the users, evolution of the existing web applications into the semantically enriched ones, still remains an under researched topic. The research presented in this dissertation has proposed a methodology for adapting existing Java web applications based on the model-view-controller pattern to the Semantic Web. Ideas and principles underlying the Semantic Web and the Web as we know it today are rather different. Therefore the realization in such environments have mutual differences. The dissertation gives an overview of the both fields, specifically concentrating on the Java Web applications. Based on thorough analysis of the Semantic Web applications, basic functionality and dissimilar categories pertaining to Semantic Web applications were defined. Separate architecture has been proposed for each separate category. Afterwards, basic similarities and dissimilarities between application development processes have been observed. Using the results obtained from aforementioned analysis, a process for adapting existing Web applications to the Semantic Web has been proposed. Finally, an applicability estimate of proposed methodology is given based on verification on real cases.

Keywords

- semantic web
- web application
- MVC
- Java
- adapting
- methodology
- architecture
- knowledge management
- integration
- tagging

Životopis

Lidia Rovan rođena je 20. lipnja 1980. godine u Splitu. Nakon završene 3. gimnazije u Splitu upisuje studij na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu. Diplomirala je u lipnju 2003. godine na studiju Računarstva s odličnim uspjehom.

Od kolovoza 2003. do lipnja 2004. godine zaposlena je kao tehnički konzultant u tvrtki TIS-OIS. U listopadu 2003. godine upisuje poslijediplomski studij na Fakultetu elektrotehnike i računarstva u Zagrebu. Od lipnja 2004. godine zaposlena je na Zavodu za primijenjenu matematiku kao znanstveni novak (u međuvremenu se Zavod izdvojio u Zavod za primijenjeno računarstvo).

U okviru nastavnih djelatnosti na Fakultetu elektrotehnike i računarstva sudjelovala je u nastavi iz predmeta "Programiranje i programsko inženjerstvo", “Uvod u baze podataka”, "Baze podataka", "Sustavi baza podataka" i “Napredni modeli i baze podataka”. Također, sudjeluje u radu sa studentima kroz seminare, projekte i izrade diplomskih radova.

Znanstveno se bavi područjem programskog inženjerstva i semantičkog weba. Iz područja semantičkog weba objavila je 4 rada na međunarodnim skupovima.

Biography

Lidia Rovan was born on June 20th, 1980. in Split. After graduating from secondary school in Split, she started the undergraduate study program at the Faculty of Electrical Engineering and Computing, University of Zagreb where she graduated in 2003.

Since August 2003. till June 2004. she worked as technical consultant at TIS-OIS. In October 2003. she enrolled the Postgraduate Study at the Faculty of Electrical Engineering and Computing. Since June 2004 she has been employed at the Department of Applied Computing of the Faculty of Electrical Engineering and Computing as a research assistant.

At the Faculty of Electrical Engineering and Computing she participated as teaching assistant on courses "Programming and Program Engineering", "Introduction to Databases", "Databases", "Database Systems" and "Advanced Databases".

Her scientific interest are software engineering and the semantic web. She is the author or co-author of a number of scientific, professional and educational publications in her fields of interest.

Dodatak

Opisi analiziranih aplikacija

Seco

Opis: Seco [Harth2004] je prva aplikacija od tima koji je kasnije razvio još niz semantičkih web aplikacija (http://www.seco.tkk.fi/). Seco se sastoji od tri osnovna dijela: web robot (eng. crawler) koji prikuplja podatke u RDF-u iz cijelog weba, medijatora koji povezuju tako prikupljene podatke i korisničkog sučelja. Integriranim Seco podacima mogu pristupati i druge aplikacije. Prikupljeni podaci opisuju osobe i razne vijesti, tako da su dva podržana vokabulara FOAF (opis ljudi) i RSS 1.0. (opis vijesti). Integrirani podaci se pohranjuju u vlastitom spremištu podataka.

Slučaj uporabe: Korisniku se prilagođeno prikazuju integrirani podaci, nad kojima onda može obavljati pregledavanje ili pretraživanje po ključnim riječima.

Funkcionalnosti:

- integracija
- pretraživanje

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

AnnoTerra

Opis: AnnoTerra [Ramagem2004] je aplikacija koja se temelji na podacima organizacije NASA iz domene "znanost zemlje". Baza znanja na kojoj se aplikacija temelji formirana je iz tri izvora podataka. Nove informacije se stalno crpe iz RSS feed-a: vijesti koje dolaze iz opservacije zemlje (*NASA's Earth Observatory - http://earthobservatory.nasa.gov*). Budući da su takve informacije pisane prirodnim jezikom, prilikom njihovog prikupljanja izvlače se potencijalne ključne riječi. Pretraživanje koje aplikacija podržava izvodi se na tako prepoznatim ključnim riječima. Da bi se moglo podržati istovremeno pretraživanje više izvora podataka izgrađena je OWL ontologija koja objedinjuje (spaja) različite izvore podataka. Osim spomenutog izvora vijesti, dodatno se integriraju izvori GCMD *(Global Change Master Directory)* i ECHO *(Earth Observing System ClearingHouse)*, koji su zapravo katalozi (popisi) dostupnih izvora podataka. Ontologija je izgrađena ne temelju rječnika za GCMD i ECHO izvore podataka, odnosno pojmova tih rječnika, a obuhvaća i poveznice između pojmova ta dva rječnika. Na takav način omogućeno je istovremeno pretraživanje svih izvora podataka strukture koju propisuje GCMD ili ECHO ontologija. Koriste se i mogućnosti zaključivanja nad podacima pohranjenim u ontologiji.

Slučaj uporabe: Na temelju unesenih ključnih riječi aplikacija nudi prikaz relevantnih vijesti i poveznice na izvore podataka iz GCMD i ECHO popisa koji su na neki način povezani uz vijest. Iskusnim korisnicima na ovaj se način pruža sistematizirani pregled, a neiskusnima uvid u dodatne informacije za koje nisu ni znali da su također relevantne za traženi upit.

Funkcionalnosti:

- integracija
- pretraživanje uz zaključivanje

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

CS AKTive Space

Opis: CS AKTive Space [Shadbolt2004] pruža mogućnosti pregledavanja i otkrivanja novih informacija vezanih uz istraživanja iz područja računarstva. Podaci se prikupljaju raznim metodama, ovisno o formatu u kojem se nalaze, s javno dostupnih stranica institucija (prikupljanjem podataka iz sadržaja stranice), iz baza podataka, različitih dokumenata, i sl. Kao posebna mogućnost implementirano je da se svi geoprostorni podaci prikazuju na karti, pa korisnik ima mogućnost i ugodnijeg pretraživanja takvih podataka po karti. Također, postoji mogućnost i detaljnijeg pregleda istraživača, njegovih osobnih podataka, publikacija i istraživačke grupe kojoj pripada. Osim samog pretraživanja, postoji i mogućnost pregledavanja sadržaja pohranjenog u repozitorij (korištenjem navigacije).

Slučaj uporabe: Korisnik može pretraživati i pregledavati podatke po tri kriterija: regiji (pretraživanje na karti), područje istraživanja i istraživači.

Funkcionalnosti:

- integracija
- pretraživanje (vizualno)

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

SWEET

Opis: Sweet [Raskin2005] aplikacija se temelji na ontologijama koje opisuju pojmove "zemlje i okoliša". Pri izradi aplikacije korišteni su i vokabulari kreirani u AnnoTerra projektu. Na osnovu kreiranih ontologija (napravilo ih se više, od kojih neke čine facete) implementirana je aplikacija koja pruža podršku pretraživanju, odnosno na osnovu postavljenog upita vraća sve relevantne pojmove (npr. sinonime, uži pojam, općenitiji pojam). Osnovni cilj projekta je bilo stvaranje samih ontologija, a implementacija pretraživanja je obavljena u svrhu prototipa i verifikacije valjanosti ontologija. Za traženje sličnosti pojmova korišten je LSA algoritam, a nije poznato jesu li korištene metode zaključivanja iz područja ontologija. Aplikacija radi samo na pojmovima ontologije, ne koristi nikakve dodatne izvore podataka (instance). Za pohranu podataka aplikacije ne koristi semantički repozitorij (vjerojatno iz razloga što je nastala 2003.), već koristi proširenja relacijske baze podataka.

Slučaj uporabe: Pretraživanje pojmova iz područja "zemlje i okoliša". Kao rezultat korisnik dobije skup relevantnih pojmova.

Funkcionalnosti:

- pretraživanje

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene koja semantičke tehnologije koristi u svrhu boljeg pretraživanja
- **procjenitelj B:** aplikacija posebne primjene koja semantičke tehnologije koristi u svrhu boljeg pretraživanja

MusiDB

Opis: MusiDB [Stegers2006] je personalizirani sustav koji nudi preporuke u području glazbe na temelju otkrivenih preferencija korisnika. Aplikacija radi na dva integrirana izvora: Amazon i MusicBrainz. Da bi se mogle ostvariti preporuke od korisnika se traži (u određenim intervalima) da popuni anketu u kojoj iznese neke osnovne preferencije o stilovima glazbe. Dalje, svi korisnici ocjenjuju izvođače i pjesme, te se i tako dodijeljene ocjene uzimaju u obzir kod generiranja preporuka. Sam taj sustav preporuka je samo dio cijele aplikacije, koja u trenutku pisanja radova koji su dostupni o aplikaciji nije bila dovršena. Npr. dodatna predviđena funkcionalnost je pretraživanje, kao i pristup drugim izvorima podataka osim spomenuta dva.

Slučaj uporabe: Pri korištenju aplikacije korisnika se povremeno pita da odgovori na pitanja ankete ili da ocijeni sadržaj. Kasnije, na temelju unesenih podataka (kada ih bude dovoljno) korisniku se predloži sadržaj koji bi ga mogao zanimati.

Funkcionalnosti:

- integracija
- personalizacija
- preporučen sadržaj

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

SemanticOrganizer

Opis: SemanticOrganizer [Keller2004] je aplikacija koja je zamišljena kao potpora kolaboraciji više različitih, prostorno dislociranih timova NASA-e. Okosnica sustava je repozitorij u kojem se pohranjuju semantički strukturirani podaci, a sam sustav je zamišljen kao pristupna točka putem koje su dostupne sve informacije vezane za projekte koji su u tijeku. U repozitorij zaposlenici mogu pohraniti dokumente, slike i sav ostali relevantan multimedijalni sadržaj, ali i meta-podatke o samim pohranjenim objektima. Sustav omogućuje intuitivan pristup svim korisnicima (pohranjuje veze između sadržaja, pa te iste kasnije korisniku nudi) premda oni dolaze iz različitih sredina, pa se očekuje razilaženje u terminologiji i općenito u shvaćanju istih pojmova. Da bi se omogućio takav pristup, a podržale i dodatne funkcionalnosti implementirano je i zaključivanje nad repozitorijom (koristi se i kod pretraživanja). Postoji i dodatna funkcionalnost automatskog očitavanja i pohrane u sustav rezultata mjerenja sa strojeva na terenu.

Slučaj uporabe: Prvi korisnici sustava su bili Early Microbial Ecosystems Research Group (EMERG), odnosno internacionalni tim od preko 35 ljudi različitih struka smještenih u 8 različitih institucija. Tim je provodio zajedno istraživanje koje se sastojalo od brojnih eksperimenata provedenih na terenima. Sustav je služio kao mjesto na kojem se pohranjuju podaci o testiranjima kao i rezultati, te su svi sudionici imali trenutan pristup svim podacima.

Funkcionalnosti:

- integracija
- pretraživanje uz zaključivanje
- označavanje
- semantički repozitorij

Odabrana kategorija:

- **procjenitelj A:** semantički sustav za upravljanje znanjem
- **procjenitelj B:** semantički sustav za upravljanje znanjem

PlatypusWiki

Opis: PlatypusWiki [Tazzoli2004] aplikacija posjeduje sve karakteristike klasične Wiki Wiki Web aplikacije, ali ima i dodatne funkcionalnosti potpomognute semantičkim web tehnologijama. Semantičke tehnologije, konkretno RDF i OWL koriste se za opise veza između stranica, a dodatno svaka HTML stranica ima opis njenog sadržaja i u RDF-u. PlatypusWiki može koristiti jedna osoba, zajednica ili radne skupine. Mogu kroz zajedničku suradnju kreirati i dijeliti informacije, vokabulare, ontologije, ili općenito, bazu znanja.

Slučaj uporabe: Korisnik može unositi sadržaj određene teme (wiki stranica). Suradnja se odražava u tome što drugi korisnici mogu nadopunjavati takve sadržaje, graditi nove kao i umetati veze među njima.

