

Jakub Jaśkowiec  
Michał Pieróg

Gr. środa 17:00

# Podstawy Inżynierii Wiedzy

## Semantic Web – sprawozdanie 3

### 1. Semantic vocabularies: Dublin Core

### 2. 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://abcd.com/mylib#" xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdf:Description rdf:about="http://abcd.com/mylib/books/#book-elementarz">
    <dc:creator>Anna Czohra</dc:creator>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Mój pierwszy
elementarz</dc:title>
    <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1997</dc:date>
    <rdf:type rdf:resource="http://abcd.com/mylib#Book"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://abcd.com/mylib/cds#cd-rosenrot">
    <dc:creator>Rammstein</dc:creator>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Rosenrot</dc:title>
    <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1995</dc:date>
    <rdf:type rdf:resource="http://abcd.com/mylib#MusicCD"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://abcd.com/mylib/books#favorites">
    <lib:MyFavouriteBooks >
      <rdf:Bag>
        <rdf:li rdf:resource="http://abcd.com/mylib/books/TheGirlWithTheDragonTattoo"/>
        <rdf:li rdf:resource="http://abcd.com/mylib/books/TheGirlWhoPlayedWithFire"/>
        <rdf:li rdf:resource="http://abcd.com/mylib/books/TheGirlWhoKickedTheHornetsNest"/>
      </rdf:Bag>
    </lib:MyFavouriteBooks >
  </rdf:Description>

  <rdf:Description rdf:about="http://abcd.com/mylib/dvds#dvd-shutter-island">
    <dc:creator rdf:parseType="Collection">
      <rdf:Description rdf:about="http://abcd.com/mylib/dvds/LeonardoDiCaprio"/>
      <rdf:Description rdf:about="http://abcd.com/mylib/dvds/BenKingsley"/>
      <rdf:Description rdf:about="http://abcd.com/mylib/dvds/MarkRuffalo"/>
    </dc:creator>
  </rdf:Description>
```

```

    <rdf:Description rdf:about="http://abcd.com/mylib/dvds/MichelleWilliams"/>
    </dc:creator>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Shutter Island</dc:title>
    <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2010</dc:date>
    <rdf:type rdf:resource="http://abcd.com/mylib#DVD"/>
  </rdf:Description>

  <rdfs:Class rdf:ID="Multimedialtem" />
  <rdfs:Class rdf:ID="MusicCD">
    <rdfs:subClassOf rdf:resource="#Multimedialtem"/>
    <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="#Multimedialtem"/>
    <rdfs:label>Books class</rdfs:label>
    <rdfs:comment>Class of all books in my library.</rdfs:comment>
  </rdfs:Class>

  <rdfs:Class rdf:ID="DVD">
    <rdfs:subClassOf rdf:resource="#Multimedialtem"/>
    <rdfs:label>DVD class</rdfs:label>
    <rdfs:comment>Class of all the DVDs in my library.</rdfs:comment>
  </rdfs:Class>

</rdf:RDF>

```

### 3. SPARQL - demo

**foaf** (Friend Of A Friend) - powiazania pomiedzy osobami  
**dc** (Dublin Core) - opis multimediiów i książek  
**sioc** (Semantically-Interlinked Online Communities Project) - opis społeczności internetowych  
**rss** (Really Simple Syndication) - publikacja często zmieniających się treści  
**rev** (Review Vocabulary) - słownictwo do wyrażania recenzji oraz ocen  
**owl** (Web Ontology Language) - reprezentacja i przetwarzanie danych w sieci www  
**rdfs** (RDF Schema) - rozszerzenie RDF  
**prj** (Project VOcabulary) - opis projektów niezależnie od ich dziedziny

Kwerenda SELECT - pozwala na otrzymywanie danych z kolekcji, rezultaty zwracane są w formie tabeli; jest odpowiednikiem tej samej konstrukcji w SQL

Klauzula CONSTRUCT - na wejściu przyjmuje klauzulę WHERE, która opisuje wzorzec, w tym definicje zmiennych; zwraca pojedynczy graf RDF (zbiór trójek), który jest określony za pomocą szablonu grafu.

## 4. SPARQL queries - basics

### a) 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 DISTINCT ?name ?email
WHERE {
    ?person rdf:type foaf:Person;
            foaf:knows ?friend.
    ?friend foaf:name ?name;
            foaf:mbox ?email
}
```

### b) 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 DISTINCT ?name ?email ?homepage
WHERE {
    ?person rdf:type foaf:Person;
            foaf:knows ?friend.
    ?friend foaf:name ?name;
            foaf:mbox ?email.
    OPTIONAL { ?friend foaf:homepage ?homepage}
}
```

### c) 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 DISTINCT ?name ?email ?homepage
FROM <http://www.dajobe.org/foaf.rdf>
WHERE {
    ?person rdf:type foaf:Person;
            foaf:knows ?friend.
    ?friend foaf:name ?name;
            foaf:mbox ?email.
    OPTIONAL { ?friend foaf:homepage ?homepage}
}
ORDER BY DESC(?name)
```

## 5. SPARQL queries - options

### a) 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 DISTINCT ?person
WHERE {
    ?person rdf:type foaf:Person;
           foaf:name ?name.
FILTER regex(?name,'^K')
}
ORDER BY DESC(?name)
```

### b) 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 DISTINCT ?person
WHERE {
    ?person rdf:type foaf:Person;
           foaf:age ?age.
    FILTER(xsd:decimal(?age)>18).
}
```

### c) 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#>

```
SELECT DISTINCT ?person
WHERE {
    ?person rdf:type foaf:Person;
           foaf:name ?name;
           foaf:age ?age.
    FILTER(xsd:decimal(?age)>18)
    FILTER(regex(?name,'^K','i'))
}
```

**d) 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#>

```
SELECT DISTINCT ?person ?email
WHERE {
    ?person rdf:type foaf:Person;
           foaf:mbox ?email.
    FILTER(regex(?email,'@student.agh.edu.pl$', 'i'))
}
```

**e) 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
WHERE { {
    ?person rdf:type foaf:Person;
           foaf:mbox ?email;
           foaf:name ?name.
    FILTER(regex(?email,'@student.agh.edu.pl$', 'i'))
}
UNION {
    ?person rdf:type foaf:Person;
           foaf:homepage ?www;
           foaf:name ?name.
    FILTER(regex(xsd:string(?www), '^http://student.agh.edu.pl/~', 'i'))
}
}
```

## 6) Open Data Sets

Dla typowego użytkownika Internetu zasoby te są niedostępne, gdyż do ich wyszukiwania potrzebna jest znajomość SPARQL, znacznie wygodniejsze jest użycie typowej przeglądarki. Niektóre zapytania są przetwarzane bardzo długo, zależnie od rozmiaru dokumentu i poziomu skomplikowania zapytania.