

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:lb="http://www.example.com/mylibrary"
  xmlns:dc="http://purl.org/dc/elements/1.1/">

  <rdfs:Class rdf:ID="MultimediaItem" />
  <rdfs:Class rdf:ID="MovieDVD">
    <rdfs:subClassOf rdf:resource="#MultimediaItem"/>
    <rdfs:label>Movie DV Disc class</rdfs:label>
    <rdfs:comment>Class of all the DVDs in my library.</rdfs:comment>
  </rdfs:Class>
  <rdfs:Class rdf:ID="Book">
    <rdfs:subClassOf rdf:resource="#MultimediaItem"/>
    <rdfs:label>Book class</rdfs:label>
    <rdfs:comment>Class of all the books in my library.</rdfs:comment>
  </rdfs:Class>

  <rdf:Description rdf:about="http://example.com/mylibrary#CssGuide">
    <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Eric
Meyer</dc:author>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">CSS: The
define guide</dc:title>
<dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2008<dc:date>
  <dc:language>en</dc:language>
  <dc:type rdf:resource="http://example.com/mylibrary#Book"/>
</rdf:Description>
  <rdf:Description rdf:about="http://example.com/mylibrary#Webdesing">
    <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Dan
Cederholm</dc:author>
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Buletproof
web desing</dc:title>
<lb:year rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2008<lb:year>
  <dc:language>en</dc:language>
  <dc:type rdf:resource="http://example.com/mylibrary#Book"/>
</rdf:Description>
  <rdf:Description rdf:about="http://example.com/mylibrary#Rails">
    <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Sam
Ruby</dc:author>
    <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Dave
Thomas</dc:author>
    <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">David
Heinemeier Hansson</dc:author>
```

```

    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Agile
Web Development with Rails</dc:title>
<dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2011<dc:date>
  <dc:language>en</dc:language>
  <dc:type rdf:resource="http://example.com/mylibrary#Book"/>
</rdf:Description>
<rdf:Description rdf:about="http://example.com/mylibrary#HeartOfDarkness">
  <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Joseph
Conrad</dc:author>
  <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Heart of
Darkness</dc:title>
<dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2010<dc:date>
  <dc:language>en</dc:language>
  <dc:type rdf:resource="http://example.com/mylibrary#Book"/>
</rdf:Description>
<rdf:Description rdf:about="http://example.com/mylibrary#Gladiator">
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Gladiator</
dc:title>
  <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Ridley
Scott</dc:author>
  <lb:actors>
    <rdf:Bag>
      <rdf:li rdf:resource="http://example.com/actor/Russel_Crow" />
      <rdf:li rdf:resource="http://example.com/actor/Joaquin_Phoenix" />
      <rdf:li rdf:resource="http://example.com/actor/Connie_Nielsen" />
    </rdf:Bag>
  </lb:actors>
  <dc:date>2000</dc:date>
  <dc:language>en</dc:language>
  <dc:type rdf:resource="http://example.com/mylibrary#MovieDVD"/>
</rdf:Description>

<rdf:Description rdf:about="http://example.com/mylibrary#Gump">
  <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Forest
Gump</dc:title>
  <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Robert
Zemeckis</dc:author>
  <lb:actors>
    <rdf:Bag>
      <rdf:li rdf:resource="http://example.com/actor/Robin_Wright" />
      <rdf:li rdf:resource="http://example.com/actor/Tom_Hanks" />
    </rdf:Bag>
  </lb:actors>
  <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">1994</
dc:date>
  <dc:language>en</dc:language>
  <dc:type rdf:resource="http://example.com/mylibrary#Book"/>
</rdf:Description>
</rdf:RDF>

```

3 SPARQL - demo

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

- Friend of a Friend - to track relationships among people

- Dublin Core - describe web and physical resources
- Rev - describe reviews
- SIOC - describe blogs, mailing lists etc.

What do select queries do?

They get the specified values and show them in a tabular representation.

What do construct queries do?

They get the values and present them as RDF

4 SPARQL queries - basics

friends who have name and e-mail defined

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

```
SELECT DISTINCT ?name ?email
WHERE {
  ?x rdf:type foaf:Person.
  ?x foaf:knows ?y.
  ?y foaf:name ?name.
  ?y foaf:name ?email.
}
```

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

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

```
SELECT DISTINCT ?name ?email ?homepage
WHERE {
  ?x rdf:type foaf:Person.
  ?x foaf:knows ?y.
  ?y foaf:name ?name.
  ?y foaf:name ?email.
  OPTIONAL {
    ?y foaf:homepage ?homepage
  }
}
```

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

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

```
SELECT DISTINCT ?name ?email ?homepage
WHERE {
  ?x rdf:type foaf:Person.
  ?x foaf:knows ?y.
  ?y foaf:name ?name.
  ?y foaf:name ?email.
  OPTIONAL {
    ?y foaf:homepage ?homepage
  }
}
ORDER BY DESC(?name)
```

5 SPARQL queries - options

people whose name starts with 'K'

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

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT DISTINCT ?name
WHERE {
    ?x rdf:type foaf:Person.
    ?x foaf:name ?name.
FILTER regex(?name, '^K')
}
```

people who are older than 18 years old

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

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT DISTINCT ?name
WHERE {
    ?x rdf:type foaf:Person.
    ?x foaf:age ?age.
FILTER(?age > 18)
}
```

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

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

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

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

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

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

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

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

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

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

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

```
SELECT DISTINCT ?name
WHERE {
```

```
?x rdf:type foaf:Person.  
?x foaf:name ?name.  
?x foaf:homepage ?hp. FILTER regex(?hp, 'student.agh.edu.pl', 'i').  
OPTIONAL {  
  ?x foaf:email ?email. FILTER regex(?email, '@student.agh.edu.pl$').  
}  
}
```

6 Open Data Sets

What are the main limitations of using (querying for information) the RDF datasets such as DBpedia or MusicBrainz?

Main limitation is long response time. Its id caused by complexity of RDF resources and queries. Another limitation is that access to knowledge have only this people who knows SPARQL language.