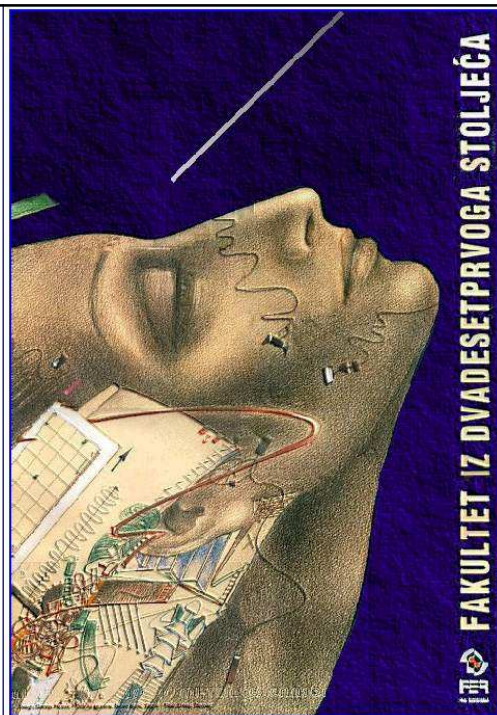


Napredni modeli i baze podataka

Predavanja
siječanj 2009.

Integracija podataka iz različitih izvora - semantički web



Semantički web i integracija heterogenih izvora podataka

- Što je web?
- Što je semantički web?
- Što su semantičke web tehnologije?
- Kako semantičke web tehnologije mogu pomoći pri integraciji heterogenih izvora podataka?

Što je Web?

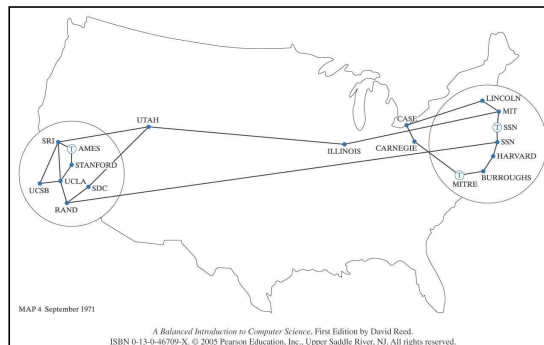
- Često se miješaju pojmovi **web** i **Internet**, odnosno smatraju se sinonimima
 - Internet je mrežna infrastruktura
 - Web (World Wide Web) skup dokumenata pohranjen na serverima koji su povezani Internetom, te koji su dostupni preko HTTP protokola

Povijest Internet-a (1)

- ~1960
 - profesor sa MIT J.C.R. Licklider objavio je niz članaka koji opisuju "Galactic Network" – mrežu računala koja omogućava korisnicima prikupljanje podataka i pristup programima od bilo kud u svijetu (spominje i socijalnu interakciju putem mreže)
 - 1962. Licklider postaje voditelj odjela za računalna istraživanja pri ministarstvu obrane SAD-a - *Advanced Research Project Agency (ARPA)*
 - 1967. Licklider zapošljava Larry Roberts-a da dizajnira i implementira viziju mreže "Galactic Network"
- ARPANet (prethodnik Interneta) postao je stvarnost 1969.
 - povezivao je računala na četiri fakulteta: UCLA, UCSB, SRI i Utah
 - promet se odvijao kabelima - brzina prijenosa je bila 56K bits/sec
 - ARPANet je bio pokazatelj mogućnosti da istraživači na različitim lokacijama mogu komunicirati, dijeliti podatke i pokretati programe s udaljenih lokacija

Povijest Internet-a (2)

- ARPANet je imao za cilj samo povezati vojne institucije i one fakultete koji su sudjelovali u vladinim projektima
 - do 1971. 18 lokacija je bilo povezano koristeći Interface Message Processors (IMPs) koji je omogućavao spajanje 4 računala na pojedinoj lokaciji
 - lokacije označene sa T koristile su Terminal Interface Processors (TIPs), koji je dozvoljavao spajanje do 64 računala na pojedinoj lokaciji

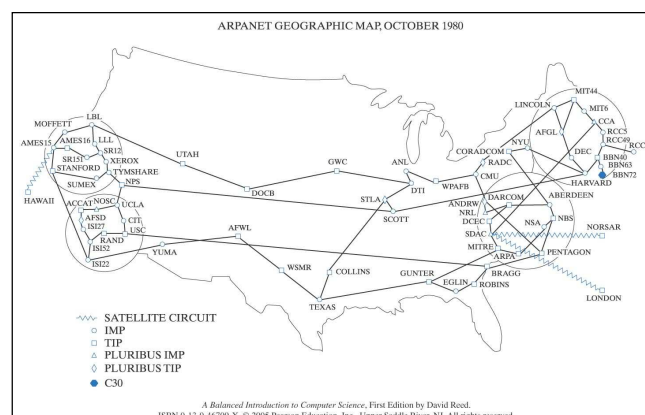


© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Povijest Internet-a (3)

