



I. Używane do opisu projektów oprogramowania, w szczególności open-source.

II. <https://github.com/edumbill/doap/wiki/Sites>

### Ad. 3.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:admin="http://webns.net/mvcb/">
<foaf:PersonalProfileDocument rdf:about="">
  <foaf:maker rdf:resource="#me"/>
  <foaf:primaryTopic rdf:resource="#me"/>
  <admin:generatorAgent rdf:resource="http://www.ldodds.com/foaf/foaf-a-matic"/>
  <admin:errorReportsTo rdf:resource="mailto:leigh@ldodds.com"/>
</foaf:PersonalProfileDocument>
<foaf:Person rdf:ID="me">
<foaf:name>Adrian Sudol</foaf:name>
<foaf:title>Mr</foaf:title>
<foaf:givenname>Adrian</foaf:givenname>
<foaf:family_name>Sudol</foaf:family_name>
<foaf:nick>Conker</foaf:nick>
<foaf:mbox_sha1sum>a18b380dac3593b94694e5a0404c5f83bea822ff</foaf:mbox_sha1sum>
<foaf:depiction rdf:resource="http://a4.sphotos.ak.fbcdn.net/hphotos-ak-
ash4/s720x720/317184_298421196849856_100000459335346_1138629_2092456009_n.jpg"/>
<foaf:workplaceHomepage rdf:resource="http://www.otago.pl/index.html"/>
<foaf:workInfoHomepage rdf:resource="http://www.otago.pl/praca.html"/>
<foaf:schoolHomepage rdf:resource="http://www.agh.edu.pl/en"/>
<foaf:knows>
<foaf:Person>
<foaf:name>Konrad</foaf:name>
<foaf:mbox_sha1sum>a8b7f3e489df66cd4804409aefb6b883ba66a84a</foaf:mbox_sha1sum></foaf:Person></fo
af:knows>
<foaf:knows>
<foaf:Person>
<foaf:name>Piotr</foaf:name>
<foaf:mbox_sha1sum>99e0df2596e4a3f4f2e2a201068615643b66d025</foaf:mbox_sha1sum></foaf:Person></fo
af:knows>
<foaf:knows>
<foaf:Person>
<foaf:name>Mateusz</foaf:name>
<foaf:mbox_sha1sum>7a31bf289076fc901457579f87ab996c6d213491</foaf:mbox_sha1sum></foaf:Person></foa
f:knows></foaf:Person>
</rdf:RDF>
```

### Ad. 4.

Starsze ontologie są stosunkowo proste i nierozbudowane.

### Ad. 5.

Nie da się.

## **Ad. 6.**

Nie da się.

## **Ad. 7.**

DATASETS:

Last.FM Artists (DBTune) :

Reprezentacja RDF artystów i podobnych artystów oparta na API Last.FM. Można używać nazwy artysty, identyfikatora musicbrainz, albo URI opartego na musicbrainz.

DBTropes :

Reprezentacja stron wiki TvTropes. Mogą być typu Item (gra, film), Feature (trope - motyw), Feature Cat (kategoria motywu), Ignore (selfexplanatory).

NASA (Data Incubator) :

Reprezentacja różnych zbiorów danych (datasets) na RDF. Dane o misjach, statkach, osobach.

New York Times :

Reprezentacja wiadomości. Cztery typy: ludzie, organizacje, miejsca, opisy.

Goodwin family :

Informacje o rodzenie i poszczególnych członkach. Dwa typy: rodzina, osoba.

## **Ad. 8.**

1. Wydaje mi się, że BibTEX -> RDF, JPEG -> RDF i Java -> RDF. Pierwsze dwa by ułatwić wyszukiwanie wartościowych informacji w sieci, trzecie ponieważ może zapewnić ciekawe ułatwienie browsowaniu kodu.

Dodałbym RDFizer do MP3, wyciągający informacje z tagów ID3v2

2. Razorbase, Fenfire i Objectviewer zdają się nie funkcjonować, a przynajmniej od dawna nie są rozwijane.

3. Dość łatwe, wystarczy wiedzieć czego tak naprawdę szukamy. Działają na analogicznej zasadzie do zwykłych wyszukiwarek.

4. Najciekawszym zdaje się być LinkedGeoData Browser, zbierający od użytkowników informacje o miejscach i łączący w sobie funkcjonalności mapy i encyklopedii.

## **Ad. CQ**

1. Annotacje Semantic Web można dodać do strony w postaci danych RDF (można je wygenerować lub napisać ręcznie)

2. Semantic Web Stack reprezentuje wartswową strukturę Semantic Web. Ukazuje hierarchię języków, gdzie każda warstwa używa możliwości warstwy niższej.

3. RDF tworzony jest w postaci trójek (triples), gdzie każda trójka zawiera węzeł podmiotu, predykat i węzeł dopełnienia. Predykat stanowi krawędź grafu stworzonego z węzłów. Taka składnia zapewne przejrzystość co jest ważne gry kod osiąga duże rozmiary, jak to często wygląda w przypadku RDF.

4. Są używane jako słownik służący do opisu zasobów danego typu.

5. Używanie URI jako nazw dla rzeczy,

Używanie HTTP URI aby ludzie mogli wyszukiwać te rzeczy,

Po wyszukaniu URI, zapewnienie wartościowych informacji za pomocą standardów (takich jak RDF),

Załączanie linków do innych URI, by umożliwić odnajdywanie innych rzeczy.

## XML

### Ad. 1.

- `<a><b/><a>`

WRONG! no closing tag for the root element `<a>`

- `<a><b>foo</b></a><a>bar</a>`

WRONG! no root element

- `<a><b>foo</b><b>bar</a>`

WRONG! no closing tag for the 2nd `<b>` element

- `<a><b><c>foo</b>bar</c></a>`

WRONG! bad nesting of `<b>` and `<c>`

- `<a/><b>foo</b><b>bar<b>`

WRONG! why close the root element before one puts the content inside it?

- `<a><b><c>foo</c>baz<c>bar</c></b></a>`