Funkcionalnosti:

- označavanje
- semantički repozitorij

Odabrana kategorija:

- **procjenitelj A:** semantički sustav za upravljanje znanjem
- **procjenitelj B:** semantički sustav za upravljanje znanjem

MuseumFinland

Opis: MuseumFinland [Hyvönen2005] aplikacija je još jedan projekt od Seco istraživačke grupe. Aplikacija se zasniva na čak 7 ontologija, a sve su iz područja kulturne baštine. Podaci iz različitih izvora podataka (baza podataka različitih muzeja) su sakupljeni u samo jednu bazu znanja nad kojoj radi aplikacija. Radi se o personaliziranoj aplikaciji koja na temelju profila korisnika i metrika sličnosti generira preporučen sadržaj. Aplikacija podržava dva načina pretrage: multi-facetno (facete: umjetnina, materijal, autor, lokacija, vrijeme stvaranja, korisnik umjetnine, lokacije gdje se koristila, situacija korištenja, zbirka) i po ključnim riječima. Facetno pretraživanje ima brojne prednosti kada korisnik ne zna točno što traži, dok u drugom slučaju kada korisnik traži nešto specifično ipak pretraživanje po ključnim riječima puno bolje odgovara.

Slučaj uporabe: Korisnik može vizualno pretraživati bazu znanja, korištenjem faceta, ili može pretraživati bazu znanja po ključnim riječima.

Funkcionalnosti:

- integracija
- personalizacija
- preporučen sadržaj
- pretraživanje (po ključnim riječima i facetno)
- semantički repozitorij

URL aplikacije: http://www.museosuomi.fi/

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

Semantic portal of international affairs (SPIA)

Opis: SPIA [Contreras2004] aplikacija je zamišljena kao nadogradnja postojećeg portala koji objavljuje vijesti vezane uz vanjske poslove republike Španjolske. Aplikacija integrira više izvora podataka, te se pri agregiranju tih izvora podataka oni automatski označavaju (korištenjem algoritama za analizu prirodnog jezika). Nadogradnja se sastoji od implementacije poboljšanog pretraživanja. Pretraživati je moguće prirodnim jezikom ili kroz sučelje s ponuđenim poljima po kojima se definira uvjet pretrage. Podaci nad kojima aplikacija radi skupljaju se iz nekoliko izvora, a pomoću posebnog modula izvlače se informacije iz dostupnih izvora i transformiraju u semantičku bazu znanja. Portal prikazuje izvorni sadržaj označen pojmovima iz ontologije. Sadržaj se može i pregledavati, a poveznice između sadržaja generirane su u skladu sa semantički označenim sadržajem.

Slučaj uporabe: Korisnik može pretraživati upisom ključnih riječi, ali za razliku od tradicionalnih pretraživača kao rezultat neće dobiti tekst s pronađenim riječima nego instance koje čine odgovor. Postoje poveznice koje onda sa tih instanci mogu odvesti korisnika do izvornog dokumenta.

Funkcionalnosti:

- integracija
- pretraživanje
- semantički repozitorij

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

Flink

Opis: Flink [Mika2005] je vrlo jednostavna aplikacija i osnovni cilj za njeno stvaranje je bila demonstracija semantičkih web tehnologija. Svrha aplikacije je prikazivanje radova i društvene povezanosti istraživača semantičkog weba. Navigacija unutar aplikacije se obavlja po društvenoj mreži istraživača. Društvena mreža se prikazuje i grafički, osobe su čvorovi, a veze među njima su grane. Podaci se prikupljaju iz četiri izvora: HTML stranice, javno dostupna elektronska pošta, podaci o publikacijama (google scholar) i profili (FOAF). Podaci iz svih izvora podataka se transformiraju u jednu ontologiju. Pri unosu podataka u ontologiju obavlja se i zaključivanje nad tim podacima. U svom profilu svaki istraživač definira svoje interese, te aplikacija pruža i grafički prikaz tema tih interesa koji su međusobno povezani ovisno o tome koliki broj korisnika dijeli interes za određen par tema.

Slučaj uporabe: Korisnik može vizualno pregledavati društvenu mrežu istraživača semantičkog weba, kao i njihove veze s područjima interesa.

Funkcionalnosti:

- integracija
- pretraživanje (vizualno)

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

DynamicView

Opis: Osnovna funkcionalnost DynamicView [Gao2009] aplikacije je vizualizacija distribucije i evolucije područja istraživanja na 20 popularnih fakulteta u SAD-u i u Kinu. Fakulteti koji dolaze u obzir su odabrani u skladu s objavljenim listama od strane ministarstva i novina. Web stranice tih fakulteta se analiziraju i posebnim metodama izvlače područja istraživanja i istraživači koji se tim područjima bave. Aplikacija bi mogla pomoći istraživačima da vide na jednom mjestu koliko im je područje aktualno i tko se sve bavi istim područjem.

Slučaj uporabe: Korisnik može pogledati popularne teme istraživanja u SAD-u kao i u Kini. Može se pregledati i kroz vremenski period kako se popularnost mijenjala. Svi podaci se prikazuju u obliku grafova.

Funkcionalnosti:

- integracija

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

Personal Publication Reader (PPR)

Opis: PPR [Baumgartner2005] ja aplikacija koja pruža personaliziran i jedinstven pogled na publikacije koje se mogu naći na webu. Informacije o publikacijama se periodički dohvaćaju te se pri procesiranju novih podataka automatski pokušava detektirati novo znanje, odnosno, izvlači se i indeksira u repozitorij. Za potrebe aplikacije kreirane su dvije ontologije: jedna za opis publikacija i druga za opis autora. Napravljene su i veze između ontologija. Aplikacija pruža pregled svake publikacije pojedinačno: tko je autor, koje su grupe istraživača vezane uz rad, koje je sve druge radove objavila ista grupa istraživača, koje su publikacije slične, itd… Prikazuju se i detaljni podaci o istraživačima, npr. broj telefona, mail, uloga u istraživačkoj grupi, itd… Postoji poseban modul koji kreira i personaliziran pregled podataka ovisno o interesu koji korisnik specificira. Dakle, ovisno o području interesa koje korisnik unese, aplikacija mu nudi samo publikacije vezane uz to područje.

Slučaj uporabe: Korisnik po specificiranom području interesa može pregledavati publikacije koje tom području pripadaju (može pregledavati same radove, autore radova, itd..).

Funkcionalnosti:

- pretraživanje
- integracija
- personalizacija (preporučen sadržaj)
- semantički repozitorij

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

CONFOTO

Opis: Confoto [Nowack2006] je servis koji pruža mogućnosti pregledavanja i označavanja fotografija sa konferencija. U pozadini rješenja je RDF, ali sa korisničke strane aplikacije korišteni su posljednji trendovi iz razvoja web aplikacija da bi aplikaciju bilo što ugodnije koristiti. Slike nije nužno fizički kopirati u repozitorij, već je moguće referenciranje pomoću veza. Servis je izgrađen u sklopu postojećeg CMS rješenja, kao dodatna funkcionalnost.

Slučaj uporabe: Budući da aplikacija nije dostupna on-line, ne može se vidjeti kako se koristi cjelokupni sustav. Što se samog servisa tiče, moguće je dodavati, komentirati, označavati i napredno pregledavati fotografije. Postoji i mogućnost pretraživanja osoba koje se javljaju na slikama, a naravno bili su sudionici konferencija.

Funkcionalnosti:

- integracija
- pretraživanje
- označavanje
- semantički repozitorij

Odabrana kategorija:

- **procjenitelj A:** semantički sustav za upravljanje znanjem
- **procjenitelj B:** semantički sustav za upravljanje znanjem

EKOSS

Opis: EKOSS [Kraines2007] je razvijen s namjerom razmjene ekspertnog znanja istraživača iz područja vezanih uz biologiju, inženjerstvo i okoliš. Znanje je pohranjeno u obliku različitih pisanih radova, predavanja, raznih izračuna i drugih multimedijalnih sadržaja. Sav materijal koji čini znanje pri pohranjivanju se označava (korištenjem pojmova ontologije) od strane autora. Kasnije se na temelju oznaka drugim korisnicima pruža i napredno semantičko pretraživanje znanja.

Slučaj uporabe: *O projektu je nedovoljno podataka da bi se mogao pretpostaviti.*

Funkcionalnosti:

- integracija
- pretraživanje
- označavanje

Odabrana kategorija:

- **procjenitelj A:** semantički sustav za upravljanje znanjem
- **procjenitelj B:** semantički sustav za upravljanje znanjem

PaperPuppy

Opis: PaperPuppy [Golbeck2006] je zamišljen kao aplikacija koja bi omogućila da se mogu identificirati članci, osobe i institucije od značaja, kao i mogući oblici suradnje unutar zajednice istraživača semantičkog weba. Isto tako, pomoću aplikacije mogu se pratiti novi trendovi u istraživanjima. Korisnici mogu i sami unositi pravila koja se onda koriste kod zaključivanja (npr. da su dvije instance jednake). Također, korisnici mogu pohranjivati i označavati slike, te pomoću tih istih oznaka dohvaćati ih kasnije. Podaci nad kojima aplikacija radi skupljani su iz različitih izvora podataka ili uneseni od strane korisnika.

Slučaj uporabe: Korisnici mogu pretraživati i pregledavati podatke o istraživačima semantičkog weba. Dodatno, mogu pisati i komentare na neke podatke. Napredniji korisnici mogu definirati i nova pravila koja bi se koristila u radu aplikacije.

Funkcionalnosti:

- integracija
- pretraživanje
- označavanje

URL aplikacije: http://paperpuppy.mindswap.org/

Odabrana kategorija:

- **procjenitelj A:** semantički sustav za upravljanje znanjem
- **procjenitelj B:** semantički sustav za upravljanje znanjem

Foafing the music

Opis: Foafing the music [Celma2005] je aplikacija kojoj je osnovna namjena da sugerira korisnicima preporuke u području glazbe na temelju glazbenog ukusa korisnika i navika slušanja. Informacije o muzici sakupljaju se iz preko tisuću RSS feed-ova (npr. najave koncerata, nova izdanja, mp3 blogova itd…). Interesi korisnika (omiljeni glazbenici i pjesme) kao i svi ostali njegovi privatni podaci, pohranjeni su u FOAF profilu korisnika. Ovaj projekt je zapravo nastavak SIMAC projekta koji je također ovdje opisan, s dodatkom da integrira podatke iz više izvora i nudi personaliziran pristup aplikaciji.

Slučaj uporabe: Aplikacija ima brojne funkcionalnosti za korisnika. Sustav korisniku nudi sadržaj koji mu može biti zanimljiv. Takav sadržaj može pregledavati, može preuzimati mp3 sadržaj, prikazuje mu se i mjesto koncerata na karti, može slušati pjesme, itd… Ograničenje je da korisnik ne može pristupati podacima van svojih interesa.

Funkcionalnosti:

- integracija
- preporučen sadržaj

URL aplikacije: http://foafing-the-music.iua.upf.edu/

Odabrana kategorija:

- **procjenitelj A:** sustav preporučenog sadržaja
- **procjenitelj B:** sustav preporučenog sadržaja

MultimediaN E-Culture demonstrator

Opis: MultimediaN E-Culture demonstrator [Schreiber2006] je aplikacija koja omogućava pregled nad integriranim podacima (multimedijalan sadržaj) iz područja kulturne baštine u Nizozemskoj. Integrirani su podaci iz nekoliko muzeja. Kao semantička podrška podacima koristi se pet vokabulara. Korisniku je omogućeno pregledavanje, pretraživanje, označavanje sadržaja, kao i facetno pretraživanje po pojmovima različitih vokabulara.

Slučaj uporabe: Korisnik može samo pregledavati sadržaj, pretraživati sadržaj upisom ključnih riječi (sustav predlaže unos) ili facetno pretraživati.

Funkcionalnosti:

- integracija
- pretraživanje (facetno)

URL aplikacije: http://e-culture.multimedian.nl/demo/session/search

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

PressIndex

Opis: PressIndex [Amardeilh2006] je sustav koji koriste tvrtke za nadgledanje medija i pohranjivanje novinskih isječaka u kojima se spominju. Budući da je proces automatiziran, korištenjem algoritama obrade prirodnog jezika, prepoznaju se članci u kojima su tvrtke citirane, takav sadržaj se izvlači, preformulira korištenjem ontologije i sprema u semantički repozitorij. Korekcije automatskog označavanja mogu se obaviti manualno. Iste vijesti se mogu javiti u nekoliko izvora koji se nadgledaju, uz pomoć ontologije sustav je u mogućnosti prepoznavati iste informacije te ih na odgovarajući način i tretirati, kao i zaključiti nove informacije. Budući da je sav sadržaj pohranjen u semantički repozitorij, moguća su napredna pretraživanja.

Slučaj uporabe: Korisnici imaju svjež sadržaj, koji se automatski generira, novinskih članaka od važnosti za njihovu tvrtku. Nad takvim sadržajem korisnici mogu provoditi napredna semantička pretraživanja.