- do 1980. blizu 100 lokacija je bilo povezano ARPANet-om
 - satelitske veze su omogućavale spajanje sa lokacijama izvan kontinentalnog dijela SAD-a

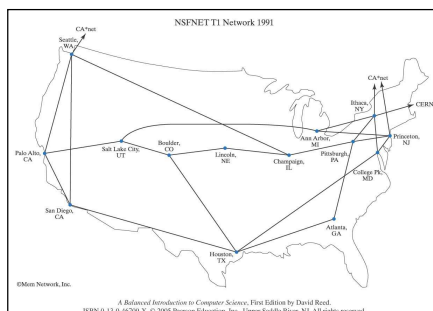


© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Povijest Internet-a (4)

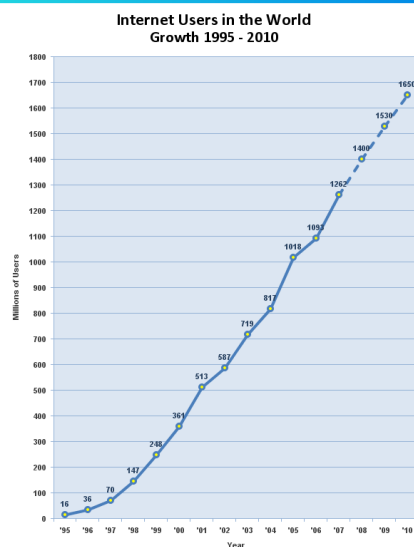
- tijekom 80ih
 - mrežne aplikacije kao e-mail i newsgroup-e postaju sve popularnije među populacijom koja je umrežena (ostvaruje se predviđen social networking)
 - do 1984. ARPANet je sadržavao više od 1000 računala
- da bi se omogućio daljnji rast, National Science Foundation (NSF) uključuje se u ARPANet 1984.
 - NSF je financirao konstrukciju vodova velikih brzina koji su činili temelj današnje mreže



© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Internet danas

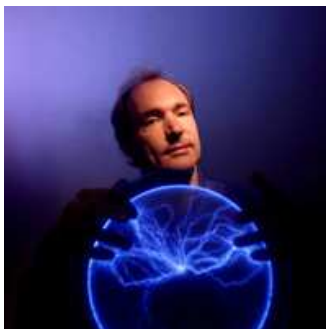


© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Web

- World Wide Web često se kraće naziva Web
- Prema W3C-u Web je:
 - Informacijski prostor u kojem su smješteni izvori informacija identificirani URI-em (Uniform Resource Identifier)



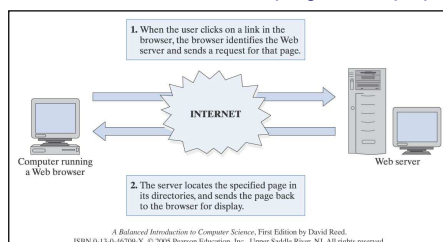
Tim Berners Lee

© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Povijest Web-a (1)

- World Wide Web je multimedijalno okruženje u kojem dokumenti mogu biti povezni putem Interneta
 - predložen od strane Tim Berners-Lee u CERN-u 1989.
 - dizajniran da bi omogućio dijeljenje informacija između istraživača lociranih diljem Europe koji pritom koriste različita računala i programsku potporu



Osnovne ideje:

1. hipertekst (dokumenti sa međusobno povezanim sadržajem)
 - Web stranice mogu sadržavati slike i veze na druge stranice
2. uz pomoć interneta
 - stranice mogu biti pohranjene na računalima diljem interneta (*Web serveri*)
 - logičke veze između dokumenata neovisne su o fizičkim vezama

© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Povijest Web-a (2)

- **1990:** Berners-Lee je proizveo radne prototipe Web preglednika i servera
- **1991:** napravljena prva web stranica
- **1993:** kreiran je prvi vizualni preglednik: Mosaic
 - Mosaic je integrirao tekst, slike i veze, te je tako surfanje postalo intuitivno
- **1994:** osnovan je Netscape, koji je proizveo još jedan popularan preglednik: Netscape Navigator
- **1995:** Microsoft je proizveo Internet Explorer → počinje rat preglednika!; osniva se Amazon.com; izlazi Java Applet; izlazi Netscape preglednik 2.0B3 sa podrškom za JavaScript
- **1996:** osniva se Google; izlazi Macromedia Flash 1.0; proizvodi se IntelliMouse (miš s kotačem); izlazi Internet Explorer 3.0 sa podrškom za JavaScript
- **1997:** izlazi Hotmail, besplatni e-mail server
- **1999:** izlazi Napster
- **1998:** pojavljuju se prvi blogovi; osnovan je PayPal; Amazon kupuje Imdb
- **2000:** *Tim Berners Lee predstavlja viziju semantičkog web-a*

Semantički web - definicija

- Semantički web je vizija: ideja o mogućnosti da podaci na web-u budu definirani i povezani na način na koji bi se mogli koristiti osim za sami prikaz i za automatizaciju, integraciju i ponovnu iskoristivost u različitim aplikacijama. Na taj način obećava se mogućnost pronalaženja, sortiranja i klasificiranja informacija, dakle svih onih zadataka koje troše puno vremena provedenog on i off-line [W3C].

