

Podstawy Inżynierii Wiedzy

Sprawozdanie 3

Grupa: wtorek 16:00
Bartłomiej Hyży
Witold Baran

Resource Description Framework (RDF) in use

1 Semantic vocabularies: Dublin Core

2 RDFSchema

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:lib="http://example.org/mylibrary#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdfs:Class rdf:ID="MultimediaItem" />
  <rdfs:Class rdf:ID="MusicCD">
    <rdfs:subClassOf rdf:resource="#MultimediaItem"/>
    <rdfs:label>Music Compact Discs class</rdfs:label>
    <rdfs:comment>Class of all the CDs in my library.</rdfs:comment>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Book">
    <rdfs:subClassOf rdf:resource="#MultimediaItem"/>
    <rdfs:label>Books class</rdfs:label>
    <rdfs:comment>Class of all the books in my library.</rdfs:comment>
  </rdfs:Class>

  <rdf:Description rdf:about="http://example.org/mylibrary/books/#book-porwaniegabki">
    <dc:creator>Stanisław Pagaczewski</dc:creator>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Porwanie Baltazara
Gąbki</lib:dc>
    <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1965</dc:date>
    <rdf:type rdf:resource="http://example.org/mylibrary#Book"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://example.org/mylibrary/cds#cd-darkside">
    <dc:creator>Pink Floyd</dc:creator>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The Dark Side of the
Moon</dc:title>
    <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1973</dc:date>
    <rdf:type rdf:resource="http://example.org/mylibrary#MusicCD"/>
  </rdf:Description>
</rdf:RDF>
```

```

</rdf:Description>

<rdf:Description rdf:about="http://example.org/mylibrary#favorites">
  <lib:MyFavouriteBooks >
    <rdf:Bag>
      <rdf:li rdf:resource="http://example.org/mylibrary/books/TytusRomekiAtomek"/>
      <rdf:li rdf:resource="http://example.org/mylibrary/books/CzerwonyKaptorek"/>
      <rdf:li rdf:resource="http://example.org/mylibrary/books/JakWojtekZostalStrazakiem"/>
      <rdf:li rdf:resource="http://example.org/mylibrary/books/JankoMuzykant"/>
    </rdf:Bag>
  </lib:MyFavouriteBooks >
</rdf:Description>

<rdf:Description rdf:about="http://example.org/mylibrary/cds#cd-harpattack">
  <lib:authors rdf:parseType="Collection">
    <rdf:Description rdf:about="http://example.org/mylibrary/CareyBell"/>
    <rdf:Description rdf:about="http://example.org/mylibrary/BillyBranch"/>
    <rdf:Description rdf:about="http://example.org/mylibrary/JamesCotton"/>
    <rdf:Description rdf:about="http://example.org/mylibrary/JuniorWells"/>
  </lib:authors>
  <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Harp Attack</dc:title>
  <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1990</dc:date>
  <rdf:type rdf:resource="http://example.org/mylibrary#MusicCD"/>
</rdf:Description>

</rdf:RDF>

```

3 SPARQL - demo

3.1. What semantic vocabularies are used in the queries? What are they for?

FOAF (Friend of a Friend) - opisywanie osób, powiązań między nimi, czym się zajmują.

DC (Dublin Core) - słownik służący do opisu dokumentów.

SIOC (Semantically-Interlinked Online Communities Project) - łączenie ze sobą platform dyskusyjnych takich jak blogi, fora czy listy mailingowe.

REV (Review Vocabulary) - opis recenzji oraz rankingów.

PRJ (Project Vocabulary) - określanie projektów, niezależnie od jego dziedziny.

RSS (Really Simple Syndication) - powiadamianie o nowościach na stronach i wszelkiego rodzaju serwisach internetowych

3.3. What do SELECT queries do?

Zapytania SELECT pozwalają na otrzymanie wartości atrybutów (zmiennych) spełniających zadane kryterium. Atrybuty te dotyczą zasobów różnego rodzaju opisanych za pomocą RDF. Wyniki mogą zostać zwrócone w różnych formatach (JSON, XML, CSV, itp.). Dyrektywa SELECT jest odpowiednikiem analogicznej instrukcji z języka SQL i działa w podobny sposób, dokonując projekcji zbioru rozwiązań (np. otrzymanych z WHERE) na zbiór wynikowy.

3.4. What do CONSTRUCT queries do?

Zapytania CONSTRUCT pozwalają na otrzymanie grafu RDF. Najpierw na podstawie kryterium zapytania (WHERE) konstruowane są trójki (tripes) RDF. Następnie na ich podstawie generowany jest graf RDF w oparciu o dostarczony wzorzec oraz uzyskane wyniki zapytania.

4 SPARQL queries - basics

Uwaga: we wszystkich zapytaniach poniżej używamy dla adresu e-mail atrybutu foaf:mbox_sha1sum zamiast

foaf:mbox, gdyż nie zahashowane adresy nie zostały określone dla naszego grafu FOAF.

Execute queries on your foaf file to retrieve :

- **friends who have name and e-mail defined**

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?name ?email
FROM <http://student.agh.edu.pl/~emp/foaf.rdf>
WHERE {
    ?x rdf:type foaf:Person;
        foaf:knows ?friend.
    ?friend foaf:name ?name;
            foaf:mbox_shalsum ?email
}
```