Funkcionalnosti:

- integracija
- pretraživanje (facetno)
- označavanje
- semantički repozitorij

Odabrana kategorija:

- **procjenitelj A:** semantički sustav za upravljanje znanjem
- **procjenitelj B:** semantički sustav za upravljanje znanjem

Semantic MediaWiki

Opis: Semantic MediaWiki [Krötzsch2006] je proširenje klasične Wiki aplikacije, na način da pruža poboljšane funkcionalnosti u pogledu pretraživanja, organiziranja, označavanja, pregledavanja, evaluiranja i dijeljenja wiki sadržaja. Dok klasične Wiki aplikacije sadrže samo teskt, koji računala ne mogu razumijeti, Semantic MediaWiki posjeduje i semantičke oznake koje omogućavaju veću razinu kolaboracije. Označavanje se svodi na definiranje *kategorije*, *veza* i *atributa* za neku wiki stranicu. Korist od označavanja je vidljiva u pretraživanju koje wiki podržava. Zanimljiva je mogućnost da se upit ugradi u stranicu i tako pruža uvijek dinamički sadržaj (npr. prikaži sve konferencije održane 2009.).

Slučaj uporabe: Korištenje ove aplikacije je potpuno jednako korištenju standardnih Wiki aplikacija, s tim da pruža dodatne mogućnosti kod označavanja sadržaja koji se unosi. A s druge strane, pruža i više mogućnosti u pogledu korištenja takvog sadržaja, npr. obavljanje naprednih upita.

Funkcionalnosti:

- semantički repozitorij
- označavanje
- pretraživanje

URL aplikacije: http://semantic-mediawiki.org/wiki/Semantic_MediaWiki

Odabrana kategorija:

- **procjenitelj A:** semantički sustav za upravljanje znanjem
- **procjenitelj B:** semantički sustav za upravljanje znanjem

Falcon-S

Opis: Falcon-S [Wu2006a] je aplikacija koja radi nad domenom nogometa. Podaci se prikupljaju iz dva izvora, iz službenih stranica FIFA-e i UEFA-e. Robot pretražuje sadržaj tih stranica i izvlači slike, analizira njihov kontekst i indeksira ih na odgovarajući način u bazi znanja. Trenutno aplikacija nudi tri mogućnosti: pretraživanje, semantičko pretraživanje i semantičko pregledavanje. Obično pretraživanje se ostvaruje po ključnim riječima. Semantičko pretraživanje nudi grafički prikaz u kojem korisnik slaže upite koristeći predefinirane komponente. Pregledavanje se odvija po pojmovima iz ontologije. Kao zanimljivu pogodnost koju aplikacija nudi je korištenje boje dresova kod indeksiranja slika uz neku momčad (budući da timovi imaju propisane boje dresove).

Slučaj uporabe: Korištenjem semantičkog pretraživanja korisnik može postaviti upit kao što je: "Pronađi mi sve igrače koji igraju u istom timu sa Ronaldom".

Funkcionalnosti:

- integracija
- pretraživanje

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

Cantabria

Opis: Cilj Cantabria [Carrascal2009] projekta je bio iskoristiti semantičke tehnologije za implementaciju inteligentne integracije informacija o regiji u Španjolskoj (Kantabrija) i njenom kulturnom nasljeđu. Aplikacija je izgrađena nad podacima koji se skupljaju iz više izvora i koji se transformiraju u ontologiju koja je kombinacija nekoliko standardnih modela. Autori navode da je bio izazov dizajnirati aplikaciju koja radi i nad specifičnim podacima te da je tako mogu koristiti eksperti u tom području, a da se pri tom istovremeno omogući i intuitivno pregledavanje sadržaja ostalim „običnim“ korisnicima. Ovisno o profilu korisnika aplikacija će nuditi drugačije funkcionalnosti. Aplikacija omogućava obično pretraživanje po ključnim riječima, kao i napredno pretraživanje, te pretraživanje po karti i vremenskoj liniji. Također postoji i „oblak oznaka“ (eng. tagcloud) koji korisnika direktno usmjeri na najpopularnije pojmove ontologije. Aplikacija ima dva izbornika koji također nude direktan pristup pojmovima ontologija i kontekstualno sličnim pojmovima ovisno o tome što korisnik pregledava. Dodatno, aplikacija ima i repozitorij dokumenata, tako dok korisnik pregledava neki pojam, nude mu se poveznice na dokumente u kojima se on spominje. Kako bi aplikacija bila što ugodnija za korištenje, postoji mogućnost vizualizacije, odnosno prikaza slika. Podaci potrebni za opis slika pohranjuju se u posebnu ontologiju.

Slučaj uporabe: Korisnik može pregledavati i pretraživati pojmove o regiji Kantabrija koristeći obično pretraživanje ili pretraživanje korištenjem karte ili vremenske linije. Ovisno o profilu korisnika nude mu se različiti sadržaji, te ovisno o pojmovima koje je korisnik pregledavao mijenjat će se sadržaj ponuđenih poveznica na druge relevantne pojmove.

Funkcionalnosti:

- integracija
- pretraživanje
- personalizacija

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

CHIP

Opis: CHIP [Aroyo2007] projekt je nastavak projekta „MultimediaN E-Culture demonstrator“. CHIP aplikacija pruža personalizirani pristup digitalnim podacima muzeja Nizozemske. Za semantički opis podataka koriste se četiri vokabulara. Osnovne tri funkcije aplikacije su:

- sustav preporuka koji na temelju ocjena dodijeljenih umjetničkim djelima od strane korisnika sugerira sadržaj koji bi ga mogao zanimati
- generator personalizirane rute po muzeju na temelju profila korisnika
- modul za mobilne uređaje koji korisnicima pomaže da fizički prate rutu koja im je preporučena

Korisnik predložene rute može pretraživati i prikazivati ih na različite načine, poredane po vremenskoj liniji ovisno kada su djela nastala, prikazati ih na karti ili poredati po autorima. Korisnik može i sam kreirati ture pretražujući sadržaj muzeja.

Slučaj uporabe: Korisnik dodjeljuje ocjene umjetničkim djelima, te su te ocjene temelj na osnovu kojeg se omogućuje osnovna funkcionalnost aplikacija, a ta je: generiranje muzejskih ruta za svakog korisnika pojedinačno. Takve rute korisnik može pretraživati i pregledavati na različite načine ili, ako je nezadovoljan ponuđenim, može kreirati vlastitu rutu.

URL aplikacije: http://www.chip-project.org/

Funkcionalnosti:

- integracija
- pretraživanje
- personalizacija
- preporučen sadržaj

Odabrana kategorija:

- **procjenitelj A:** sustav preporučenog sadržaja
- **procjenitelj B:** sustav preporučenog sadržaja

Doris

Opis: Doris [Bhagdev2007] je programska potpora za izradu aplikacija kojima se može stvarati sadržaj, spremati nestrukturirani i isti semantički obogatiti te pretraživati sav pohranjeni sadržaj. Osnovna funkcionalnost je omogućavanje prepoznavanja i pohrane znanja za vrijeme kreiranja sadržaja. Podaci koji se stvaraju od strane korisnika označavaju se pojmovima iz ontologije domene, te se kasnije koriste kod pretraživanja. Kako bi se moglo semantički označiti i starije podatke (word dokumenti, ppt prezentacije, itd…), koji se nisu kreirali kroz sustav, podržano je automatsko izdvajanje znanja iz postojećih dokumenata. Također, podržana je i mogućnost za ručnim označavanjem, ako je potrebno nešto dodatno označiti ili popraviti rezultate automatskog izdvajanja znanja i označavanja. Korisniku je omogućeno napredno pretraživanje korištenjem ključnih riječi potpomognuto semantikom. Kod kreiranja novih sadržaja također se koristi pohranjeno znanje da bi korisniku olakšalo unos i korištenje.

Slučaj uporabe: Kod unosa novog sadržaja korisničko sučelje koristi podatke iz baze znanja da bi sugeriralo unos, te na taj način olakšalo korištenje. Korisnik može pretraživati sav pohranjen sadržaj korištenjem vizualnog prikaza ontologije i odabirom određene kategorije ili pojma, te dodatno može kombinirati tako postavljen uvjet sa unošenjem ključnih riječi (sustav ih isto može sugerirati).

Funkcionalnosti:

- izdvajanje sadržaja iz nestrukturiranih sadržaja
- označavanje sadržaja
- pretraživanje
- zaključivanje
- semantički repozitorij

Odabrana kategorija:

- **procjenitelj A:** semantički sustav za upravljanje znanjem
- **procjenitelj B:** semantički sustav za upravljanje znanjem

EachWiki

Opis: EachWiki [Zhang2009] je nadogradnja postojećeg Wiki sustava koji bi trebao poboljšati i pomoći autorima pri kreiranju sadržaja. Sustav se temelji na AJAX tehnologiji i zamišljena je njegova visoka interaktivnost pri sugeriranju sadržaja korisnika. Osnovna funkcionalnosti sustava su: preporuka poveznica (jer je otkriveno da u Wiki sustavima često poveznice između dokumenata nisu postavljene, jer ih je nepraktično tražiti za vrijeme kreiranja sadržaja), preporuka kategorija i preporuka značajki. Svaki sadržaj koji se kreira pridodaje se nekoj kategoriji, a nije sasvim jednoznačno u koju ga kategoriju treba svrstati. Na temelju sadržaja stranice koju je korisnik kreirao sugeriraju se potencijalne kategorije od kojih korisnik odabire relevantne. Kreiranja značajki radi na sličan način, sustav prepozna kakav se sadržaj unosi pa ponudi i druge značajke čije bi vrijednosti trebalo unijeti.

Slučaj uporabe: Korisnik unese „Microsoft“, sustav mu odmah nudi da unese podatke: „tip tvrtke“, „naziv tvrtke“ i „godina osnivanja“. Također, korisnik može odabrati opciju „ponudi kategorije“, pa iz liste odabrati kategorije u koje smatra da sadržaj pripada. Postoje još brojne mogućnosti automatskog sugeriranja.

Funkcionalnosti:

- semantički repozitorij
- označavanje (poboljšana struktura i veze baze znanje)

Odabrana kategorija:

- **procjenitelj A:** semantički sustav za upravljanje znanjem
- **procjenitelj B:** semantički sustav za upravljanje znanjem

GroupMe!

Opis: GroupMe! 0 aplikacija omogućava korisnicima da grupiraju i označe multimedijalan sadržaj dostupan na webu. Aplikacija ima sučelje koje je lako koristiti i kroz koje korisnici sakupljaju i grupiraju sadržaj, te koje im dozvoljava da označe sve što im se sviđa. Svaka interakcija korisnika se pohranjuje kao RDF opis. Korisnici mogu sami kreirati grupu u koju uključe sadržaj koji žele (npr. Beer – grupa u koju se stavlja sadržaj o pivu). Taj isti sadržaj se onda može označavati od strane različitih korisnika. Oznake se kasnije koriste kod pretraživanja. Sadržaj se može i samo pregledavati kroz sustav navigacije, npr. pristupanjem iz "oblaka oznaka".

Slučaj uporabe: Korisnik može kreirati grupu, u nju postaviti sadržaj po izboru, preurediti je po volji i davati oznake kako svom sadržaju tako i tuđem. Kroz aplikaciju može pregledavati i pretraživati sadržaj svih grupa.

URL aplikacije: http://groupme.org/GroupMe/

Funkcionalnosti:

- integracija
- personalizacija
- označavanje
- pretraživanje

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene – označavanje
- **procjenitelj B:** aplikacija posebne primjene – označavanje

Media Watch

Opis: Media Watch [Scharl2009] agregira, filtrira i vizualizira sadržaj sa 150 stranica koje sadrže podatke o okolišu. Pri pohranjivanju podataka koristi se automatski sustav za označavanje (dodaju se geoprostorne i vremenske semantičke oznake), koji provjerava točnost ručnog unosa (traži konflikte), a tamo gdje nedostaje opisnih podataka dodjeljuje ih automatski.
Osnovni ekran prikazuje najpopularnije pojmove u zadnjih tjedan dana, tako da se može prepoznati što je bio trend u tom vremenskom periodu. Na osnovu dokumenta koji korisnik pregledava (centralna funkcionalnost) nude se poveznice na dokumente koji posjeduju slične pojmove ili su blizu po geografskoj lokaciji. Također, korisniku se prikaže i karta na kojoj su označene lokacije koje se javljaju u tekstu. Dodatno, postoje mogućnosti pregledavanja semantičkih poveznica između dokumenata.

Slučaj uporabe: Korisniku se prikaže ekran podijeljen u 7 osnovnih okvira. Korištenje aplikacije započne pretragom po ključnim riječima ili odabirom učestale u vijestima iz zadnjih tjedan dana (takve riječi su odmah ponuđene). Iz dobivenih rezultata korisnik odabire onu koja ga najviše zanima. U dva okvira se pojavljuju poveznice na slične dokumente onom kojeg pregledava u glavnom okviru. Mogu se također gledati i vizualizacije na 3 različite mape: semantička sličnost dokumenata i slične teme, geografska lokacija na karti i prikaz pojmova ontologije u obliku grafa.