Semantički web

- daje značenje (opis) sadržaju na web-u
- sadržaj mora biti u takvom obliku da ga računala mogu procesirati, te mogu izvući veze između pronađenih informacija
- korištenjem tehnologija semantičkog web-a (RDF, OWL, RDFa, ...) omogućava se korisnicima davanje značenja podacima

Semantički web – davanje značenja podacima



Daljnji razvoj web-a

- 2001: stvorena je wikipedia
- 2003: osnovan je MySpace, Web 2.0
- 2004: Google najavljuje Gmail; izlazi Firefox 1.0
- 2005: osnovan i predstavljen on-line YouTube; Yahoo kupuje Flickr; pojava Ajax-a; eRDF
- 2007: Facebook (boom socijalnih web aplikacija)
- 2008: RDFa

Od vizije do realizacije

- Tim Berners Lee je očekivao da će razvoj trajati 10 godina, no već u dosadašnjih 8 godina primarne tehnologije web-a se znatno mijenjaju, pa im se prilagođava i vizija semantičkog web-a
 - socijalne web aplikacije postaju jako popularne
 - gotovo sve web aplikacije su “rich internet” aplikacije (Ajax, Web 2.0.)
- Veliki problem semantičkog web-a → još uvijek ne postoji “killer app”, koja bi dovela velik broj korisnika (a i programera)

Semantički web danas (1)

- Počinju se razlikovati dvije vizije:
 - data web
 - otvaranje sadržaja baza podataka za web
 - potpunije rješenje
 - annotated web (potiče ga Web 2.0 i socijalne aplikacije)
 - označavanje podataka na web-u
 - jednostavna semantika
- Trenutno se više razvija druga vizija:
 - osobe uključene u razvoj web aplikacija više teže brzom i prljavom rješenju, nego dugoročnijem i čistijem rješenju

Semantički web danas (2)

- WWW je bio brzo i široko prihvaćen, prvenstveno zbog sljedećih razloga:
 - jednostavan za korištenje
 - jednostavan za kreiranje
- Semantičkom webu nedostaju alati (pravila) koji bi pridonijeli jednostavnosti korištenja i njegovog kreiranja

Motivacija za ideju semantičkog web-a (1)

- Najzastupljeniji primjer:
 - pretraživači (Google i sl.) ipak vrte velik broj irelevantnih rezultata; pridodavanje značenja podacima na web-u trebalo bi poboljšati rezultate

Rješenja:

- Yahoo izdaje SearchMonkey koji potiče korištenje anotacija, kao što je RDFa (<http://developer.yahoo.com/searchmonkey/>)

Motivacija za ideju semantičkog web-a (2)

- Obično se trebaju kombinirati podaci na web-u:
 - primjer organizacije putovanja:
 - na jednoj stranici se nalaze podaci o smještaju, na drugoj o prijevozu, a na trećoj o zanimljivostima lokacije...
- Teži se automatizaciji takvih procesa
 - ljudi bez problema kombiniraju informacije iz različitih izvora, čak i kada se koriste različite terminologije, no treba to isto omogućiti i računalima

Motivacija za ideju semantičkog web-a (3)

- prenosivost podataka
- ovo je problem novijeg datuma, javio se kao rezultat velikog broja korisnika socijalnih aplikacija

- Primjer:
 - omogućiti prenosivost profila i mreže prijatelja između različitih aplikacija da korisnici ne bi trebali ponovno unositi vlastite podatke

W3C - World Wide Web Consortium

- osnovan je 1994.
- W3C je međunarodni konzorcij pri kojem organizacije članice, zaposlenici W3C-a i javnost rade zajedno na razvoju Web standarda
- cilj W3C-a je ostvariti puni potencijal WWW-a razvijajući protokole i smjernice koje bi osigurale dugoročan rast Web-a
- W3C se brine da temeljne tehnologije WWW-a budu međusobno kompatibilne
- objavljivanjem otvorenih web standarda (za web jezike, protokole) nastoji se izbjeći tržišna, odnosno Web fragmentacija

- Primjeri W3C dokumenata:
 - HTML 4.01 Specification – 24.12.1999. (W3C recommendation)
 - XForms 1.1 – 29.11.2007. (W3C candidate recommendation)
 - SPARQL Protocol for RDF – 15.1.2008. (W3C recommendation)
 - HTML 5 – 10.6.2008. (W3C working draft)