- **friends who have name and e-mail defined and optional homepage**

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?name ?email ?homepage
FROM <http://student.agh.edu.pl/~emp/foaf.rdf>
WHERE {
    ?x rdf:type foaf:Person;
        foaf:knows ?friend.
    ?friend foaf:name ?name;
            foaf:mbox_shalsum ?email.
    OPTIONAL { ?friend foaf:homepage ?homepage }
}
```

- **friends who have name and e-mail defined and optional homepage, sorted by name descending**

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?name ?email ?homepage
FROM <http://student.agh.edu.pl/~emp/foaf.rdf>
WHERE {
    ?x rdf:type foaf:Person;
        foaf:knows ?friend.
    ?friend foaf:name ?name;
            foaf:mbox_shalsum ?email.
    OPTIONAL { ?friend foaf:homepage ?homepage }
}
ORDER BY DESC(?name)
```

5 SPARQL queries - options

- **people whose name starts with 'K'**

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?person
```

```

FROM <http://student.agh.edu.pl/~emp/foaf.rdf>
WHERE {
    ?person rdf:type foaf:Person;
           foaf:name ?name.
    FILTER regex(?name, '^K').
}

```

- **people who are older than 18 years old**

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?person
FROM <http://student.agh.edu.pl/~emp/foaf.rdf>
WHERE {
    ?person rdf:type foaf:Person;
           foaf:age ?age.
    FILTER (xsd:decimal(?age) > 18).
}

```

- **people whose name starts with 'K' or are older than 18 years old, make search caseinsensitive**

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?person
FROM <http://student.agh.edu.pl/~emp/foaf.rdf>
WHERE {
    ?person rdf:type foaf:Person;
           foaf:age ?age;
           foaf:name ?name.
    FILTER (xsd:decimal(?age) > 18 || regex(?name, '^K', 'i')).
}

```

- **people having e-mails on student.agh.edu.pl server**

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT ?person
FROM <http://student.agh.edu.pl/~emp/foaf.rdf>
WHERE {
    ?person rdf:type foaf:Person;
           foaf:mbox ?mail.
    FILTER regex(?mail, '@student.agh.edu.pl', 'i').
}

```

- **name of people, who have homepage or e-mail on student.agh.edu.pl server**

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT DISTINCT ?name
FROM <http://student.agh.edu.pl/~emp/foaf.rdf>
WHERE {
    {
        ?person rdf:type foaf:Person;
               foaf:name ?name;
               foaf:mbox ?mail.
    }
}

```

```
    FILTER regex(?mail, '@student.agh.edu.pl', 'i').
  }
UNION
{
  ?person rdf:type foaf:Person;
    foaf:name ?name;
    foaf:homepage ?homepage.
  FILTER regex(xsd:string(?homepage), 'student.agh.edu.pl/~', 'i').
}
}
```

6 Open Data Sets

Podstawową wadą zapytań do baz takich jak DBPedia czy MusicBrainz jest niepełne pokrycie wiedzy z Wikipedii i innych źródeł odpowiednimi ontologiami, co skutkuje zwykle otrzymaniem jedynie częściowej odpowiedzi. Ponadto spora część faktów opisana jest za pomocą niespójnych, niejednoznacznych ontologii, dochodzi więc do sytuacji, gdy te same “rzeczy” opisywane są przez kilka różnych atrybutów, co utrudnia wyszukiwanie interesujących nas informacji. Istotną wadą jest również konieczność znajomości ontologii w dziedzinie, w której wykonujemy zapytania. Wreszcie za ograniczenia z pewnością można uznać stosunkowo niską wydajność bardziej skomplikowanych zapytań oraz konieczność stosowania języka SPARQL do ich konstrukcji, co stanowi istotną przeszkodę dla typowego użytkownika Internetu.