URL aplikacije: http://www.ecoresearch.net/climate/#1

Funkcionalnosti:

- integracija
- pretraživanje
- vizualizacija

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

Revyu.com

Opis: Revyu.com [Heath2007] je aplikacija kojoj je osnovna svrha pisanje komentara (osvrta) korisnika na bilo što (npr. gradovi, konferencije, restorani, itd…). Kada korisnik napiše komentar, može ga označiti ključnim riječima ili pružiti poveznice prema stranicama na kojima se nalaze informacije o sadržaju na kojeg se komentar odnosi. Moguće je pretraživanje po ključnim riječima, postoje i direktne poveznice iz "oblaka veza", a mogu se pregledavati i osobe koje su kreirale komentare. Cijela aplikacija je rađena u skladu s najnovijom vizijom semantičkog weba "povezani podaci". Kod registracije korisnika nije potrebno unositi podatke ukoliko osoba već ima FOAF profil objavljen negdje na webu.

Slučaj uporabe: Korisnik može napisati komentar (osvrt), dati ocjenu, označiti ga oznakama da ga je lakše pronaći, kao i pružiti poveznice prema sadržaju koji ima dodatne opise o pojmu koji se ocjenjuje.

URL aplikacije: http://revyu.com/

Funkcionalnosti:

- označavanje
- pretraživanje

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene – označavanje
- **procjenitelj B:** aplikacija posebne primjene – označavanje

RKB Explorer

Opis: RKB Explorer [Glaser2007] pruža jedinstven pogled na integrirane heterogene podatke domene "održivi sustavi". Radi se o podacima o publikacijama, projektima, osobama koji se bave tim područjem, ustanovama na kojima rade i predmetima koji se na tim ustanovama predaju. Sučelje aplikacije je podijeljeno u 6 okvira. U glavnom okviru se prikazuje predmet koju korisnik pregledava (npr. projekt ili osoba). U ostalim okvirima se prikazuju sadržaji vezani uz sadržaj prikazan u glavnom okviru. Glavni okvir pruža i mogućnosti vizualizacije, tako da se npr. kod odabira pojedine osobe prikaže i njegova društvena mreža. Po toj mreži se može pregledavati i odabirati dalje i druge osobe. Za osobu se prikazuje sasvim detaljan pregled (npr. kontakt podaci). Postoji i mogućnost pretraživanja po ključnim riječima podataka nad kojima aplikacija radi. Aplikacija je prilagođena viziji semantičkog weba "povezani podaci".

Slučaj uporabe: Aplikacija je zamišljena da pomogne ljudima koji su zainteresirani za područje "održivih sustava" da lakše pronađu osobe koje se bave tim područjem, na kojim projektima one rade i koje se publikacije objavili.

URL aplikacije: http://www.rkbexplorer.com/

Funkcionalnosti:

- integracija
- pretraživanje (facetno)
- vizualizacija

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

int.ere.st

Opis: int.ere.st [Kim2007] je aplikacija koja agregira, dohvaća i dijeli podatke o oznakama SCOT (Social Semantic Cloud of Tags) ontologije. Ideja je da se podrži standardan zapis oznaka (tag-ova), te da se na taj način omogući njihovo dijeljenje između različitih aplikacija i osoba. Oznake se prikazuju vizualno pomoću „oblaka oznaka“ (tag cloud), te korisnici preko njega (uz dodatne facete) mogu pretraživati oznake.

Slučaj uporabe: Iako je aplikacija dostupna on-line, vrlo je neintuitivna i teško je zaključiti koju krajnju korist donosi korisniku.

URL aplikacije: http://int.ere.st/

Funkcionalnosti:

- integriranje
- označavanje
- razmjena i dijeljenje oznaka
- pretraživanje (uključuje i facetno pretraživanje)

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene - označavanje
- **procjenitelj B:** aplikacija posebne primjene - označavanje

iFanzy

Opis: iFanzy aplikacija [Bellekens2007] pruža personaliziran pristup TV sadržaju: TV rasporedima i IMDB podacima o filmovima, glumcima, itd... Pretraživanje takvog sadržaja odvija se po tri facete: vrijeme (prikazan kao kalendar), žanr (prikazano kao stablo) i lokacija (prikazano kao stablo). Lokacija se odnosi na mjesta snimanja filmova što je dostupno iz IMDB baze podataka. Osim pretraživanja po facetama korisnik može pretraživati i po ključnim riječima. Takvo pretraživanje se može širiti na način da se traže i slični pojmovi koristeći uvedenu semantiku.

Slučaj uporabe: Korisnik odabire vrijeme i željeni žanr, a sustav mu ponudi iz TV programa sadržaj koji zadovoljava kriterije. Ili korisnik unese ime glumca i sustav mu ponudi sve njegove filmove uz pronađeno vrijeme i program emitiranja.

URL aplikacije: http://wwwis.win.tue.nl:8888/SenSee/SenSeeWebClient.html
Novi URL aplikacije: http://www.ifanzy.nl/

Funkcionalnosti:

- integracija
- personalizacija (preporučen sadržaj)
- pretraživanje (facetno pretraživanje)

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

SMART

Opis: SMART [Battista2007] je aplikacija koja rukuje podacima iz područja biologije iz samo jednog izvora podataka. Osnovna funkcionalnost je intuitivno pretraživanje semantički pohranjenih podataka od strane stručnjaka iz područja biologije. Na pohranjenim podacima se koristi funkcionalnost zaključivanja da bi se proširila baza znanja.

Slučaj uporabe: Osnovna funkcionalnost je pretraživanje, alat nudi nadopunjavanje teksta na temelju unesenog sadržaja i tako pomaže pri izgradnji upita. Također, prikazuje upit u obliku grafa, a kao rezultat daje podatke i poveznice (pretraživanje ontologije) na podatke koji su relevantni.

Funkcionalnosti:

- pretraživanje – grafičko prikazivanje upita
- zaključivanje
- semantički repozitorij

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene – pretraživanje i zaključivanje
- **procjenitelj B:** aplikacija posebne primjene – pretraživanje i zaključivanje

WWWatch

Opis: WWWatch [Sumsiripong2007] omogućava vizualizaciju znanja u tri domene: *društvene veze država*, *odnosi između država u vremenskoj liniji* i *promjene u okolišu na karti svijeta*. Vizualiziraju se društvene veze između različitih država svijeta na temelju izvučenih podataka iz arhive vijesti. Veze između država mogu biti neutralne, pozitivne i negativne, ovisno o sadržaju vijesti na temelju koje se dobivaju podaci. Ako u vijesti ima više riječi koje se okarakteriziraju kao negativne, stvorit će se negativna veza, a ako ima više pozitivnih stvorit će se pozitivna veza (izračun koristi dodatno i podatke o povjerenu izvora s kojeg je vijest, itd.). Korisnicima je omogućeno pretraživanje takvih relacija u nekom zadanom vremenskom periodu. Rezultati se prikazuju ili kao grafovi ili kao dijagrami na vremenskoj liniji. Također, dodatno se mogu vizualizirati podaci na karti svijeta o promjenama u prirodi, kao npr. posljedicama globalnog zatopljenja, koji su izvučeni iz različitih dostupnih arhiva vijesti. Budući da se radi samo o pokaznoj aplikaciji, povezane su dvije sasvim različite domene i nema baš neke direktne koristi od funkcionalnosti koje aplikacija pruža.

Slučaj uporabe: Korisnik može u nekom vremenskom periodu dobiti prikaz društvenih odnosa (vijesti) između te dvije države. Također, korisnik može locirati na karti svijeta informacije vezane uz posljedice globalnog zatopljenja.

Funkcionalnosti:

- integriranje
- pretraživanje
- vizualno prikazivanje

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

CultureSampo

Opis: CultureSampo [Hyvonen2008] je posljednji rezultat dugogodišnjeg istraživanja na području semantičkih tehnologija i kulturne baštine u Finskoj. Podaci nad kojima se aplikacija temelji su skupljeni iz brojnih heterogenih izvora podataka iz područja kulturne baštine. Podaci se dodatno skupljaju i iz vanjskih izvora podataka, npr. članaka Wikipedije i sl. Ontologije u koje se ti podaci transformiraju su rezultat prethodnih projekata. Nad sakupljenim podacima se obavi zaključivanje da bi se dobilo još i dodatno znanje. Jedna zanimljivost koju aplikacija pruža da bi se dobila dodatna korist je da se dozvoljava korisnicima da označavaju stare fotografije, npr. kada prepoznaju osobe na njima, da bi se tako dobile nove informacije. Podatke je moguće pretraživati po ključnim riječima, te se pri tom nudi sugeriranje sadržaja pri unosu pitanja. Postoji 9 perspektiva kroz koje se koristi aplikacija: pregled geografskih karti, relacijsko pretraživanje (pretraživanje veza između objekata; koristi se za otkrivanje novog znanja, npr. što Napoleon i Akseli Gallen-Kallela imaju zajedničko), facetno pretraživanje, pregled kolekcija, pregled finske povijesti, vještine i procesi, biografije, Kalevala (finski folklor), Karelia (finska regija). Aplikacija dodatno pruža i mogućnost preporuke sadržaja po mjerama sličnosti.

Slučaj uporabe: Na početku korištenja aplikacije korisnik bira jednu od 9 perspektiva. Dalje, može pregledavati sadržaj (sustav će mu nuditi preporuke), kao i što može pretraživati cjelokupni sadržaj.

URL aplikacije: http://www.kulttuurisampo.fi

Funkcionalnosti:

- integracija
- pretraživanje (facetno pretraživanje)
- preporučen sadržaj

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

IYOUIT

Opis: IYOUIT [Boehm2008] aplikacija je prvenstveno namijenjena kao aplikacija za mobitele, ali postoji i njena verzija za web (ima nešto manje funkcionalnosti). Aplikacija prikuplja podatke o korisniku i oko korisnika, te putem nje korisnici mogu pisati blog, pisati komentare, označavati svoju trenutnu lokaciju (može se i automatski GPS-om preuzimati), komentirati pročitane knjige, međusobno slati poruke ili međusobno dijeliti i označavati multimedijalan sadržaj. Korisnik odabire svoje prijatelje, odnosno kreira svoju društvenu mrežu, te mu je onda omogućena interakcija s tim osobama. Aplikacija može kreirati grupe korisnika na osnovu neke veze, npr. svi prijatelji u istom gradu, ili svi prijatelji istog raspoloženja, i sl. Po osnovnim postavkama nitko osim prijatelja ne može pregledavati sadržaj korisnika, a na korisniku je da po želji promijeni dozvole na vlastiti sadržaj. Moguća je i suradnja IYOUIT aplikacije sa sličnim aplikacijama kao Facebook, twitter, Flickr, last.fm, itd… Npr. sa Facebookom je moguća izmjena statusa. Postoji i mogućnost pretraživanja sadržaja unosom ključnih riječi.

Slučaj uporabe: Aplikacija ima potpuni smisao samo kad je koristi grupa korisnika, tada korisnik između ostalog može sa njima izmjenjivati poruke, multimedijalan sadržaj, odnosno obavljati interakciju.

Funkcionalnosti:

- personalizacija
- društvena mreža (interakcija)
- pretraživanje

URL aplikacije: http://www.iyouit.eu/portal/

Odabrana kategorija:

- **procjenitelj A:** društvena semantička web aplikacija
- **procjenitelj B:** društvena semantička web aplikacija

LODr

Opis: LODr [Passant2009c] je aplikacija koja omogućava da oznake koje su korisnici već postavili na sadržaj diljem weba pretvore u semantičke oznake. Korisnik registrira FOAF profil iz kojeg se pročitaju aplikacije koje korisnik koristi, te im aplikacija onda pristupa da bi mogla preuzeti sve označene podatke. Korisnik tada daje značenje dohvaćenim oznakama, ali nije nužno da označava takve podatke, može koristiti i druge opcije kao eksport u RDF format. Za definiranje semantičkih oznaka koristi se MOAT. Aplikacija pruža i mogućnost pretrage semantički označenih podataka, kao i direktan pristup tim podacima preko „oblaka oznaka“. Usluge pretraživanja se dostupne i neregistriranim korisnicima.

Slučaj uporabe: Nakon što se korisnik registrira, može pregledavati sve svoje označene podatke na jednom mjestu. Može im pristupati direktno preko „oblaka oznaka“. I ono najvažnije, korisnik može svim tim oznakama dati i semantičko značenje (URI).

Funkcionalnosti:

- integracija
- označavanje

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene – označavanje
- **procjenitelj B:** aplikacija posebne primjene – označavanje

paggr

Opis: Paggr [Nowack2008] je web aplikacija pomoću koje se u samo nekoliko koraka mogu kreirati portali koji integriraju više izvora i različitih formata. Sastoji se od dvije osnovne komponente: dijela u kojem se definiraju komponente (npr. widgeti) i dijela u kojem se te komponente uzimaju i koriste pri izradi portala. Paggr koristi tzv. Sparqlete koji su zapravo widgeti, ali se u njima piše kod koristeći SPARQLScript (razvijen od istih autora), pa je na taj način moguće pristupati izvorima podataka u formatu RSS, Atom, RDFa, mikroformati ili SPARQL krajnjim točkama. Aplikacija se temelji na "drag & drop" principu, uglavnom se preuzimaju već gotovi predlošci koji se prilagođavaju u jednu cjelinu. Aplikacija je vrlo slična već poznatim alatima za kreiranje portala korištenjem portleta i sl., ali s puno više mogućnosti.