Integracija heterogenih izvora podataka

- podaci se nalaze u različitim oblicima
 - polustrukturirani (web dokumenti)
 - strukturirani (XML dokumenti, baze podataka, ...)
- tehnologije semantičkog web-a mogu pomoći pri integraciji heterogenih izvora podataka

Integracija baza podataka

- problem integracije je u tom što su baze podataka obično heterogene po strukturi i sadržaju
- Zašto (kada) je potrebna integracija?
 - velik broj servisa zahtjeva rukovanje podacima iz više baza podataka
 - nakon udruživanja tvrtki
 - povezivanje raznih administrativnih podataka (npr. za porez)
 - biološka, genetska, farmaceutska i medicinska istraživanja
 - itd...

Tehnologije semantičkog web-a

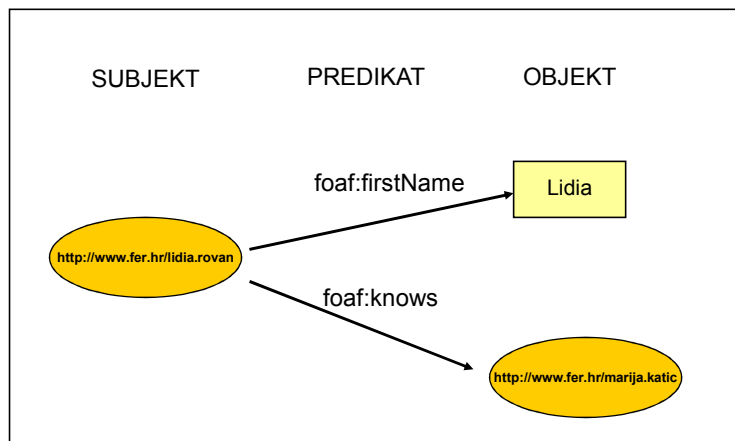
- RDF
- RDFS
- OWL
- RDFa
- SPARQL

RDF – Resource Description Framework

- namijenjen za spremanje podataka o podacima
- prilagođen računalima
- temelji se na konceptu trojki: subjekt – predikat – objekt
 - **subjekt** ima određeno svojstvo (**predikat**) čija je vrijednost **objekt**
 - subjekt i predikat su URI-ji, dok objekt može biti ili literal ili URI
- RDF trojka predstavlja izjavu
- RDF notacija
 - trojna notacija (N3)
 - XML notacija
 - vizualni prikaz u obliku grafa

RDF model (1)

RDF model



© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

RDF model (2)

XML

```

<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:knows rdf:resource="http://www.fer.hr/marija.katic"/>
    <foaf:firstName>Lidia</foaf:firstName>
  </rdf:Description>
</rdf:RDF>
  
```

N3

```

<http://www.fer.hr/lidia.rovan>
  <http://xmlns.com/foaf/0.1/firstName>
    "Lidia" ;
  <http://xmlns.com/foaf/0.1/knows>
    <http://www.fer.hr/marija.katic> .
  
```

© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

RDF model – kreiranje izjava

```
String personURI    = "http://www.fer.hr/lidia.rovan";
String firstName    = "Lidia";
String friendURI    = "http://www.fer.hr/marija.katic";

// kreiranje praznog modela
Model model = ModelFactory.createDefaultModel();

// dodavanje izjava u model
model.createResource(personURI)
    .addProperty(ResourceFactory.createProperty("http://xmlns.com/foaf/0.1/", "firstName"),
        firstName)
    .addProperty(ResourceFactory.createProperty("http://xmlns.com/foaf/0.1/", "knows"),
        ResourceFactory.createResource(friendURI));
```

RDFS – RDF Schema

- definira značenja, karakteristike i odnose između dozvoljenih termina - omogućuje kreiranje vlastitog RDF rječnika
- prvi korak do “dodatnog znanja”
- omogućava izvođenje zaključaka iz RDFS pravila

Osnovni elementi:

- rdfs:Resource
- rdfs:Class
- rdfs:Literal
- rdfs:Property
- rdfs:domain
- rdfs:range
- rdfs:subClassOf
- rdfs:subPropertyOf

RDFS – primjer

```
<rdf:Property rdf:about="http://xmlns.com/foaf/0.1/firstName"
  rdfs:label="firstName" rdfs:comment="The first name of a person.">
  <rdf:type rdf:resource = "http://www.w3.org/2002/07/owl#Datatype property"/>
  <rdf:domain rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
  <rdf:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
  <rdf:isDefinedBy rdf:resource="http://xmlns.com/foaf/0.1/">
</rdf:Property>
```

SPARQL (1)

- SPARQL = SPARQL Protocol and RDF Query Language
- upitni jezik za RDF

4 tipa upita za čitanje podataka:

- SELECT
- ASK
- DESCRIBE
- CONSTRUCT

SPARQL (2)

