

Semantic Web

FOAF and RDF Query Languages

F. Abel, N. Henze, and D. Krause

IVS Semantic Web Group

20.11.2008

- Social Networks - Networks of social interaction
- E.g.
 - in Academics: Co-authoring, advising, serving on communities
 - Movie staff: Directing and acting
 - between people by making phone calls or transmitting infections
 - between countries via trading relations
 - between papers through citation
 - between Web pages by hyperlinking them to other Web pages
 - ...

How to analyze social networks?

- one approach: analyze graph properties:
 - connectivity
 - distance
 - in-degree
 - out-degree
 - ...

Communities everywhere

Based on <http://www-106.ibm.com/developerworks/xml/library/x-foaf.html>

- Augment e-mail filtering by prioritizing mails from trusted colleagues
- Provide assistance to new entrants in a community
- Locate people with interests similar to yours

Based on <http://rdfweb.org/mt/foaflog/archives/2003/07/28/12.46.56/index.html>

- FOAF is the abbreviation for “Friend of a Friend”
- Vision: “FOAF is all about creating and using machine-readable homepages that describe people, the links between them and the things they create and do.”
<http://rdfweb.org/mt/foaflog/archives/2003/07/28/12.46.56/index.html>
- Schema: The FOAF Vocabulary Specification (Life Edition: May 19th, 2005: <http://xmlns.com/foaf/0.1/>)
- Basic ingredients: FOAF document format + Semantic Web idea:
 - people publish information in the FOAF document format
 - by following links between people’s descriptions, machines can identify the social network between all FOAF participants
 - based on this information, additional services become possible (e.g. expert search)

- **The unique name problem again:** When you start entering data about real-world things, like people, into a computer system, you come up against the difficulty of creating a unique name for each thing.
- Here: we are looking for a **decentralized naming strategy**
- What about using mail-addresses similar to the URI approach?
`mailto:edd@xml.com` lives in the UK
- ... but this is nonsense
- you should write
`mailto:edd@xml.com` is an e-mail address
- ... a person and that person's email address are not the same thing
- thus correctly you might write
The person with e-mail address `mailto:edd@xml.com` lives in the UK.

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

  <foaf:Person>
    <foaf:name>Edd Dumbill</foaf:name>
    <foaf:mbox rdf:resource="mailto:edd@xml.com" />
  </foaf:Person>

</rdf:RDF>
```

Example (describing a person)

```
<rdf:RDF
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <foaf:Person>
    <foaf:name>Dr. Stefan Decker</foaf:name>
    <foaf:homepage>
      http://www.isi.edu/~stefan
    </foaf:homepage>
    <foaf:title>Dr.</foaf:title>
    <foaf:depiction>
      http://www.deri.ie/images/members/stefan_decker.jpg
    </foaf:depiction>
    <foaf:family_name>Decker</foaf:family_name>
    <foaf:givenname>Stefan</foaf:givenname>
    <foaf:mbox rdf:resource="stefan.decker@deri.org" />
  </foaf:Person>
</rdf:RDF>
```

- The property **foaf:knows** can be used to link two people together, and it has the basic semantics of some kind of personal acquaintance.

Example (Establishing the acquaintance of two people)

```
<foaf:Person>
  <foaf:name>Edd Dumbill</foaf:name>
  <foaf:mbox rdf:resource="mailto:edd@xml.com" />
  ...
  <foaf:knows>
    <foaf:Person>
      <foaf:mbox
        rdf:resource="mailto:simon@xmlhack.com"/>
      <foaf:name>Simon St.Laurent</foaf:name>
    </foaf:Person>
  </foaf:knows>
</foaf:Person>
```

Property: foaf:knows

Based on <http://xmlns.com/foaf/0.1/>

- knows - A person known by this person (indicating some level of reciprocated interaction between the parties).
- The foaf:knows property relates a foaf:Person to another foaf:Person that he or she knows.
- Weak semantic specification (friend, acquaintance, co-worker?).
- No face-to-face knowledge of other person required
- Doesn't need to be complete, probably never will be complete (Web-like).
- To provide additional levels of representation beyond mere 'knows', FOAF applications can do several things.
 - specialize foaf:knows by adding subproperties.
 - use additional relations from other RDF schemas.

Based on *<http://xmlns.com/foaf/0.1/>*

- Better approach: encode mail addresses
- SHA1 (RFC3174) checksum is (with very high probability) still unique, and mail address unreadable

Example (Encoded mail address)