Slučaj uporabe: Korisnik se registrira i kreira vlastiti portal koristeći već definirane komponente, predloške ili tako da kreira svoje vlastite.

Funkcionalnosti:

- izrada portala

URL aplikacije: http://paggr.com/

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene
- **procjenitelj B:** aplikacija posebne primjene

MT - Redland

Opis: MT – Redland [Williams2009] je nadogradnja postojećeg poznatog sustava za pisanje blogova (Movable Type) u smislu da je dodano RDF trajno spremište podataka (Redlend). Dodane su nove funkcionalnosti u sučelju koje omogućavaju unos meta-podataka (npr. teme, osvrte i lokacije). Za meta-podatke koriste se poznato vokabulari: RSS, FOAF (za opis autora) , SKOS (opis kategorija) i DC (uobičajeni atributi kao datumi i sl.). Budući da su u ovoj novoj inačici sadržaji označeni vokabularima moguće ih je pretraživati korištenjem SPARQL-a.

Slučaj uporabe: Korisnik pomoću ove aplikacije kreira blogove sa mogućnosti da unese semantiku u sadržaj pomoću sučelja. Također, sadržaj blogova se može pretraživati ili eksportirati u RSS.

Funkcionalnosti:

- označavanje

Odabrana kategorija:

- **procjenitelj A:** sustav za upravljanje znanjem
- **procjenitelj B:** sustav za upravljanje znanjem

SIMAC

Opis: SIMAC [Herrera2006] je aplikacija koja nudi preporuke na temelju glazbenog ukusa korisnika. Iz FOAF profila korisnika iščitavaju se interesi i prepoznaju se oni vezani uz glazbu (izvođači). Iz izvora o glazbenim događajima i autorima koje aplikacija obrađuje, pa pohranjuje u vlastiti repozitorij (npr. RSS feed-ovi), pronalaze se relevantni podaci u skladu s profilom korisnika. Dodatne informacije, na temelju kojih se vrše preporuke, izvlače se iz dostupnih demo snimaka, npr: ritam, tempo, tonalitet, glasnoća i sl. Upravo otkrivanje, pohranjivanje i korištenje tih karakteristika skladbi je bio i osnovni cilj projekta koji je rezultirao SIMAC aplikacijom. Implementiran je poseban algoritam koji na temelju karakteristika skladbe pronalazi slične.

Slučaj uporabe: Aplikacija pomoću korisnikovog FOAF profila ponudi listu izvođača (sa vezama na vijesti o njima) složenu prema sličnosti koji bi korisniku mogli biti zanimljivi, jer su u skladu s definiranim interesima.

Funkcionalnosti:

- preporučen sadržaj
- semantički repozitorij

Odabrana kategorija:

- **procjenitelj A:** sustav preporučenog sadržaja
- **procjenitelj B:** sustav preporučenog sadržaja

FOAFMap

Opis: FOAFMap [Passant2009b] je razvijen primarno kao prototip integracije Weba 2.0 i semantičkih tehnologija. Osnovna ideja je da korisnik u aplikaciju unese put do svog FOAF profila gdje god da se nalazi (ideja je da korisnik sam održava profil, a ne da ga aplikacija posjeduje). Iz FOAF profila aplikacija pročita trenutnu lokaciju korisnika (ako je unesena), sve kontakte (foaf:knows) i interese korisnika. Lokacija korisnika i svih kontakata prikaže se na karti. Postoji mogućnost pregleda interesa svih korisnika preko „oblaka oznaka“, te se tako direktno može pristupiti pojedinom interesu i vidjeti kod kojih se sve osoba on javio.

Slučaj uporabe: Korisnik može na karti vidjeti svoju lokaciju, kao i lokacije svih svojih kontakata. Također, može pregledati „oblak oznaka“ interesa i vidjeti koje osobe dijeli iste interese.

Funkcionalnosti:

- integracija
- vizualizacija

URL aplikacije: http://foafmap.net/

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene – vizualizacija
- **procjenitelj B:** aplikacija posebne primjene – vizualizacija

eMerges

Opis: eMerges [Tanasescu2006] je prototip aplikacije koja bi trebala pomagati u hitnim slučajevima. Aplikacija se temelji na 3 izvora podataka: meteorološki podaci (podaci o olujama), geoprostorni podaci i podaci za komunikaciju preko „istovremenih poruka“. Profil za komunikaciju je pohranjen u FOAF/RDF formatu i u njemu se dodatno specificira lokacija korisnika (geo:Point). Koristeći podatke o mjestu oluje, položaju korisnika i podacima o skloništima na određenom području (poligon) sustav savjetuje gdje bi se korisnik mogao skloniti. Korisnik može odabrati neko od ponuđenih skloništa te mu sustav nudi i komunikaciju preko „istovremenih poruka“ s osobama koji bi mu mogle pružiti više informacija oko samog skloništa.

Slučaj uporabe: Korisnik u nekom trenutku zatraži moguća skloništa od elementarne nepogode. Sustav mu na karti prikaže područje oluje i moguća skloništa. Nakon što korisnik odabere neko od tih skloništa, dalje mu se nudi mogućnost komunikacije s osobama koji imaju više informacija o tome kako da dođe do utočišta.

Funkcionalnosti:

- integracija
- vizualizacija

Odabrana kategorija:

- **procjenitelj A:** semantički web portal (semantički ekspertni web sustav)
- **procjenitelj B:** semantički ekspertni web sustav

DOAP Store

Opis: DOAP Store [Passant2009a] je vrlo jednostavna aplikacija kojoj je osnovna uloga na jednom mjestu okupiti informacije o projektima. DOAP vokabular je upravo kreiran za tu namjeru, dakle za opisivanje projekata. Aplikacija se temelji na još jednoj postojećoj tehnologiji, a to je Ping The Semantic Web. Radi se o servisu koji dohvaća listu projekata opisanih DOAP-om. U kratkim intervalima aplikacija poziva servis, preuzima nove opise projekata i pohranjuje ih u vlastiti semantički repozitorij. Aplikacija nudi mogućnost pregledavanja i pretraživanja tako pohranjenih podataka o projektima. Također, aplikacija nudi i SPARQL pristup, pa se podaci mogu pretraživati i proizvoljnim upitima.

Slučaj uporabe: Korisnik može pregledavati listu projekata ili može pretraživati projekte po različitim uvjetima.

Funkcionalnosti:

- integracija
- pretraživanje

URL aplikacije: http://doapstore.org/

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

SIOC Explorer

Opis: SIOC explorer [Heitmann2007b] pruža integriran pogled nad javno dostupnim podacima pisanim SIOC vokabularom. Dakle, radi se o izvorima kao što su forumi, blogovi, mail liste, itd… Početna stranica aplikacije prikazuje sav sadržaj koji je obuhvaćen integracijom (npr. forum, pa pod njim sve postove, itd…). Aplikacija je realizirana korištenjem ActiveRDF programskog okvira koji je neovisan o korištenim vokabularima, tako da aplikacija uspješno rukuje i drugim vokabularima kojima može biti označen sadržaj. Npr. FOAF za opis autora bloga. Još jedna prednost korištenja takvog neovisnog programskog okvira je mogućnost implementacije facetnog pretraživanja.

Slučaj uporabe: Korisnik može pregledavati i pretraživati sav integriran sadržaj, dakle npr. više foruma odjednom.

Funkcionalnosti:

- integracija
- pretraživanje (facetno)

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

SPARQLBot

Opis: SPARQLBot [Nowack2009] je aplikacija putem koje se mogu izvoditi upiti nad postojećim podacima na Webu pisanim u skladu s vizijom "povezanih podataka". Korisnici definiraju naredbe (imena naredbi), a sam upit, kao i format rezultata izgrade koristeći SPARQLScript jezik. Za definiranje naredbi korisnik mora biti prijavljen u aplikaciju (korištenjem OpenID identiteta), a sve ostale opcije može koristiti i kao anoniman korisnik.

Slučaj uporabe: Korisnik se prijavi, te može birati od već kreiranih naredbi koju izvesti ili može definirati novu naredbu, pa pomoću nje obaviti pretraživanje.

Funkcionalnosti:

- integracija
- pretraživanje

URL aplikacije: http://sparqlbot.semsol.org/

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene – pretraživanje i integracija
- **procjenitelj B:** aplikacija posebne primjene – pretraživanje i integracija

Researcher's map

Opis: Researcher's map [Hartig2009] je vrlo jednostavna aplikacija koje se potpuno zasniva na "povezanim podacima". Aplikacija prikuplja podatke o profesorima u Njemačkoj sa stranica fakulteta ili drugih stranica, važno je samo da postoji URI stranice i veza na osobne podatke o profesoru. Profesori se prikazuju na karti na lokaciji koja odgovara njihovom radnom mjestu. Prikaz se može filtrirati po interesima profesorima. Odabirom određenog profesora prikazuju se podaci o njegovim publikacijama (dohvaćene iz DBLP baze koja se objavljuje po pravilima "povezanih podataka").

Slučaj uporabe: Korisnik pristupa aplikaciji u kojoj je prikazana Google karta (Google Map). Može odabrati profesora na karti ili iz liste i tako dobije popis svih njegovih publikacija. Dodatno, može listu profesora filtrirati po njegovim interesima (npr. svi profesori koji se bave semantičkim webom).

Funkcionalnosti:

- integracija
- pretraživanje
- vizualizacija

URL aplikacije: http://researchersmap.informatik.hu-berlin.de/

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

A Linked Open Data Resource List Management (RLM) Tool for Undergraduate Students

Opis: Ova RLM aplikacija [Clarke2009] potpomognuta semantičkim web tehnologijama je u produkciji od 2008. godine. Radi se o aplikaciji namijenjenoj studentima i profesorima. Na početku semestra studenti za svaki kolegij dobiju popis literature (multimedijalan sadržaj). Pred par godina uvedena je mogućnost da popis literature koji su prethodno dobivali na papirima se objavljuje elektronski. Na taj način postala je moguća i direktna veza s fakultetskom knjižnicom (mogućnost rezervacije, broj raspoloživih primjeraka, itd..). Takav sustav ipak krije niz prepoznatih nedostataka, npr. nepostojanost veza (linkova na materijale), nedovoljna interoperabilnost između različitih sustava i nemogućnost dijeljenja podataka (npr. ako je ista literatura potrebna za više kolegija ili izmjena iskustava između studenata), kao i nedostatak povratnih informacija (što studenti misle o ponuđenoj literaturi). Realizirana aplikacija nudi rješenje na navedene nedostatke. Tijekom realizacije razvijene su dvije ontologije: jedna za opis literature, a druga za opis studija, kolegija, modula i sl. Korisnicima je omogućeno označavanje literature za vlastite potrebe (npr. trenutno čitam, planiram pročitati, itd…), a dozvoljeno im je i pisanje komentara na literaturu kao pomoć drugim studentima ili kao povratna informacija profesorima o kvaliteti literature (npr. poglavlje 3 obuhvaća predavanja 5 -7). Ponuđen materijal je povezan, gdje je to moguće, s drugim izvorima u cilju pružanja dodatnih informacija (vizija povezanih podataka). Svi prikazani podaci su opisani RDFa-om što omogućava zanimljivu mogućnost s tehničke strane, naime korisnici sve promjene obavljaju „drag & drop“ naredbama koje mijenjaju model (grafove) u prezentacijskom sloju i nije potrebno stalno kontaktiranje poslužitelja. Tek kada korisnik odabere akciju pohrane podataka, podaci se i spreme na poslužitelju.

Slučaj uporabe: Studenti mogu pregledati dodatne izvore predložene od strane profesora za pojedine kolegije, dobiti dodatne informacije pomoću uspostavljene interoperbilnosti, označavati je i davati komentare na njenu kvalitetu. Profesori mogu pregledavati komentare studenata u cilju poboljšana ponude literature.

Funkcionalnosti:

- integracija

- pretraživanje
- kolaboracija (interakcija)

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene
- **procjenitelj B:** sustav za upravljanje znanjem

A Semantic Web Content Repository for Clinical Research - SemanticDB

Opis: SemanticDB [SWCSUC2009] je aplikacija rađena za potrebe klinike iz Clevelanda. Podaci o pacijentu se pohranjuju u RDF spremište, automatski iz nestrukturiranih izvora, kao i kroz forme za unos koje čine dio aplikacije. Moguća je pohrana u obje forme, kao nestrukturirane datoteke i kao RDF. Nad takvim RDF spremištem moguće je izvoditi nove zaključke. Kao osnovna prednost ovog rješenja ističe se mogućnost korištenja doktorima bliske terminologije, jer su ukinuta ograničenja naziva polja u relacijskoj bazi i na ekranima.