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

SPARQL primjer (1)

SELECT

```
PREFIX foaf: http://xmlns.com/foaf/0.1/
SELECT DISTINCT ?name ?mbox
WHERE { ?x foaf:name ?name;
        foaf:mbox ?mbox }
ORDER BY DESC(?name)
LIMIT 5
OFFSET 2
```

CONSTRUCT

```
PREFIX data: http://example.org/
PREFIX foaf: http://xmlns.com/foaf/0.1/
PREFIX vcard: <http://www.w3.org/2001/vcard-rdf/3.0#>
FROM <http://example.org/dft.ttl>
FROM NAMED <http://example.org/alice>
FROM NAMED http://example.org/bob
CONSTRUCT { ?x vcard:givenName ?name;
             vcard:email ?mbox . }
WHERE { ?x foaf:name ?name;
        foaf:mbox ?mbox . }
```

SPARUL

- SPARUL = SPARQL/Update = Update Language For Rdf Graphs
- Vrste naredbi:
 - INSERT
 - DELETE
 - MODIFY
 - CLEAR
 - LOAD
 - CREATE GRAPH
 - DROP GRAPH

SPARQL protokol

- SPARQL Protocol je jednostavna metoda za postavljanje SPARQL upita i primanje rezultata preko HTTP-a. SPARQL URL se sastoji od tri dijela:
 1. URL SPARQL endpoint-a
npr. `http://dbpedia.org/sparql`
 2. (Opcionalno, kao dio niza koji čini upit) Imena grafova nad kojima se postavlja upit
npr. `default-graph-uri=http://dbpedia.org`
 3. (Kao dio niza koji čini upit) Sam SPARQL upit
npr. `Query=SELECT distinct ?x
WHERE { ?x a <http://umbel.org/umbel/sc/Artist> }`

Primjer SPARQL upita:

```
http://dbpedia.org/sparql?default-graph-
uri=http%3A%2F%2Fdbpedia.org&query=SELECT+distinct+%
3Fx+WHERE+%3F%3Fx+a+%3Chttp%3A%2F%2Fumbel.org%2Fumbel%
2Fsc%2FArtist%3E+
```

Ontologije

- Ontologija je model podataka koji predstavlja skup pojmova unutar neke domene i veze između tih pojmova
- Jezici ontologija:
 - RDF – iskazivanje tvrdnji
 - RDFS – opisivanje vokabulara
 - OWL – opisivanje veza između vokabulara, ograničenja, karakteristika...

OWL – Web Ontology Language

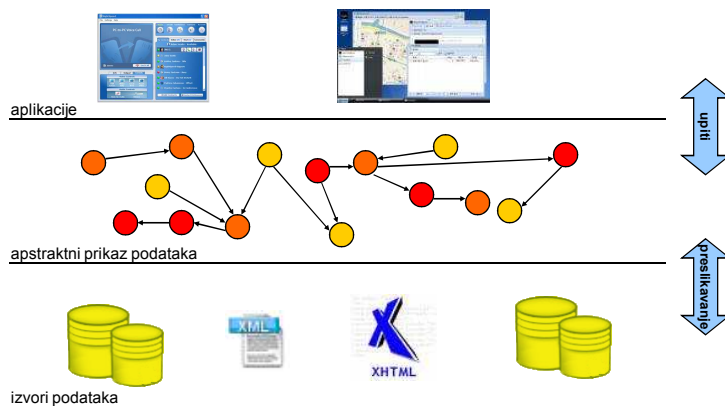
- Verzije OWL-a:
 - OWL Lite
 - OWL DL
 - OWL Full
- karakteristike svojstava (TransitiveProperty, SymmetricProperty, InverseOf, ...)
- ograničenja svojstava (minCardinality, maxCardinality, ...)
- preslikavanja (equivalentClass, equivalentProperty, sameAs, ...)
- složeni razredi (intersectionOf, unionOf, ...)

OWL - primjer

```
<owl:SymmetricProperty
  rdf:about="http://purl.org/vocab/relationship/friendOf">
  <rdfs:isDefinedBy rdf:resource="&rel;" />
  <rdfs:subPropertyOf
    rdf:resource="http://xmlns.com/foaf/0.1/knows"
    rdfs:label="knows" />
  <rdfs:label xml:lang="en">Friend Of</rdfs:label>
  <rdfs:comment xml:lang="en">A person who shares mutual
  friendship with this person.</rdfs:comment>
  <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Person"
    rdfs:label="Person" />
  <rdfs:range rdf:resource="http://xmlns.com/foaf/0.1/Person"
    rdfs:label="Person" />
  <rdfs:isDefinedBy
    rdf:resource="http://purl.org/vocab/relationship/" />
</owl:SymmetricProperty>
```

