

Resource Description Framework (RDF) in use

Leszek Sosnowski
Aleksander Steliga

Public Core & RDF Schema

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

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

  <rdf:Description rdf:about="http://aleksanderLeszek.com/mylibrary#book-catch22">
    <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Joseph Heller</dc:author>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Catch 22</ dc:title>
    <dc:publisher rdf:datatype="http://www.w3.org/2001/XMLSchema#string">PRIMA</ dc:publisher>
    <rdf:type rdf:resource="http://aleksanderLeszek.com/mylibrary#" />
  </rdf:Description>

  <rdf:Description rdf:about="http://aleksanderLeszek.com/mylibrary#book-gameofthrones">
    <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">George R.R. Martin</ dc:author>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">A Game of Thrones</ dc:title>
    <dc:publisher rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Nova</ dc:publisher>
    <rdf:type rdf:resource="http://aleksanderLeszek.com/mylibrary#" />
  </rdf:Description>

  <rdf:Description rdf:about="http://aleksanderLeszek.com/mylibrary#book-dancewithdragons">
    <dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">George R.R. Martin</ dc:author>
    <dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">A Dance With Dragons</ dc:title>
```

```
<dc:publisher rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Nova</ dc:publisher>
<rdf:type rdf:resource=" http://aleksanderLeszek.com/mylibrary#" />
</rdf:Description>
```

```
<rdf:Description rdf:about="http://aleksanderLeszek.com/mylibrary#book-catcher">
<dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Jerome David Salinger</ dc:author>
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">The Catcher in the Rye</ dc:title>
<dc:publisher rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Iskry</ dc:publisher>
<rdf:type rdf:resource=" http://aleksanderLeszek.com/mylibrary#" />
</rdf:Description>
```

```
<rdf:Description rdf:about="http://aleksanderLeszek.com/mylibrary#book-11minutes">
<dc:author rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Paulo Coelho</ dc:author>
<dc:title rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Eleven Minutes</ dc:title>
<dc:publisher rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Drzewo Babel</ dc:publisher>
<rdf:type rdf:resource=" http://aleksanderLeszek.com/mylibrary#" />
</rdf:Description>
```

```
<rdf:Description rdf:about="http://aleksanderLeszek.com/mylibrary/MyFavouriteBooks">
  <MyFavouriteBooks>
    <rdf:Bag>
      <rdf:li rdf:resource="http://aleksanderLeszek.com/mylibrary#book-catch22"/>
      <rdf:li rdf:resource="http://aleksanderLeszek.com/mylibrary#book-catcher"/>
    </rdf:Bag>
  </ MyFavouriteBooks>
</rdf:Description>
</rdf:RDF>
```

SPARQL – demo

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

FOAF – opisywanie osób i znajomości między nimi

DC – opisywanie zasobów elektronicznych

PJV – opisywanie projektów

RSS – opisywanie i powiadamianie o zmianach na stronach internetowych

SIOC – łączenie ze sobą platform dyskusyjnych

REV – opisywanie recenzji, rankingów

What do SELECT queries do?

Używane są do wyciągnięcia danych spełniających podane kryterium. Wyniki są zwracane w formie tabeli.

What do CONSTRUCT queries do?

Służą do pobierania informacji i tworzeniu trójek (triples) RDF i transformowaniu ich na graf RDF.

SPARQL queries – basics

friends who have name and e-mail defined

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT $name $mbox $homepage FROM <http://student.agh.edu.pl/lich/test.rdf>
WHERE
{ $x a foaf:Person .
  $x foaf:name $name .
  $x foaf:mbox $mbox }
```

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

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT $name $mbox $homepage FROM <http://student.agh.edu.pl/lich/test.rdf>
WHERE
{ $x a foaf:Person .
  $x foaf:name $name .
  $x foaf:mbox $mbox .
  optional { $x 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/>
SELECT $name $mbox $homepage FROM <http://student.agh.edu.pl/lich/test.rdf>
WHERE
{ $x a foaf:Person .
  $x foaf:name $name .
  $x foaf:mbox $mbox .
  optional { $x foaf:homepage $homepage } }
ORDER BY DESC ($desc)
```

people whose name starts with 'K':

```
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:

```
SELECT DISTINCT $name, $age
WHERE {
  $x rdf:type foaf:Person ;
  $x foaf:name $name ;
  $x foaf:age $age .
```

```
FILTER( $age > 18 )}
```

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

caseinsensitive:

```
SELECT DISTINCT $name, $age
WHERE {
  $x rdf:type foaf:Person ;
  $x foaf:name $name.
  $x foaf:age $age .
  FILTER ($age > 18) || regex($name, „^K.*”, „i”)}
```

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

```
SELECT DISTINCT $name, $mail
WHERE {
  $x rdf:type foaf:Person ;
  $x foaf:name $name ;
  $x foaf:mbox $mail
  FILTER regex($mail, „.*student.agh.edu.pl$”)}
```

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

```
SELECT DISTINCT $name, $mail, $page
WHERE {
  $x rdf:type foaf:Person ;
  $x foaf:name $name ;
  $x foaf:mbox $mail ;
  $x foaf:homepage $page ;
  FILTER regex($mail, „.*student.agh.edu.pl$”) || regex($page, „^http://
student.agh.edu.pl.*”)}
```

Open Data Sets

Poważnym problemem jest niska wydajność wyszukiwarki – czas oczekiwania jest tym większy im większa jest baza i stopień skomplikowania zapytania. Ponadto część artykułów jest napisana za pomocą niespójnych i niepokrywających się ontologii. Kolejnym problemem jest konieczność tworzenia zapytań w języku SPARQL, który może być problematyczny dla przeciętnego użytkownika internetu.