```
<foaf:Person>
  <foaf:name>Dan Brickley</foaf:name>
  <foaf:mbox_sha1sum>
    241021fb0e6289f92815fc210f9e9137262c252e
  </foaf:mbox_sha1sum>
  <foaf:homepage
    rdf:resource="http://rdfweb.org/people/danbri/" />
  <foaf:img
    rdf:resource=".../danbri/mugshot/danbri-small.jpeg" />
</foaf:Person>
```

- Many FOAF tools use foaf:mbox_sha1sum in preference to exposing mailbox information. This is usually for privacy and SPAM-avoidance reasons. Other relevant techniques include the use of PGP encryption and the use of FOAF-based whitelists for mail filtering.

- Problem: how to find your friend's FOAF file
- Solution: explicit link to RDF file(s) containing referred FOAF descriptions

- 1 Alter the `rdf:RDF` element to add the RDF Schema namespace (add `rdfs`), as follows:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

- 2 You can then add links to other FOAF descriptions by adding one `rdfs:seeAlso` element for each additional file, as follows:

```
<rdfs:seeAlso
  rdf:resource="http://www.example.com/friends.xrdf"/>
<rdfs:seeAlso
  rdf:resource="http://www.ldodds.com/webwho.xrdf"/>
```

Autodiscovery: associating FOAF information with your web pages

Based on <http://rdfweb.org/topic/Autodiscovery>

- Autodiscovery is a means for automatically finding machine-processable resources associated with a particular web page. The discovery can be automated because the resource is linked to in a well-defined way (using the HTML link tag).

- ```
<html>
<head>
<link rel="meta" type="application/rdf+xml"
 title="FOAF" href="foaf.rdf" />
</head>
...
</html>
```

- This seems a bit weird. Would a FOAF file be an "metainformation" for the document? If it is in a person's profile page, it would. However, this autodiscovery isn't used only in profile pages...

- Therefore, we need to indicate the relationship between FOAF files and documents. Here's one solution:

```
<link rel="meta FOAF.MAKER" type="application/rdf+xml"
 title="FOAF" href="foaf.rdf"/>
```

- but then: machines won't know the linktype "FOAF.MAKER", so we have to also include the path to the foaf schema document. The complete "semantic autodiscovery" will be like this. Thus:

```
<link rel="schema.FOAF" href="http://xmlns.com/foaf/0.1/" />
<link rel="meta FOAF.MAKER" type="application/rdf+xml"
 title="FOAF" href="foaf.rdf"/>
```

## FOAF Basics

- [Agent](#)
- [Person](#)
- [name](#)
- [nick](#)
- [title](#)
- [homepage](#)
- [mbox](#)
- [mbox\\_sha1sum](#)
- [img](#)
- [depiction](#) (depicts)
- [surname](#)
- [family\\_name](#)
- [givenname](#)
- [firstName](#)

## Personal Info

- [weblog](#)
- [knows](#)
- [interest](#)
- [currentProject](#)
- [pastProject](#)
- [plan](#)
- [based\\_near](#)
- [workplaceHomepage](#)
- [workInfoHomepage](#)
- [schoolHomepage](#)
- [topic\\_interest](#)
- [publications](#)
- [geekcode](#)
- [myersBriggs](#)
- [dnaChecksum](#)

## Online Accounts / IM

- [OnlineAccount](#)
- [OnlineChatAccount](#)
- [OnlineEcommerceAccount](#)
- [OnlineGamingAccount](#)
- [holdsAccount](#)
- [accountServiceHomepage](#)
- [accountName](#)
- [icqChatID](#)
- [msnChatID](#)
- [aimChatID](#)
- [jabberID](#)
- [yahooChatID](#)

## Projects and Groups

- [Project](#)
- [Organization](#)
- [Group](#)
- [member](#)
- [membershipClass](#)
- [fundedBy](#)
- [theme](#)

## Documents and Images

- [Document](#)
- [Image](#)
- [PersonalProfileDocument](#)
- [topic](#) (page)
- [primaryTopic](#)
- [tipjar](#)
- [sha1](#)
- [made](#) (maker)
- [thumbnail](#)
- [logo](#)

## Example

```
<foaf:Image
 rdf:about=".../mugshot/danbri-small.jpeg"/>
```

## Example

```
<foaf:Person rdf:nodeID="person1">
 <foaf:depiction
 rdf:resource=".../mugshot/danbri-small.jpeg"/>
</foaf:Person>
```

## Example