Integracija podataka

- Osnovni koraci integracije podataka:
 1. dati značenje podacima – preslikati ih u uvriježeno značenje
 2. udružiti preslikane podatke
 3. obavljanje upita nad cijelim skupom podataka



Integracija podataka – primjer (1)

- Pretpostavimo da treba integrirati podatke iz sljedećih izvora:
 - relacijska baza bp1
 - relacijska baza bp2
 - Web 2.0 aplikacija sa sučeljem u XHTML-u
 - RSS/RDF feed
 - SPARQL endpoint
- Treba obaviti sljedeće transformacije:
 - relacijske baze → RDF
 - XHTML → RDF
 - RSS/RDF = RDF ✓
 - SPARQL endpoint = RDF (preko upita) ✓

RSS/RDF

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://purl.org/rss/1.0/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
>
  <channel rdf:about="http://example.com/news.rss">
    <title>Example Channel</title>
    <link>http://example.com/</link>
    <description>My example channel</description>
    <items>
      <rdf:Seq>
        <rdf:li resource="http://example.com/2002/09/01/">
          <rdf:li resource="http://example.com/2002/09/02/">
            </rdf:li>
        </rdf:Seq>
      </items>
    </channel>
    <item rdf:about="http://example.com/2002/09/01/">
      <title>News for September the First</title>
      <link>http://example.com/2002/09/01/</link>
      <description>other things happened today</description>
      <dc:date>2002-09-01</dc:date>
    </item>
    <item rdf:about="http://example.com/2002/09/02/">
      <title>News for September the Second</title>
      <link>http://example.com/2002/09/02/</link>
      <dc:date>2002-09-02</dc:date>
    </item>
  </rdf:RDF>
```

GRDDL

- GRDDL = Gleaning Resource Descriptions from Dialects of Languages
- most između XML podataka i RDF podataka
- koristi oznake (markup) kojima se naznači da XML dokument sadrži podatke koji se mogu transformirati u RDF, kao i vezu na algoritam kojim se podaci transformiraju (obično XSLT)
- propisuje način izvlačenja RDF podataka iz XML dijalekata



Primjer:

```
<head profile="http://www.w3.org/2003/g/data-view http://dublincore.org/documents/dcq-html/ http://gmpg.org/xfn/11">
<link rel="transformation" href="grokXFN.xsl"/>
```

GRDDL algoritam

© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

GRDDL - XSLT

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <xsl:transform xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:mb="http://musicbrainz.org/"
  xmlns:mm="http://musicmoz.org/" version="1.0">
- <xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes" />
- <xsl:template match="/" />
- <rdf:RDF>
  <xsl:apply-templates select="/musicmoz/category" />
  <xsl:apply-templates select="/musicmoz/style" />
</rdf:RDF>
- <xsl:template match="musicmoz/category[(@type='band' or @type='artist') and resource/@name='musicbrainz']">
  <xsl:if test="not(contains(resource/@name='musicbrainz']/@link, 'showartist'))">
    <xsl:variable name="artist_uri" select="resource[@name='musicbrainz']/@link" />
    <!-- Property: mm:style -->
    <xsl:for-each select="style">
      <xsl:variable name="style_reformatted" select="concat('http://musicmoz.org/style/', translate(text(), 'Ã@Ã Ã²Ã Ã³Ã-;', '-
      eaeoui'))" />
      <rdf:Description rdf:about="{ $artist_uri }">
        <mm:hasStyle rdf:resource="{ $style_reformatted }" />
      </rdf:Description>
    </xsl:for-each>
    <!-- Property: mm:from -->
    <rdf:Description rdf:about="{ $artist_uri }">
      <mm:from>
        <xsl:value-of select="from" />
      </mm:from>
    </rdf:Description>
  </xsl:if>
</xsl:template>
- <xsl:template match="musicmoz/style">
  <xsl:variable name="style_reformatted" select="concat('http://musicmoz.org/style/', translate(name, 'Ã@Ã Ã²Ã Ã³Ã-;', '-eaeoui'))" />
  <mm:Style rdf:about="{ $style_reformatted }">
    <rdfs:label>
      <xsl:value-of select="name" />
    </rdfs:label>
  </mm:Style>
</xsl:template>
<xsl:template match="*" />
</xsl:transform>
```

© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Integracija podataka – primjer (2)

- Pretpostavimo da treba integrirati podatke iz sljedećih izvora:
 - relacijska baza bp1
 - relacijska baza bp2
 - Web 2.0. aplikacija sa sučeljem u XHTML-u
 - RSS/RDF feed
 - SPARQL endpoint
- Treba obaviti sljedeće transformacije:
 - relacijske baze → RDF
 - XHTML → RDF = GRDDL ✓
 - RSS/RDF → RDF ✓
 - SPARQL endpoint = RDF (preko upita) ✓

© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Relacijske baze i RDF (1)

- n-torke relacija su slične RDF trojkama
 - sama n-torka se može gledati kao **subjekt** (čvor)
 - atribut kao **predikat**
 - vrijednost atributa pojedine n-torke kao **objekt**
- ipak preslikavanje nije tako jednostavno
 - kako odrediti URI, odgovara li on primarnom ključu?
 - u bazi podataka iste trojke se mogu javiti više puta
 - prilagođavanje tipova podataka odabranoj ontologiji
 - šifrirani podaci (npr. spol 'M' i 'Ž')
 - ponekad je potrebno spajati više vrijednosti atributa (npr. ime i prezime čine jednu vrijednost u ontologiji)
 - NULL vrijednosti
 - ...

© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Relacijske baze i RDF (2)

GRAD

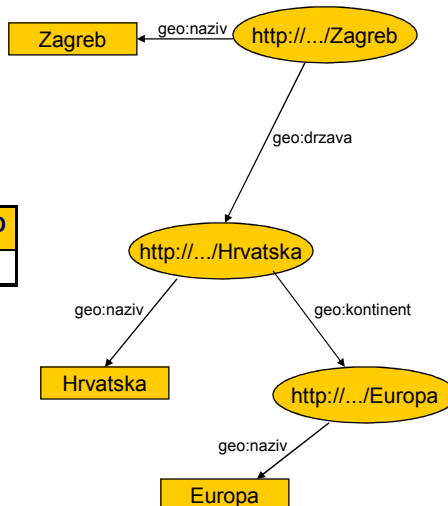
ID	naziv	oznDrzava
1	Zagreb	HR

DRŽAVA

oznDrzava	naziv	kontinentID
HR	Hrvatska	1

KONTINENT

ID	naziv
1	Europa



Pitanje: Jesu li URI-ji dovoljno dobro definirani?

© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Relacijske baze i RDF (3)

Postojeće tehnologije preslikavanja:

- D2RQ / D2R server
- Virtuoso
- DartGrid
- SPASQL

© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

Integracija podataka – primjer (3)

- Pretpostavimo da treba integrirati podatke iz sljedećih izvora:
 - relacijska baza bp1
 - relacijska baza bp2
 - Web 2.0 aplikacija sa sučeljem u XHTML-u
 - RSS/RDF feed
 - SPARQL endpoint
- Treba obaviti sljedeće transformacije:
 - relacijske baze → RDF = D2RQ ✓
 - XHTML → RDF = GRDDL ✓
 - RSS/RDF → RDF ✓
 - SPARQL endpoint = RDF (preko upita) ✓

Odabir ontologije

- za obavljanje preslikavanja treba odabrati ciljanu ontologiju
- budući da se u semantičkom web-u potiče dijeljenje ontologija, potrebno je pregledati dostupnost odgovarajuće ontologije (ili barem pojma)
 - npr. <http://swoogle.umbc.edu/>, www.sindice.com
- ukoliko primjenjiva ontologija nije dostupna, treba ju kreirati

D2RQ / D2R (1)

- definira se datoteka koja sadrži preslikavanja strukture relacijske baze u ontologiju
- preslikavanje se definira upotrebom posebnog jezika

preslikavanje relacija u razrede

```
map:Grad a d2rq:ClassMap;
    d2rq:uriPattern "grad/@@Grad.ID@";
    d2rq:class geo:Grad;
    d2rq:dataStorage map:Baza;
```

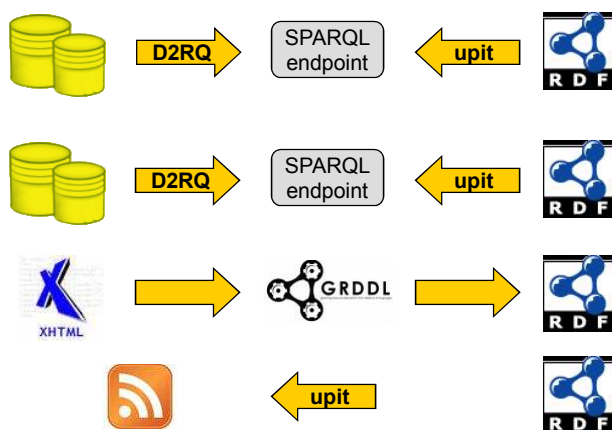
preslikavanje atributa u predikate

```
map:NazivGrada a d2rq:PropertyBridge;
    d2rq:belongsToClassMap map:Grad;
    d2rq:property geo:naziv;
    d2rq:column "Grad.Naziv";
```

D2RQ / D2R (2)

- preslikanim podacima se može pristupati preko SPARQL endpoint-a
- SPARQL upit prevodi se u SQL upit koji se izvodi nad relacijskom bazom te se dobiveni rezultati transformiraju u rezultate SPARQL upita
- sve promjene u bazi su odmah vidljive i na RDF modelu s obzirom da RDF model nije učitao u memoriju nego se upiti uvijek izvode nad relacijskim modelom