Slučaj uporabe: Doktorima je omogućeno pregledavanje povijesti bolesti pacijenta na vremenskoj liniji, uz postavljanje ad-hoc upita i pretraživanje korištenjem pojmova iz ontologije.

Funkcionalnosti:

- pretraživanje
- zaključivanje
- semantički repozitorij

Odabrana kategorija:

- **procjenitelj A:** semantički sustav za upravljanje znanjem
- **procjenitelj B:** semantički sustav za upravljanje znanjem

An Intelligent Search Engine for Online Services for Public Administrations

Opis: Grad Zaragoza svojim građanima pruža između 500 i 600 online servisa, koji su organizirani u 20ak kategorija, kao npr. porezi, promet, okoliš, edukacija, itd… Praksa je pokazala da građani ne uspijevaju uvijek pronaći traženu informaciju zbog više problema: neke web stranice imaju popis usluga koje pružaju, pa građani moraju pronaći onaj koji ih zanima iz popisa, ili kada postoji tražilica, građani pri pretraživanju ne koriste vokabular koji je korišten od strane gradskih službi (npr. kada korisnik želi baciti stari kućanski aparat, mora znati da traži uslugu koja se zove „skupljanje velikih stvari"). Kao rješenje na uočene probleme izgrađen je semantički pretraživač na stranicama grada Zaragoze [Cardoso2008]. Značenje svake usluge definirano je ontologijom. Nad upitom korisnika provode se metoda analize prirodnog jezika i zaključivanja nad ontologijom.

Slučaj uporabe: Korisnik putem web aplikacije grada Zaragoze u tražilicu upisuje upit glede usluga koje pružaju javne djelatnosti grada Zaragoze. Kao rezultat korisnik dobije konkretan odgovor umjesto duge liste dokumenata. Sustav je u mogućnosti ponuditi i povezane usluge.

Funkcionalnosti:

- pretraživanje

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene – pretraživanje
- **procjenitelj B:** aplikacija posebne primjene – pretraživanje

PharmaSURVEYOR

Opis: PharmaSURVEYOR je aplikacija kojom se mogu provjeravati moguće nuspojave uzimanja lijekova. Osnovna funkcionalnost je provjeravanje posljedica koje može uzrokovati kombinacija lijekova. Klasičnim pretraživanjem korisnici ne mogu doći do traženih odgovora budući da je vrlo važno kombinirati znanje da bi se moglo identificirati potencijalne probleme. Kako bi se moglo implementirati takvo znanje nužno je koristiti više izvora podataka, kao i osigurati kolaboraciju farmaceuta, doktora medicine i pacijenata. Budući da je veliki razvoj u farmaciji i medicini, statičke strukture za opis znanja (kao baza podataka) nisu pogodne, jer se stalno trebaju dodavati nove karakteristike. Zbog navedenih razloga, za pohranu navedenog znanja kreirana je ontologija.

Slučaj uporabe: Korisnik unese lijekove koje koristi, kao i način na koji ih uzima. Sustav izbaci listu potencijalnih posljedica (simptoma tih posljedica), kao i potencijalne opasnosti kod kombiniranja pojedinih lijekova.

Funkcionalnosti:

- zaključivanje

URL aplikacije: http://www.pharmasurveyor.com/

Odabrana kategorija:

- **procjenitelj A:** semantički ekspertni web sustav
- **procjenitelj B:** semantički ekspertni web sustav

CRUZAR

Opis: CRUZAR aplikacija je namijenjena turistima grada Zaragoze. Pomoću ove web stranice nastoji se automatizirati proces koji inače obavlja turist, a to je generiranje ruta za razgledavanje. Korisnik ima mogućnost specificirati svoje interese i preferencije, te mu sustav predloži rutu za razgledavanje. Na generiranu rutu korisnik opet može utjecati i raditi modifikacije. Važan preduvjet koji je bio zadovoljen da bi ovakav projekt mogao i zaživjeti je taj što su svi potrebni podaci već postojali, samo su razasuti po različitim bazama podataka (npr. podaci o zbivanjima u gradu, geoinformacijski sustav koji sadrži podatke o svim lokacijama, baza o svim znamenitostima Zaragoze, itd…), koje su upravo semantičkim tehnologijama integrirane. Kreirana ontologija obuhvaća sljedeće podatke: znamenitosti, profil korisnika i konfiguracija ruta. Za izračun ruta koristi se kompleksan algoritam koji nastoji ugraditi prihvatljiv omjer kvalitete i kvantitete u rute za razgledavanje.

Slučaj uporabe: Korisnik unese podatke o svojim preferencijama (voli li šetnje, kakve znamenitosti ga zanimaju, itd..), unese datume svog boravka, mjesto gdje je smješten, te mu sustav ponudi rute za razgledavanje.

Funkcionalnosti:

- integracija
- preporučen sadržaj

URL aplikacije: http://www.zaragoza.es/turruta/Turruta/en/index_Ruta

Odabrana kategorija:

- **procjenitelj A:** sustav preporučenog sadržaja
- **procjenitelj B:** sustav preporučenog sadržaja

Enhancement and Integration of Corporate Social Software Using the Semantic Web

Opis: Ova aplikacija je proizvod najveće energetske tvrtke u Francuskoj. Da bi poboljšali kolaboraciju i dijeljenje znanja, tvrtka je uvela u svoj odjel razvoja novi sustav. No, kod korištenja sustava uočeni su sljedeći problemi: znanje kreirano korištenjem Wiki aplikacija nije lako pretraživati, označavanje podataka unosi heterogenost i dvosmislenost, a RSS feed-ovi produciraju puno podataka, pa je na taj način teško pratiti informacije o tvrtki. Rješenje na navedene probleme uvedeno je semantičkim tehnologijama. Sada već poznati vokabulari i tehnike semantičkog weba upravo su rezultat ovog projekta. Za uvođenje strukturiranosti u sadržaj koji kreiraju djelatnici kreiran je SIOC vokabular. U Wiki sustav dodani su moduli koji onda omogućavaju unos podataka koji su označeni SIOC, FOAF, DC, Geonames i SKOS vokabularima. Kao rješenje na nered koji uvodi klasičan način označavanja, kreiran je programski okvir MOAT. U centralni semantički repozitorij pohranjuju se i podaci iz drugih aplikacija, npr. blogovi, RSS feed-ovi, itd.. Uvođenjem semantike u sustav omogućeno je točnije pretraživanje znanja.

Slučaj uporabe: Korisnici unose znanje vezano uz tvrtku kroz module Wiki sustava. Takav sadržaj mogu označavati korištenjem MOAT-a, a mogu ga i pretraživati korištenjem semantičkih metoda i tako dobiti preciznije odgovore.

Funkcionalnosti:

- integracija
- pretraživanje
- označavanje

Odabrana kategorija:

- **procjenitelj A:** sustav za upravljanje znanjem
- **procjenitelj B:** sustav za upravljanje znanjem

Enhancing Content Search Using the Semantic Web

Opis: Oracle Technology Network (OTN) je osnovni izvor tehničkih informacija za zajednicu osoba zaduženih za razvoj u Oracle tehnologijama. Njihove web stranice omogućavaju pristup blog-ovima, forumima, dokumentaciji o proizvodima, obavijestima o novim izdanjima proizvoda, kao i načinima za preuzimanje programske potpore. Budući da se radi o različitim i jako dinamičnim informacijama tradicionalne tražilice više nisu zadovoljavale potrebe korisnika. Kao rješenje na uočen problem realizirana je aplikacija koja integrira sve izvore podataka. Pri agregiranju izvora podaci se automatski označavaju kreiranom ontologijom. Na taj način stvoreno je jedino mjesto za navigaciju i pretraživanje podataka, s mogućnošću personalizacije i dodatnim mogućnostima jednostavnijeg pristupa informacijama preko „oblaka oznaka“ ili vremenskih linija.

Slučaj uporabe: Korisnik putem aplikacije može pregledavati i pretraživati sve informacije vezane uz Oracle tehnologije. Također, korisnik specificiranjem svojih interesa može personalizirati izgled aplikacije.

Funkcionalnosti:

- integracija
- pretraživanje
- personalizacija

Odabrana kategorija:

- **procjenitelj A:** semantički web portal
- **procjenitelj B:** semantički web portal

Real Time Suggestion of Related Ideas in the Financial Industry

Opis: Budući da je financijska industrija jako kompetitivna, prisutan je veliki pritisak za generiranje novih ideja i proizvoda. Španjolska banka Bankinter je poznata po inovativnosti. Nove ideje su iznosili zaposlenici odjelu za upravljanje resursima. Kako je broj ideja rastao, bilo je sve teže evaluirati ideje i razaznati koje su nove, odnosno koliko se preklapaju. Da bi se rasteretio taj manualni proces, kreiran je sustav kojim se unose nove ideje, te se pri samom unosu analizom teksta traže koncepti i njihova sličnost sa već pohranjenim idejama. Na taj način korisnik pri unosu ideje već zna radi li se o zbilja novoj ideji. Također, budući da je tekst povezan s pojmovima ontologije iz domene financija, omogućeno je pretraživanje svih iznesenih ideja.

Slučaj uporabe: Aplikacija ima dvije vrste korisnika, oni koji unose nove ideje i oni koje takve ideje moraju evaluirati. Sustav pruža niz mogućnosti za analizu tako unesenih ideja, npr. izvještaj po datumu, po zaposlenicima, odjelima… Također, sustav može pronaći sve slične ideje i obrazložiti zašto ih je tako ocijenio.

Funkcionalnosti:

- pretraživanje

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene
- **procjenitelj A:** aplikacija posebne primjene (sustav za upravljanje znanjem)

Faviki

Opis: Faviki [SWCSUC2009] je aplikacija koja omogućava označavanje sadržaja pojmovima iz Wikipedije (DBPedije). Oznake mogu biti unesene na različitim jezicima.

Slučaj uporabe: Korisnik započne unos pojma kojeg želi označiti, te mu sustav ponudi pojmove iz Wikipedije koji odgovaraju unosu iz kojih može odabrati resurs kojim će označiti sadržaj.

Funkcionalnosti:

- označavanje sadržaja

Odabrana kategorija:

- **procjenitelj A:** aplikacija posebne primjene – označavanje sadržaja
- **procjenitelj B:** aplikacija posebne primjene – označavanje sadržaja

Twine

Opis: Twine [SWCSUC2009] je aplikacija koja je namijenjena da korisnicima prikazuje samo sadržaj za koji su zainteresirani i na taj način da ih sačuva od prekomjerne izloženosti brojnim informacijama. Pojam *„twine“* predstavlja grupu sadržaja određene teme koju kreiraju korisnici aplikacije. Umjesto da korisnik prati više izvora koji su mu zanimljivi pojedinačno, može promatrati samo određeni twine u kojem korisnici sakupljaju sadržaj, te je na taj način pokriven veći skup izvora od onih koje bi mogli sami kontrolirati. Osim dodavanja sadržaja od strane korisnika, podržano je i automatsko dohvaćanje sadržaja. Sadržaj se može i označavati, pa kasnije po tom i pretraživati, a dodatno sadržaj se može i komentirati. Twine ne mora biti javan, moguće je imati i privatne, zaštićene koji su namijenjeni samo užem krugu ljudi, npr. sa neke konferencije, iz škole, i sl. Korisnici se mogu međusobno povezivati u grupe, po raznim kriterijima i međusobno komunicirati.
Sustav može korisniku i predlagati druge twineove na temelju onih kojima se korisnik pridružio, dakle implementiran je određen stupanj personalizacije.

Slučaj uporabe: Korisnik se može pridružiti već postojećem twineu koji ga zanima i na taj način pratiti nove informacije u tom području. Može kreirati i vlastiti twine kojem se kasnije mogu pridružiti drugi korisnici.