```
<foaf:Person>
 <foaf:name>Dan Brickley</foaf:name>
 <foaf:depiction>
 <foaf:Image/>
 </foaf:depiction>
</foaf:Person>
```

Examples from <http://rdfweb.org/topic/FoafExamples>

## Example

```
<foaf:Person>
 <foaf:name>Dan Brickley</foaf:name>
 <foaf:depiction>
 <foaf:Image>
 <foaf:sha1>7eff...</foaf:sha1>
 </foaf:Image>
 </foaf:depiction>
</foaf:Person>
```

## Example

```
<foaf:Person>
 <foaf:name>Dan Brickley</foaf:name>
 <foaf:depiction>
 <foaf:Image
 rdf:about="http://.../danbri/mugshot/danbri-small.jpeg">
 <foaf:sha1>7eff...</foaf:sha1>
 </foaf:Image>
 </foaf:depiction>
</foaf:Person>
```

## Example (Dr. Diederich's FOAF file)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<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:dc="http://purl.org/dc/elements/1.1/"
 xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
 xmlns:srw="http://purl.org/net/inkel/rdf/schemas/lang/1.1#">

 <foaf:PersonalProfileDocument rdf:about="">
 <foaf:maker rdf:nodeID="jd"/>
 <foaf:primaryTopic rdf:nodeID="jd"/>
 </foaf:PersonalProfileDocument>

 <foaf:Document rdf:about="">
 <dc:title>FOAF description of Joerg Diederich (mainly professional)
 </dc:title>
 <dc:description>
 Metadata about Joerg Diederich in Friend-Of-A-Friend (FOAF) format
 </dc:description>
 <foaf:topic rdf:nodeID="jd"/><foaf:maker rdf:nodeID="jd"/>
 </foaf:Document>
```

## Example (Dr. Diederich's FOAF file)

```
<foaf:Person rdf:nodeID="jd">
 <srw:masters>en</srw:masters>
 <srw:masters>de</srw:masters>
 <srw:reads>es</srw:reads>
 <srw:reads>sv</srw:reads>
 <foaf:name>Joerg Diederich</foaf:name>
 <foaf:title>Dr.-Ing.</foaf:title>
 <foaf:givenname>Joerg</foaf:givenname>
 <foaf:family_name>Diederich</foaf:family_name>
 <foaf:gender>male</foaf:gender>
 <foaf:mbox_sha1sum>0ff4e..</foaf:mbox_sha1sum>
 <foaf:img
 rdf:resource="http://l3s.de/images/mitarbeiter/joerg_diederich.jpg"/>
 <foaf:depiction
 rdf:resource="http://l3s.de/images/mitarbeiter/joerg_diederich.jpg"/>
 <foaf:schoolHomepage
 rdf:resource="http://www.gymnasium-gross-ilsede.de/">
 <foaf:pastProject rdf:resource="http://www.comcar.de/">
 <foaf:currentProject
 rdf:resource="http://knowledgeweb.semanticweb.org/">
 <foaf:currentProject rdf:resource="http://reverse.net/">
```

## Example (Dr. Diederich's FOAF file)

```
<foaf:publications>
 <foaf:Document>
 <foaf:page rdf:resource="../../../~diederich/publications.php"/>
 </foaf:Document>
</foaf:publications>

<foaf:workInfoHomepage
 rdf:resource="http://www.l3s.de/~diederich/research.php"/>
<foaf:based_near>
 <geo:Point>
 <geo:lat>52.3211</geo:lat>
 <geo:long>9.818</geo:long>
 </geo:Point>
</foaf:based_near>
<foaf:phone rdf:resource="tel:+49-511-762-9749"/>
<foaf:member>
 <foaf:Organization>
 <foaf:homepage rdf:resource="http://www.l3s.de/" />
 <dc:title>L3S Research Center</dc:title>
 </foaf:Organization>
</foaf:member>
<foaf:homepage rdf:resource="http://www.l3s.de/~diederich" />
```

