

An Introduction to FOAF

February 4, 2004

[Leigh Dodds \(/pub/au/15\)](#)

The FOAF (<http://www.foaf-project.org>) ("Friend of a Friend") project is a community driven effort to define an RDF vocabulary for expressing metadata about people, and their interests, relationships and activities. Founded by Dan Brickley and Libby Miller, FOAF is an open community-lead initiative which is tackling head-on the wider Semantic Web goal of creating a machine processable web of data. Achieving this goal quickly requires a network-effect that will rapidly yield a mass of data. Network effects mean people. It seems a fairly safe bet that any early Semantic Web successes are going to be riding on the back of people-centric applications. Indeed, arguably everything interesting that we might want to describe on the Semantic Web was created by or involves people in some form or another. And FOAF is all about people.

FOAF facilitates the creation of the Semantic Web equivalent of the archetypal personal homepage: My name is Leigh, this is a picture of me, I'm interested in XML, and here are some links to my friends. And just like the HTML version, FOAF documents can be linked together to form a web of data, with well-defined semantics.

Being an RDF application means that FOAF can claim the usual benefits of being easily harvested and aggregated. And like all RDF vocabularies it can be easily combined with other vocabularies, allowing the capture of a very rich set of metadata. This tutorial introduces the basic terms of the FOAF vocabulary, illustrating them with a number of examples. The article concludes with a brief review of the more interesting FOAF applications and considers some other uses for the data.

The FOAF Vocabulary

Like any well-behaved vocabulary FOAF publishes both its schema and specification at its namespace URI: <http://xmlns.com/foaf/0.1> (<http://xmlns.com/foaf/0.1>). The documentation is thorough and includes definitions of all classes and properties defined in the associated RDF schema. While the schema is embedded in the XHTML specification, it can also be accessed directly (<http://xmlns.com/foaf/0.1/index.rdf>).

Rather than cover the whole vocabulary, this article will focus on two of the most commonly used classes it defines: Person (http://xmlns.com/foaf/0.1/#term_Person) and Image (http://xmlns.com/foaf/0.1/#term_Image). The remaining definitions cover the description of documents, projects, groups, and organizations; consult the specification for more information. The community also has a lively mailing list (<http://rdfweb.org/mailman/listinfo/>

<http://www.ildt.bris.ac.uk/discovery/chatlogs/foaf/>), and project wiki (<http://rdfweb.org/topic/>) which serve as invaluable sources of additional information and discussion.

Care has been taken in the schema to ensure that, where appropriate, the FOAF classes have been related to their equivalents in other ontologies. This allows FOAF data to be immediately processable by applications built to understand these ontologies, while allowing the FOAF project to defer the definition of more complex concepts, e.g. geographical metadata, to other communities.

For example, while all of the FOAF classes are tied to a definition in [Wordnet](http://www.cogsci.princeton.edu/~wn/) (<http://www.cogsci.princeton.edu/~wn/>), via an RDF view of that data (<http://xmlns.com/2001/08/wordnet/>), the Person class is additionally tied to other schemas describing [contact](http://www.w3.org/2000/10/swap/pim/contact) (<http://www.w3.org/2000/10/swap/pim/contact>) and [geographic](http://www.w3.org/2003/01/geo/) (<http://www.w3.org/2003/01/geo/>) related data.

Personal Metadata

The Person class is the core of the FOAF vocabulary. A simple example will illustrate it's basic usage:

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

  <foaf:Person>

    <foaf:name>Peter Parker</foaf:name>

    <foaf:mbox rdf:resource="mailto:peter.parker@dailybugle.com"/>

  </foaf:Person>

</rdf:RDF>
```

In other words, there is a person, with the name " Peter Parker ", who has an email address of " peter.parker@dailybugle.com ".

Publishing data containing plain text email addresses is just asking for trouble; to avoid this FOAF defines another property, `foaf:mbox_sha1sum` whose value is a SHA1 encoded email address complete with the `mailto:` URI scheme prefix. The FOAF project wiki has a handy reference page pointing to a number of [different ways of generating a SHA1 sum](http://rdfweb.org/topic/HashPrograms) (<http://rdfweb.org/topic/HashPrograms>).

The end result of applying this algorithm is a string unique to a given email address (or mailbox). The next example demonstrates the use of this and several other new properties that further describe Peter Parker.