Integracija podataka – primjer (4)



© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

URI

- URI ima ključnu ulogu kod integracije
- služi za jedinstveno identificiranje pojmova
- budući da je dodjeljivanje URI-a uglavnom decentralizirano javlja se problem da isti pojam ima dva ili više URI-a; problem se rješava obično uvođenjem tzv. "mapping ontologija" u kojima se ti pojmovi izjednačavaju

© FER - Zagreb

Napredni modeli i baze podataka 2008/2009

URI, URL i URN

- URI – Uniform Resource Identifier
 - niz sačinjen od oznaka koji identificira apstraktan ili fizički izvor (resource)
- URL – Uniform Resource Locator
 - odnosi se na podskup URI-ja koji identificiraju izvore preko mehanizma pristupa tim lokacijama (mrežna lokacija)
 - u slučaju promjene lokacije, mijenja se i URL
- URN – Uniform Resource Name
 - odnosi se na podskup URI-ja za koje je potrebno da su globalno jedinstveni i da njihov identifikator ostaje čak i kada oni prestanu postojati ili postanu nedostupni
- u službenoj W3C dokumentaciji se više ne koriste pojmovi URL i URN, samo URI

Integracija podataka – primjer (5)

- udruživanje dobivenih modela
- treba izjednačiti URI-e istih pojmova
 - Primjer:


```
http://dbpedia.org/resource/Berlin owl:sameAs
http://sws.geonames.org/2950159
```
- povezati značenja različitih ontologija
 - Primjer:


```
<owl:Class rdf:about="http://purl.org/net/biblio#Person">
  <owl:equivalentClass
    rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
  <owl:equivalentClass
    rdf:resource="http://purl.org/vocab/frbr/core#Person"/>
  <owl:equivalentClass
    rdf:resource="http://xmlns.com/wordnet/1.6/Person"/>
</owl:Class>
```

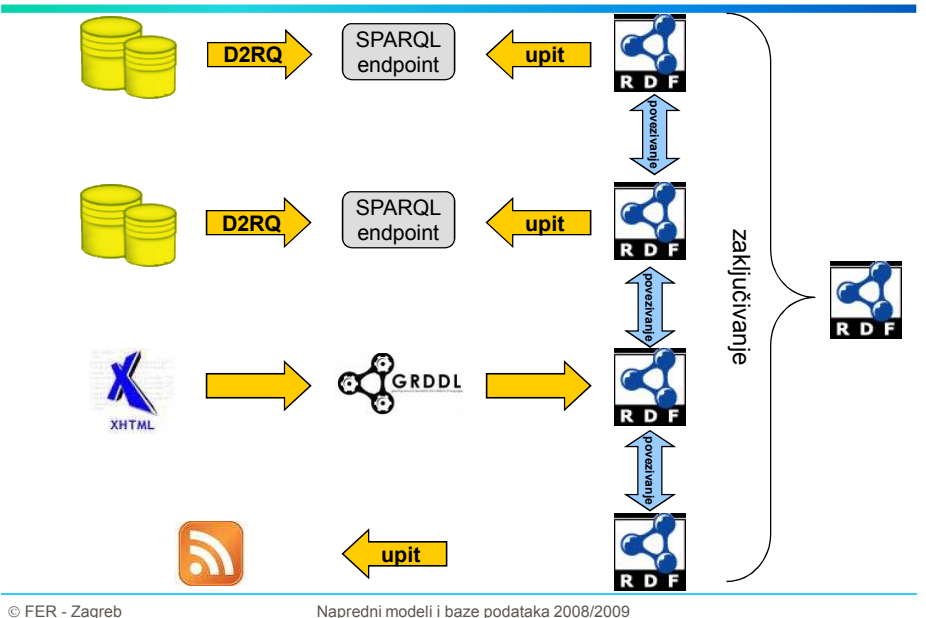
Integracija podataka – primjer (6)

- podatke dobivene u RDF formatu (RDF trojke) treba negdje pohraniti
- RDF spremišta nude i metode za dohvaćanje podataka
- “In memory”
 - Jena, Sesame
- “native”
 - Virtuoso, Sesame Native, Oracle 11g, AllegroGraph
- temeljen na relacijskim bazama podataka
 - Jena + (MySQL, PostgreSQL, Oracle, DB2)
- najbolje performanse pokazuju native RDF spremišta

Zaključivanje

- Mehanizam za zaključivanje
 - zaključivanje nad pravilima definiranim u OWL-u
 - rezultat zaključivanja su dodatne trojke
 - neki od mehanizama za zaključivanje: Jena Reasoner, Pellet, Racer,...

Integracija podataka – primjer (7)



Alati

- Java alati
 - Jena – Java framework za semantički web
 - Joseki – SPARQL server za Jenu
 - D2RQ / D2R
 - GRDDL transformator
- Slični alati postoje i za druge programske jezike