## Example (Dr. Diederich's FOAF file)

```
<foaf:knows>
 <foaf:Person>
 <foaf:name>Rudi Studer</foaf:name>
 <foaf:mbox_sha1sum>be8...</foaf:mbox_sha1sum>
 <rdfs:seeAlso
 rdf:resource="...Personen/viewPersonFOAF/foaf_57.rdf" />
 </foaf:Person>
</foaf:knows>
...
<foaf:knows>
 <foaf:Person>
 <foaf:name>Holger Wache</foaf:name>
 <foaf:mbox_sha1sum>289...</foaf:mbox_sha1sum>
 <rdfs:seeAlso
 rdf:resource="http://www.cs.vu.nl/~holger/Holger%20Wache.rdf"/>
 </foaf:Person>
</foaf:knows>
...
</foaf:Person>
</rdf:RDF>
```

## foaf:Person

```
<rdfs:Class
 rdf:about="http://xmlns.com/foaf/0.1/Person"
 rdfs:label="Person"
 rdfs:comment="A person."
 vs:term_status="stable">

<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
<rdfs:subClassOf>
 <owl:Class rdf:about="http://xmlns.com/wordnet/1.6/Person"/>
</rdfs:subClassOf>
<rdfs:subClassOf>
 <owl:Class rdf:about="http://xmlns.com/foaf/0.1/Agent"/>
</rdfs:subClassOf>

...
<rdfs:isDefinedBy rdf:resource="http://xmlns.com/foaf/0.1/" />
<owl:disjointWith rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
<owl:disjointWith rdf:resource="http://xmlns.com/foaf/0.1/Organization"/>
<owl:disjointWith rdf:resource="http://xmlns.com/foaf/0.1/Project"/>

</rdfs:Class>
```

## foaf:Agent

```
<rdfs:Class rdf:about="http://xmlns.com/foaf/0.1/Agent"
 vs:term_status="unstable"
 rdfs:label="Agent"
 rdfs:comment="An agent (eg. person, group,...).">
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
 <rdfs:subClassOf>
 <owl:Class rdf:about="http://xmlns.com/wordnet/1.6/Agent-3"/>
 </rdfs:subClassOf>
 <owl:disjointWith rdf:resource="http://xmlns.com/foaf/0.1/Document"/>
</rdfs:Class>
```

## foaf:firstName

```
<rdf:Property rdf:about="http://xmlns.com/foaf/0.1/firstName"
 vs:term_status="testing"
 rdfs:label="firstName"
 rdfs:comment="The first name of a person.">
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
 <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
 <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
 <rdfs:isDefinedBy rdf:resource="http://xmlns.com/foaf/0.1/">
</rdf:Property>
```

## foaf:gender

```
<rdf:Property rdf:about="http://xmlns.com/foaf/0.1/gender"
 vs:term_status="testing"
 rdfs:label="gender"
 rdfs:comment="The gender of this Agent (typically but not
 necessarily 'male' or 'female').">
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#FunctionalProperty"/>
 <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#DatatypeProperty"/>
 <rdfs:domain rdf:resource="http://xmlns.com/foaf/0.1/Agent"/>
 <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/>
 <rdfs:isDefinedBy rdf:resource="http://xmlns.com/foaf/0.1/" />
 <!-- whatever one's gender is, and we are liberal in leaving room for more
 options than 'male' and 'female', we model this so that an agent has only
 one gender. -->
</rdf:Property>
```

# Exercise 1 - Writing your first FOAF document

- See Tutorial at  
<http://www.bulat.f0.ru/tutorials/FOAFtut/eng/>
- OR use FOAF-a-Matic  
<http://www.ldodds.com/foaf/foaf-a-matic>
- OR let inspire yourself by the examples at  
<http://rdfweb.org/topic/FoafExamples>
- SHA Codes <http://downlode.org/perl/sha1sum.pl>
- Check you FOAF file with the RDF Validator  
<http://www.w3.org/RDF/Validator/>.
- nice visual feedback can be obtained from the FOAF Explorer  
<http://xml.mfd-consult.dk/foaf/explorer/>

## Exercise 2 - SPARQL: Query the Starwars social network

```
<foaf:Person rdf:about="#Anakin">
 <foaf:name>Darth Vader</foaf:name>
 <foaf:familyname>Skywalker</foaf:familyname>
 <foaf:givenname>Anakin</foaf:givenname>
 <foaf:gender>male</foaf:gender>
</foaf:Person>

<foaf:Person rdf:about="#Padme">
 <foaf:name>Amidala Padme</foaf:name>
 <foaf:familyname>Padme</foaf:familyname>
 <foaf:givenname>Amidala</foaf:givenname>
 <foaf:gender>female</foaf:gender>
 <foaf:knows><foaf:Person rdf:ID="Anakin" /></foaf:knows>
 <foaf:knows><foaf:Person rdf:ID="Bail" /></foaf:knows>
</foaf:Person>

<foaf:Person rdf:about="#Bail">
 <foaf:name>Bail Oranga</foaf:name>
 <foaf:knows><foaf:Person rdf:ID="Padme" /></foaf:knows>
 <foaf:knows>
 <foaf:Person><foaf:name>Yoda</foaf:name></foaf:Person>
 </foaf:knows>
</foaf:Person>
```