URL aplikacije: http://www.twine.com/

Funkcionalnosti:

- integriranje
- označavanje
- personalizacija
- preporučen sadržaj

Odabrana kategorija:

- **procjenitelj A:** društvena semantička web aplikacija
- **procjenitelj B:** društvena semantička web aplikacija

Tablica 18. Karakteristike aplikacija uzetih u razmatranje u cilju definiranja kategorija semantičkih web aplikacija

naziv	izvor	vrsta	vizija	svrha	personalizacija	upitni jezik	vrsta podataka	broj izvora podataka	vokabulari	pretraživanje	zaključivanje	pristup podacima od strane drugih servisa	prezentacijski sloj	robot	model podataka	trajno spremište podataka
Seal	ISWC 2003	programska potpora za semantički web portal	1	Ne zadovoljava uvjete web aplikacije												
Seco	ISWC 2003	semantički web portal	1	integriran pregled nad RSS feed-ovima	ne	RDQL	nema osjetljivih	više (web)	FOAF RSS	ograničeno po malom broju ključnih riječi	da	da	ne	da	ne zna se	da
AnnoTerra	ISWC 2003	semantički web portal	1	integriranje i bolje pretraživanje nad povezanim izvorima podataka (NASA)	ne	ne	nema osjetljivih	više	vlastita OWL ontologija	da	da	da	ne	ne	ne zna se	ne zna se
BuildingFinder	ISWC 2003	Nije više dostupno														
Semblog	ISWC 2003	Nije više dostupno														
CS AKTiveSpace	ISWC 2003	semantički web portal	1	integriran pregled podataka o istraživačima, institucijama i radovima iz područja računarstva	ne	RDQL	ima osjetljive, ali ih ne tretira	više	vlastiti (RDF)	da – ograničeno preko grafičkog sučelja	da	da	ne	da	ne zna se	da
SWEET	ISWC 2003	aplikacija posebne primjene - pretraživanje	1	poboljšano pretraživanje podataka o zemlji i okolišu	ne	ne zna se	nema osjetljivih	samo pojmovi ontologije	vlastita OWL ontologija	da	ne zna se	ne	ne	ne	ne zna se	da (SBP)
BioInformatics	ISWC 2003	Nije više dostupno														
GeoShare	ISWC 2003	Nije više dostupno														
MusiDB	ISWC 2004	semantički web portal	1	personaliziran sustav koji nudi sadržaj na temelju preporuka	da	ne zna se	nema osjetljivih	3	MusicBrainz i vlastita	ne	ne	ne	ne	Ne	ne zna se	ne zna se

Madiera portal	ISWC 2004	Nije više dostupno														
SWAP	ISWC 2004	Nije više dostupno														
SemanticOrganizer	ISWC 2004	sustav za upravljanje znanjem	1	budući da sustav koriste različiti timovi, potrebno je semantičko ujednačavanje podataka; sadrži repozitorij u koji se mogu pohranjivati multimedijalni podaci opisani meta-podacima	ne	ne zna se	ima osjetljivih – ne zna se kako se tretiraju	više	vlastiti	da	da	ne	mogućnost vizualnog prikaza grafa	ne	ne zna se	ne zna se
Platypus Wiki	ISWC 2004	sustav za upravljanje sadržajem	1	Tehnički detalji nisu dostupni										ne		
MuseumFinland	ISWC 2004	semantički web portal	1	integriran pregled nad kulturnom baštinom u muzejima Finske	da	ne zna se	ima osjetljive, ali ih ne tretira	više	Vlastite OWL ontologije	da - facetno pretraživanje i pretraživanje po ključnim riječima	da	ne	ne	ne	ne zna se	da
KM platform	ISWC 2004	Nije više dostupno														
Semantic portal on international affairs	ISWC 2004	semantički web portal	1, 3	semantičko pretraživanje automatski integriranih i označenih dokumenata	ne	RDQL	nema osjetljivih	više	Vlastita OWL ontologija	da - Vizualno i pretraživanje korištenjem prirodnog jezika	da	ne	ne	ne	ne zna se	ne zna se
UNSO	ISWC 2004	samostojeća aplikacija	1	Ne zadovoljava uvjete web aplikacije												
Semantic Web Assistant	ISWC 2004	sustav za rukovanje pravilima	1	Ne zadovoljava uvjete web aplikacije												
Swoogle	ISWC 2004	pretraživač	1	Ne zadovoljava uvjete web aplikacije												
Flink	ISWC 2004	semantički web portal	1	Integriran prikaz socijalne mreže istraživača semantičkog weba	ne	ne zna se	ima osjetljive, ali ih ne tretira	4	FOAF, SWRC i vlastite ontologije	ne	da	da	ne	da	da	da
Bibster	ISWC 2004	P2P sustav	1	Ne zadovoljava uvjete web aplikacije												

MOMIS	ISWC 2004	framework za ekstrakciju i integraciju informacija	1	Ne zadovoljava uvjete web aplikacije											
Annotea shared bookmarks	ISWC 2004	sustav za semantičko označavanje dokumenata	3	Ne zadovoljava uvjete web aplikacije											
GOHSE	ISWC 2004	sustav za semantičko označavanje dokumenata	3												
Pytypus	ISWC 2005	Nije više dostupno													
Web Services Execution Environment	ISWC 2005	arhitektura za podršku semantičkih web servisa	1	Ne zadovoljava uvjete web aplikacije											
DynamicView	ISWC 2005	semantički web portal	1	pregled nad podacima o znanstvenicima i područjima istraživanja	ne	ne zna se	nema osjetljivih	više	SKOS i vlastite	ne	da	ne	ne	da	ne zna se	ne zna se
Personal Publication Reader	ISWC 2005	semantički web portal	1		da	Tehnički detalji nisu dostupni										
Oyster	ISWC 2005	semantička klijentska aplikacija za P2P sustav	1	Ne zadovoljava uvjete web aplikacije											
FungalWeb	ISWC 2005	ontologija za potrebe bioinformatike i genetike	1	Ne zadovoljava uvjete web aplikacije											
CONFOTO	ISWC 2005	sustav za upravljanje znanjem	3	Pružanje mogućnosti označavanja slika i osoba uz bolje pretraživanje	ne	SPARQL	Ima osjetljive, ali ih ne tretira	više	RSS FOAF	ne	da	da	da – modul za označavanje	ne	Ne zna se	da
EKOSS	ISWC 2006	sustav za upravljanje znanjem	3	Tehnički detalji nisu dostupni										ne		
Geospatial Semantic Web Services	ISWC 2006	Nije više dostupno														
PaperPuppy	ISWC 2006	sustav za upravljanje znanjem	1	pregled autora i publikacija sa ISWC konferencija	ne	Ne zna se	Nema osjetljivih	1	Vlastita ontologija – proširuje FOAF	ne	da	ne zna se	ne	ne	ne zna se	ne zna se

SIGNS	ISWC 2006	Nije više dostupno														
Foafing the music	ISWC 2006	semantički web portal	1	personaliziran sustav koji nudi sadržaj na temelju preporuka	da	SPARQL	ima osjetljive, ali nema dobar sigurnosni mehanizam	više	RSS FOAF i vlastita OWL ontologija	ne	da	ne	ne	da	ne zna se	ne zna se
MultimediaN E-Culture demonstrator	ISWC 2006	semantički web portal	1	mogućnost pretraživanja i označavanja fotografija iz područja kulture	ne	SPARQL	nema osjetljivih	ne zna se	Vlastite RDF/OWL ontologije	da	da	da	ne	ne	ne zna se	Da
Dartgrid	ISWC 2006	programski okvir za integriranje relacijskih baza podataka	1	Ne zadovoljava uvjete web aplikacije												
PressIndex	ISWC 2006	sustav za upravljanje znanjem	1	semantička podloga za pretraživanje prirodnim jezikom	ne	ne zna se	nema osjetljivih	više	RSS Vlastita ontologija	Korištenjem prirodnog jezika i facetno pretraživanje	da	ne	ne	ne	ne zna se	ne zna se
Semantic MediaWiki	ISWC 2006	sustav za upravljanje sadržajem	3	Tehnički detalji nisu dostupni										ne		
Collimator	ISWC 2006	sustav za označavanje	3	Ne zadovoljava uvjete web aplikacije												
Falcon-S	ISWC 2006	semantički web portal	1	označavanje i pretraživanje fotografija vezanih uz nogomet	ne	SeRQL	nema osjetljivih	službene stranice FIFA i UEFA	vlastita OWL ontologija	napredno semantičko i vizualno pretraživanje	da	ne	ne	da	ne zna se	da
COHSE	ISWC 2006	sustav za označavanje	3	Ne zadovoljava uvjete web aplikacije												
ArnetMiner	ISWC 2007	pretraživač	1	Ne zadovoljava uvjete web aplikacije												
Cantabria	ISWC 2007, W3 lista	semantički web portal	1	integriran pregled podataka kulturne baštine regije Kantabrija	ne	ne zna se	nema osjetljivih	više	standardne iz područja kulture	napredno semantičko i vizualno pretraživanje	da	ne	ne	ne	ne zna se	ne zna se
CHIP	ISWC 2007	semantički web portal	1	Preporuke i kreiranje personaliziranih muzejskih tura	da	SeRQL	nema osjetljivih	više	preuzete iz prethodnih projekata iste domene	da	da	ne	ne	ne	ne zna se	da

Doris	ISWC 2007	sustav za upravljanje znanjem	3	sustav za automatsko izdvajanje znanja, označavanje i pretraživanje dokumenata	ne	ne zna se	nema osjetljivih	1	vlastite ontologije: o korisnicima i ontologija domene	da	da	ne	ne	da	Ne zna sse	da
EachWiki	ISWC 2007	sustav za upravljanje znanjem	3	sugerira pojmove i veze za označavanje na temelju sadržaja Wikipedije	Tehnički detalji nisu dostupni									ne		
GroupMe!	ISWC 2007	aplikacija posebne primjene - označavanje	3	aplikacija za grupiranje i označavanje multimedije	da	ne zna se	ima osjetljive, ne zna se kako ih tretira	1	FOAF RSS DCMI i vlastita ontologija	ne zna se	ne	da	ne	ne	ne zna se	ne zna se
JeromeDL	ISWC 2007	servisi za knjižnice	3	Ne zadovoljava uvjete web aplikacije												
Media Watch	ISWC 2007	semantički web portal	1,3	integriranje i označavanje geoprostornih podataka	ne zna se	ne zna se	ne zna se	više	vlastita OWL ontologija	ne zna se	ne zna se	ne zna se	ne zna se	da	ne zna se	ne zna se
mle	ISWC 2007	aplikacija posebne primjene	1,3	Aplikacija za arhiviranje emaila	Ne zadovoljava uvjete web aplikacije											
Notitio.us	ISWC 2007	servis za agregiranje podataka iz socijalnih aplikacija	3	Ne zadovoljava uvjete web aplikacije												
Potluck	ISWC 2007	aplikacija posebne primjene – vizualizacija	1	Aplikacija za facetno pregledavanja i povezivanje podataka sa više web stranica	Ne zadovoljava uvjete web aplikacije											
Revyu.com	ISWC 2007	aplikacija posebne primjene - označavanje	2,3	Aplikacija za evidentiranje ocjena i komentara	ne	SPARQL	nema osjetljivih	1	YAGO FOAF DBpedia	pretraživanje po tagovima	ne	da	da	ne	ne zna se	ne zna se
RKB Explorer	ISWC 2007	semantički web portal	1,3	integriran pregled podataka o istraživačima institucijama i radovima iz područja računarstva	ne	SPARQL	ne zna se	više	vlastita ontologija	facetno pretraživanje	ne zna se	ne zna se	ne zna se	ne zna se	ne zna se	da
int.ere.st	ISWC 2007, ISWC 2008	aplikacija posebne primjene	3	Sustav za označavanje, zamišljeno da radi globalno na webu	da	Tehnički detalji nisu dostupni			SIOC FOAF SCOT	Tehnički detalji nisu dostupni				ne	Tehnički detalji nisu dostupni	

SemClip	ISWC 2007	aplikacija posebne primjene	3	Aplikacija omogućava semantičko kopiranje podataka između različitih aplikacija	Ne zadovoljava uvjete web aplikacije										
iFanzy	ISWC 2007	semantički web portal	1	Personaliziran TV vodič	da	ne zna se	nema osjetljivih	više	SKOS	obično i facetno pretraživanje	da	ne	ne	ne	ne zna se	da
SMART	ISWC 2007	aplikacija posebne primjene – pretraživanje i zaključivanje	1	aplikacija za pretraživanje bioloških podataka	ne	SPARQL	nema osjetljivih	1	vlastita OWL ontologija	sugerirano pretraživanje (kontrolirani unos)	da	ne	ne	ne	ne zna se	ne
SWSE	ISWC 2007, ISWC 2009	semantička tražilica	2	Ne zadovoljava uvjete web aplikacije											
WWWatch	ISWC 2007	aplikacija posebne primjene - vizualizacija	1	vizualno prikazivanje relacija između država i prirodnih pojava kao posljedica globalnog zatopljivanja različitih država	ne	ne zna se	nema osjetljivih	više	vlastita OWL ontologija	da (specifično s obzirom na karakteristiku aplikacije)	ne	ne	ne	ne	ne zna se	da
Bio2RDF	ISWC 2008	baza genoma	2	Ne zadovoljava uvjete web aplikacije											
CardioSHARE	ISWC 2008	framework za vršenje upita nad distribuiranim podacima	1	Ne zadovoljava uvjete web aplikacije											
CultureSampo	ISWC 2008	semantički web portal	1	integriran pregled na sadržaj više od 20 finskih muzeja, kao i različite druge izvore	ne	SPARQL	nema osjetljivih	više	ONKI – ontologija finske kulture	različita pretraživanja ovisno o kategoriji	ne zna se	da	ne	ne	ne zna se	ne zna se
DBpedia mobile	ISWC 2008	aplikacija posebne primjene	2	Namijenjena turistima, prati lokaciju korisnika, omogućava pristup DBpedia podacima, kao i pisanje komentara preko Revyu aplikacije, ima pristup još nekim Lined Data izvorima	Ne zadovoljava uvjete web aplikacije										