```
a
|---b
|---|---c
|---|---|---Text: foo
|---|---Text: baz
|---|---c
|---|---|---Text:bar
```

- `<a x="1" X="2"><b>foo</b></a>`

```
a [x=1, X=2]
|---b
|---|---Text: foo
```

- `<a x="1"><b x="2">foo</b></a>`

```
a [x=1]
|---b [x=2]
|---|---Text: foo
```

- `<a x="1" y="2" x="3"><b>foo</b></a>`

WRONG! argument 'x' redefined

- <a><b x="3"><c y="1">foo</c><c>bar</c></b></a>

a

```
|---b [x=3]
|---|---c [y=1]
|---|---|---Text: foo
|---|---c
|---|---|---Text: bar
```

## Ad. 2.

```
<book>
  <author>Donald E. Knuth</author>
  <title>Sztuka programowania</title>
  <subtitle>Tom III: Sortowanie i wyszukiwanie</subtitle>
  <date>2002</date>
  <publisher>Wydawnictwa Naukowo-Techniczne, Warszawa</publisher>
  <ISBN>83-204-2554-9</ISBN>
</book>
```

## Ad. 3.

```
<!DOCTYPE book [
  <!ELEMENT book (author,title,subtitle,date,publisher,ISBN)>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT subtitle (#PCDATA)>
  <!ELEMENT date (#PCDATA)>
  <!ELEMENT publisher (#PCDATA)>
  <!ELEMENT ISBN (#PCDATA)>
]>
```

## Ad. 5.

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="book">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="author" type="xs:string"/>
        <xs:element name="title" type="xs:string"/>
        <xs:element name="subtitle" type="xs:string"/>
        <xs:element name="date" type="xs:string"/>
        <xs:element name="publisher" type="xs:string"/>
        <xs:element name="ISBN" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```