Find SPARQL queries for these tasks:

- 1 Display the name of all involved persons.
- 2 Display all female persons and persons that don't have a gender.
- 3 Display all pairs of people that know each other.
- 4 Display all people that don't know Anakin.
- 5 Display all people that know somebody who knows Anakin.
- 6 Display all people that know somebody who knows Anakin and don't know Anakin directly.
- 7 Assume that the name is build out of givenname-whitespace-familyname. Find the guy for whom this is not true.
- 8 Put all different types of names from Persons into one variable.

Use the validator at <http://sparql.org/sparql.html> to check your queries.

- 1 Display the name of all involved persons.

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

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

```
SELECT ?name
```

```
FROM <http://www.kbs.uni-hannover.de/Lehre/semweb07/sparql-example.rdf>
```

```
WHERE {
```

```
 ?p rdf:type foaf:Person.
```

```
 ?p foaf:name ?name
```

```
}
```

- 2 Display all female persons and persons that don't have a gender.

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

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

```
SELECT ?person
```

```
FROM <http://www.kbs.uni-hannover.de/Lehre/semweb07/sparql-example.rdf>
```

```
WHERE { {?person rdf:type foaf:Person.
```

```
 ?person foaf:gender "female"}
```

```
UNION
```

```
{?person rdf:type foaf:Person.
```

```
 OPTIONAL {?person foaf:gender ?g}
```

```
 FILTER (!(bound(?g)))}
```

## 3 Display all pairs of people that know each other.

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

SELECT ?person1 ?person2
FROM <http://www.kbs.uni-hannover.de/Lehre/semweb07/sparql-example.rdf>
WHERE {
 ?person1 foaf:knows ?person2
}
```

## 4 Display all people that don't know Anakin.

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

SELECT ?person
FROM <http://www.kbs.uni-hannover.de/Lehre/semweb07/sparql-example.rdf>
WHERE {
 ?person rdf:type foaf:Person.
 OPTIONAL {?person foaf:knows ?a . FILTER (?a =<#Anakin>)}
 FILTER (!(bound(?a)))
}
```

- 5** Display all people that know somebody who knows Anakin.

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

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

```
SELECT ?person
FROM <http://www.kbs.uni-hannover.de/Lehre/semweb07/sparql-example.rdf>
WHERE {
 ?person rdf:type foaf:Person.
 ?person foaf:knows ?person2.
 ?person2 foaf:knows <#Anakin>
}
```

- 6** Display all people that know somebody who knows Anakin and don't know Anakin directly.

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

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

```
SELECT ?person
FROM <http://www.kbs.uni-hannover.de/Lehre/semweb07/sparql-example.rdf>
WHERE {
 ?person rdf:type foaf:Person.
 ?person foaf:knows ?person2. ?person2 foaf:knows <#Anakin>.
 OPTIONAL {?person foaf:knows ?a . FILTER (?a =<#Anakin>)}
 FILTER (!(bound(?a)))}
```

- 7 Assume that the name is build out of givenname-whitespace-familyname. Find the guy for whom this is not true.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX fn: <http://www.w3.org/2004/07/xpath-functions>

SELECT ?person
FROM <http://www.kbs.uni-hannover.de/Lehre/semweb07/sparql-example.rdf>
WHERE {
 ?person rdf:type foaf:Person.
 OPTIONAL {?person foaf:name ?name.
 ?person foaf:familyname ?familyname.
 ?person foaf:givenname ?givenname}
 FILTER (!(?name = fn:concat(?givenname," ",?familyname)))
}
```

fn:concat seems not to work in the validator.

- 8** Put all different types of names from Persons into one variable.

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

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

```
SELECT ?name
```

```
FROM <http://www.kbs.uni-hannover.de/Lehre/semweb07/sparql-example.rdf>
```

```
WHERE {
```

```
 {?person foaf:name ?name} UNION
```

```
 {?person foaf:familyname ?name} UNION
```

```
 {?person foaf:givenname ?name}
```

```
}
```