DERI Pipes	ISWC 2008	aplikacija posebne primjene	2	Mashup alat	Ne zadovoljava uvjete web aplikacije											
DSS	ISWC 2008	servisi kao podrška označavanje	3	Sugeriranje oznaka	Ne zadovoljava uvjete web aplikacije											
Falcons	ISWC 2008	semantička tražilica	2	Ne zadovoljava uvjete web aplikacije												
HealthFinland	ISWC 2008	Sustav se sastoji od tri dijela (i) centralizirane usluge zdravstvene ontologije te pomoćnih alata, (ii) kanal za stvaranje semantičkog sadržaja baziran na nekoliko zdravstvenih organizacija i (iii) semantički portal koji slaže i predstavlja sadržaj iz intuitivne perspektive krajnjeg korisnika														
IYOUIT	ISWC 2008	društvena aplikacija	3	društvena aplikacija za web i mobilne uređaje	da	Ne zna se	Ima osjetljivih	više	Vlastita ontologija	ne zna se	da	da	ne	ne	ne zna se	ne zna se
LODr	ISWC 2008	aplikacija za označavanje	2,3	Aplikacija za označavanje sadržaja iz poznatijih socijalanih aplikacija po principima LInked Data	Tehnički detalji nisu dostupni									ne	Tehnički detalji nisu dostupni	
paggr	ISWC 2008	aplikacija posebne primjene	2	Aplikacija za kreiranje semantičkih web portala	Tehnički detalji nisu dostupni											
xOperator	ISWC 2008	semantički agent	1	Ne zadovoljava uvjete web aplikacije												
MT-Redland	SFSW2 005	sustav za upravljanje znanjem	1	objavljivanje blogova uz bolje pretraživanje	ne	SPARQL	ima i osjetljivih (e-mail, lozinka), ali i podaci o ljudima	1	RSS FOAF SIOC DC	da	ne	ima, ali se ne koristi	ne	ne	da Data mapper	RDF
SIMAC: Semantic Interaction with music Audio Content	SFSW2 005	aplikacija posebne primjene – preporučen sadržaj	1	na osnovu FOAF profila aplikacija nudi glazbenike koji bi korisniku mogli zanimati	da – traženje preporuka	RDQL	nema osjetljivih	više	FOAF	ne	ne zna se	ne	ne	da	ne zna se	da
HyperJournal	SFSW2 005	P2P sharing system	1	Ne zadovoljava uvjete web aplikacije												Sesame
Arago	SFSW2 005	framework		Alat za generiranje prezentacije iz RDF opisa	Ne zadovoljava uvjete web aplikacije											
pOWL	ISWC 2004, SFSW2 005	alat za upravljanje ontologijom	1	Ne zadovoljava uvjete web aplikacije												

RDFHomepage	SFSW2006	programski okvir		programski okvir za generiranje HTML stranica iz RDF opisa	Ne zadovoljava uvjete web aplikacije											
DBin	ISWC 2004, SFSW2006, ISWC 2006	P2P semantička klijentska aplikacija	1	Ne zadovoljava uvjete web aplikacije												
FOAFMap	SFSW2006	aplikacija posebne primjene	1	Prikaz lokacije korisnika na karti, kao i osnovnih informacija iz profila uz mogućnost označavanja na karti	ne	ne zna se	ima osjetljive, ali ih ne tretira	1	FOAF	da – slični korisnici po označenim lokacijama	ne	ne	ne	ne	ne zna se	datoteka
Lego-Note	SFSW2006	Nije više dostupno														
RDFRoom	SFSW2006	vizualizacija semantičkih podataka		igra	Ne zadovoljava uvjete web aplikacije											
eMerges	SFSW2006, ISWC 2006	semantički web portal	1	Prikazivanje geo informacija na karti (informacije o lokacijama i sl.)	ne	ne zna se	nema osjetljivih	1	vlastiti	ne	ne	ne	ne	ne	ne	ne zna se
DOAP Store	ISWC 2003, SFSW2007	semantički web portal	1	pretraživanje weba, prikazivanje podataka o svim dohvaćenim projektima opisanim DOAP vokabularom	ne	SPARQL	ima osjetljive, ali ih ne tretira	web	DOAP	da	ne	ne	ne	da	ne	da
SIOC explorer	SFSW2007	semantički portal	1	integrira više SIOC izvora i daje njihov pregled	ne	ne zna se	ima osjetljive, ali ih ne tretira	web	SIOC SKOS FOAF DC	ne	ne	ne	da – facetno pretraživane	da	da	ne zna se
Ripple	SFSW2007	skriptni jezik	2	skriptni jezik za kreiranje "linked data" podataka	Ne zadovoljava uvjete web aplikacije											
Sindice	SFSW2007	aplikacija posebne primjene	1	izrada baze indeksa semantičkog weba	Ne zadovoljava uvjete web aplikacije											
LDAP 2 SPARQL	SFSW2007	samostojeća aplikacija	1	Preslikavanje između SPARQL-a i LDAP-a	Ne zadovoljava uvjete web aplikacije											
RDF2FS	SFSW2008	samostojeća aplikacija		Pohranjuje RDF grafove u direktorije UNIX operacijskog sustava	Ne zadovoljava uvjete web aplikacije											

SMOB	SFSW2008	decentralizirani sustav za objavljivanje mikroblogova	3	klijent -server rješenje za objavljivanje mikro blogova na sem. webu	Ne zadovoljava uvjete web aplikacije
SPARQLBot	SFSW2008	aplikacija posebne primjene	2	Aplikacija za pretraživanje povezanih podataka putem definiranih komandi	Tehnički detalji nisu dostupni
MOAW	SFSW2008	Web komponenta	2,3	Komponenta za autocomplete HTML tekst polja	Ne zadovoljava uvjete web aplikacije
Ruby SemPipes	SFSW2008	mehanizam za kombiniranje informacija		Rješenje kao Yahoo! Pipes samo za semantički web	Ne zadovoljava uvjete web aplikacije
OzoneBrowser	SFSW2009	preglednik	3	preglednik koristi oznake sa prezentacijskog sloja u svrhu bolje korisničke interakcije	Ne zadovoljava uvjete web aplikacije
Researcher's map	SFSW2009	aplikacija posebne primjene	2	aplikacija na karti prikazuje znanstvenike Njemačke s mogućnošću pretrage njihovih publikacija	Tehnički detalji nisu dostupni
RDF Schema Rendering	SFSW2009	generator HTML prikaza	3	izgrađuje HTML prikaz iz RDF-a	Ne zadovoljava uvjete web aplikacije
AJAX Widget for Semantic Query Expansion	SFSW2009	Widget za sugeriranje pri pretraživanju	2	Ne zadovoljava uvjete web aplikacije	
Colibrary	SFSW2009	Web servis	2	Web servis za dohvaćanje informacija o knjizi na temelju URI-a	Ne zadovoljava uvjete web aplikacije
Practical Semantic Works	SFSW2009	Plugin za preglednik	2,3	Ne zadovoljava uvjete web aplikacije	
SPARCool	SFSW2009	Web servis za izvođenje SPARQL upita preko URL uzorka	2	Ne zadovoljava uvjete web aplikacije	

Tal4RDF	SFSW2009	jezik temeljen na predlošcima za pretvaranje RDF formata u druge	Ne zadovoljava uvjete web aplikacije													
A Digital Music Archive (DMA) for the Norwegian National Broadcaster (NRK)	W3 lista (2007)	implementacija semantičkih tehnologija za bolje arhiviranje primjenom oznaka	3	Ne zadovoljava uvjete web aplikacije												
A Linked Open Data Resource List Management Tool for Undergraduate Students	W3 lista (2009)	aplikacija posebne primjene	2,3	Aplikacija omogućava pregled literature potrebne za kolegije na studiju, kao i pisanje komentara koji služe profesorima kao povratne informacije	da	SPARQL	sprema odvojeno privatne i javne podatke	više	vlastite ontologije	ne zna se	ne	da	da	ne	ne zna se	da
A Semantic Web Content Repository for Clinical Research	W3 lista (2007)	sustav za upravljanje znanjem	1	sustav za integriranje podataka o pacijentima i njihovo pregledavanje i pretraživanje korištenjem stručne terminologije	Tehnički detalji nisu dostupni											
An Intelligent Search Engine for Online Services for Public Administrations	W3 lista (2007)	aplikacija posebne primjene	1	napredno pretraživanje usluga koje pruža grad Zaragoza	ne zna se				vlastite ontologije	pretraživanje korištenjem prirodnog jezika	ne zna se					
PharmaSURVEYOR	W3 lista (2007)	semantički ekspertni web sustav	1	napredno pretraživanje i zaključivanje o lijekovima i njihovim nuspojavama	Tehnički detalji nisu dostupni.											
CRUZAR	W3 lista (2008)	aplikacija posebne primjene	1	Integracija više izvora o kulturnim dobrima Zaragoze, te sugeriranje tura za razgledavanje	da	ne zna se	nema osjetljivih	više	vlastita ontologija	ne	da	da	ne	ne	ne zna se	ne

Enhancement and Integration of Corporate Social Software Using the Semantic Web	W3 lista (2008)	sustav za upravljanje znanjem	3	Aplikacija za potrebe R&D odjela francuske tvrtke – uključuje blogove, wiki, označavanje, i sl.	ne zna se	SPARQL	ne zna se	ne zna se	SIOC FOAF DC MOAT	ne zna se	ne	ne	ne	ne	ne zna se	da
Enhancing Content Search Using the Semantic Web	W3 lista (2007)	semantički web portal	1	Semantički web porttal sa poboljšanim pretraživanjem i vizualizacijom	da	SPARQL	ne zna se	2	ne zna se	da	Ne zna se	ne	ne	ne	ne zna se	da
Ordnance survey	W3 lista (2007)	ontologija za geo podatke Velike Britanije	1	Ne zadovoljava uvjete web aplikacije												
Segala	W3 lista (2007)	framework sa objavljivanje pouzdanih web lokacija	Ne zadovoljava uvjete web aplikacije													
SearchMonkey	W3 lista (2008)	semantička web tražilica	3	Ne zadovoljava uvjete web aplikacije												
KDE 4.0 Semantic Desktop Search and Tagging	W3 lista (2008)	semantička klijentska aplikacija		Semantičko označavanje dokumenata	Ne zadovoljava uvjete web aplikacije											
POPS	W3 lista (2008)	aplikacija posebne primjene – klijentska aplikacija	2	Aplikacija ima namjenu lakšeg pronalaženja zaposlenika NASA-e sa određenim radnim iskustvom	Ne zadovoljava uvjete web aplikacije											
Prioritization of Biological Targets for Drug Discovery	W3 lista (2007)	semantička klijentska aplikacija	1	Ne zadovoljava uvjete web aplikacije												
Real Time Suggestion of Related Ideas in the Financial Industry	W3 lista (2007)	aplikacija posebne primjene	1	Aplikacija je namijenjena kadrovskoj službi; zaposlenik unosi primjedbe i automatski se traži povezanost sa financijskim sektorom	Tehnički detalji nisu dostupni											
Semantic Content Description to Improve Discovery	W3 lista (2007)	ontologija za opis sadržaja na mobitelima	1	Ne zadovoljava uvjete web aplikacije												

Semantic MDR and IR for National Archives	W3 lista (2008)	tražilica na zadanom integriranom skupu podataka	1	Ne zadovoljava uvjete web aplikacije												
Faviki	W3 lista (2008)	aplikacija posebne primjene - označavanje	3	Aplikacija za označavanje pojmovima iz DBpedije	Tehnički detalji nisu dostupni											
Semantic Web Technology for Public Health Situation Awareness	W3 lista (2007)	sustav za integraciju podataka iz područja zdravstva	1	Ne zadovoljava uvjete web aplikacije												
Semantic-based Search and Query System for the Traditional Chinese Medicine Community	W3 lista (2007)	sustav za integraciju podataka iz područja zdravstva u Kini	1	Ne zadovoljava uvjete web aplikacije												
The Semantic Web for the Agricultural Domain, Semantic Navigation of Food, Nutrition and Agriculture Journal	W3 lista (2007)	sustav posebne primjene	1	Sustav za bolje pretraživanje podataka iz područja agrikulture	Ne zadovoljava uvjete web aplikacije											
The swoRDFish Metadata Initiative: Better, Faster, Smarter Web Content	W3 lista (2007)	alat za označavanje	3	Ne zadovoljava uvjete web aplikacije												
twine	W3 lista (2009)	semantička društvena web aplikacija	3	društvena aplikacija za označavanje i agregiranje sadržaja ovisno o oznakama	**da**	SPARQL	neki podaci su dostupni samo neko j grupi korisnika	više	Vlastita ontologija	**da** – korištenjem oznaka	**da**	**ne**	**ne**	**ne**	ne zna se	**da**
Use of Semantic Web Technologies in Natural language interface to Business Applications	W3 lista (2007)	aplikacija posebne primjene	1	aplikacija za pretraživanje prirodnim jezikom	Ne zadovoljava uvjete web aplikacije											