```
<foaf:Person>

  <foaf:name>Peter Parker</foaf:name>

  <foaf:gender>Male</foaf:gender>

  <foaf:title>Mr</foaf:title>

  <foaf:givenname>Peter</foaf:givenname>

  <foaf:family_name>Parker</foaf:family_name>

  <foaf:mbox_sha1sum>cf2f4bd069302febd8d7c26d803f63fa7f20bd82</foaf:mbox_s

  <foaf:homepage rdf:resource="http://www.peterparker.com"/>

  <foaf:weblog rdf:resource="http://www.peterparker.com/blog/" />

</foaf:Person>
```

Which is a slightly richer description of Peter Parker, including some granularity in the markup of his name through the use of `foaf:title`, `foaf:givenname`, and `foaf:family_name`. We also now know that Peter Parker is male (`foaf:gender`) and has both a homepage (`foaf:homepage`) and a weblog (`foaf:weblog`).

Identifying Marks

Keen-eyed RDF enthusiasts will already have noticed that neither of these examples assigns a URI to the resource called Peter Parker, i.e. there is no `rdf:about` attribute on the `foaf:Person` resource:

```
<foaf:Person rdf:about="..uri to identify peter..." />
```

That's because there is still some debate around both the social and technical implications of assigning URIs to people. Which URI identifies you? Who assigns these URIs? What problems are associated with having multiple URIs (assigned by different people) for the same person? Side-stepping this potential minefield, FOAF borrows the concept of an "[inverse functional pr](#)

property (<http://www.w3.org/TR/owl-ref/#InverseFunctionalProperty-def>)" (IFP) from OWL, the Web Ontology Language (<http://www.w3.org/TR/owl-features/>). An inverse functional property is simply a property whose value uniquely identifies a resource.

The FOAF schema defines several inverse functional properties, including `foaf:mbox`, `foaf:mbox_sha1sum`, and `foaf:homepage`; consult the schema documentation for the complete list. An application harvesting FOAF data can, on encountering two resources that have the same values for an inverse functional property, safely merge the description of each and the relations of which they are part. This process, often referred to as "[smushing](http://rdfweb.org/topic/Smushing)" (<http://rdfweb.org/topic/Smushing>), must be carried out when aggregating FOAF data to ensure that data about different resources is correctly merged.

As an example consider the following RDF fragment:

```
<foaf:Person>

  <foaf:name>Peter Parker</foaf:name>

  <foaf:mbox_sha1sum>cf2f4bd069302febd8d7c26d803f63fa7f20bd82</foaf:mbox_s

</foaf:Person>

<foaf:Person>

  <foaf:name>Spiderman</foaf:name>

  <foaf:mbox_sha1sum>cf2f4bd069302febd8d7c26d803f63fa7f20bd82</foaf:mbox_s

</foaf:Person>
```

Applying our knowledge that `foaf:mbox:sha1sum` is an inverse functional property, we can merge the descriptions together to discover that these statements actually describe a single person. Spiderman is unmasked! While perfectly valid, it may not be desirable in all circumstances, and flags the importance of FOAF aggregators recording the source (provenance) of their data. This allows incorrect and potentially malicious data to be identified and isolated.

Before moving on it's worth noting that while FOAF defines the email address properties (`foaf:mbox_sha1sum` and `foaf:mbox`) as uniquely identifying a person, this is not the same thing as saying that all email addresses are owned by a unique person. What the FOAF schema claims is that any email address used in a `foaf:mbox` (or encoded as a `foaf:mbox_sha1sum`) property uniquely identifies a person. If it doesn't, then it's not a suitable value for that property.

It's Who You Know

Having captured some basic metadata about Peter Parker, it's time to go a step further and begin describing his relationships with others. The `foaf:knows` property is used to assert that there is some relationship between two people. Precisely what this relationship is, and whether it's reciprocal (i.e. if you know me, do I automatically know you?), is deliberately left undefined.

For obvious reasons, modeling interpersonal relationships can be a tricky business. The FOAF project has therefore taken the prudent step of simply allowing a relationship to be defined without additional qualification. It is up to other communities (and vocabularies) to further define different types of relationships.

Using `foaf:knows` is simple: one `foaf:Person` `foaf:knows` another. The following example shows two alternative ways of writing this using the RDF/XML syntax. The first uses a cross-reference to a person defined in the same document (using `rdf:nodeID`), while the second describes the `foaf:Person` "in situ" within the `foaf:knows` property. The end result is the same: Peter Parker knows both Aunt May and Harry Osborn.

```
<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#">
```

```
<foaf:Person rdf:nodeID="harry">
```

```
    <foaf:name>Harry Osborn</foaf:name>
```

```
    <rdfs:seeAlso rdf:resource="http://www.osborn.com/harry.rdf"/>
```

```
</foaf:Person>
```

```
<foaf:Person>
```

```
    <foaf:name>Peter Parker</foaf:name>
```

```
    <foaf:knows rdf:nodeID="harry"/>
```

```
<foaf:knows>
```

```
    <foaf:Person>
```

```
        <foaf:name>Aunt May</foaf:name>
```

```
    </foaf:Person>
```

```
</foaf:knows>
```

```
</foaf:Person>
```

</rdf:RDF>

The other thing to notice is that, in addition to the `foaf:knows` relationship between Peter and Harry, a link has also been introduced to Harry's own FOAF document, using the `rdfs:seeAlso` property. Defined by the RDF Schema specification, the `rdfs:seeAlso` property indicates a resource that may contain additional information about its associated resource. In this case it's being used to point to Harry Osborn's own FOAF description.

It's through the use of the `rdfs:seeAlso` property that FOAF can be used to build a web of machine-processable metadata; `rdfs:seeAlso` is to RDF what the anchor element is to HTML. Applications can be written to spider (or "[scutter](http://rdfweb.org/topic/Scutter) (<http://rdfweb.org/topic/Scutter>)" using the FOAF community's terminology) these [RDF hyperlinks](http://www.w3.org/2001/sw/Europe/talks/xml2003/Overview-3.html) (<http://www.w3.org/2001/sw/Europe/talks/xml2003/Overview-3.html>) to build a database of FOAF data.

Finer-grained Relationships

The loose definition of `foaf:knows` won't fit all applications, particularly those geared to capture information about complex social and business networks. However, this doesn't mean that FOAF is unsuitable for such purposes; indeed FOAF has the potential to be an open interchange format used by many different social networking applications.

The expectation is that additional vocabularies will be created to refine the general FOAF `knows` relationship to create something more specific. The correct way to achieve this is to declare new sub-properties of `foaf:knows`. Stepping outside of FOAF for a moment, we can briefly demonstrate one example of this using [the relationship schema](http://www.perceive.net/schemas/20021119/relationship/) (<http://www.perceive.net/schemas/20021119/relationship/>) created by Eric Vitiello.

The relationship schema defines a number of sub-properties of `foaf:knows` including `parentOf`, `siblingOf`, `friendOf`, etc. The following example uses these properties to make some clearer statements about the relationships between Peter Parker and some of his contemporaries:

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

```
<foaf:Person rdf:ID="spiderman">
    <foaf:name>Spiderman</foaf:name>
    <rel:enemyOf rdf:resource="#green-goblin"/>
</foaf:Person>
```

```
<foaf:Person rdf:ID="green-goblin">
    <foaf:name>Green Goblin</foaf:name>
    <rel:enemyOf rdf:resource="#spiderman"/>
</foaf:Person>
```

```
<foaf:Person rdf:ID="peter">
    <foaf:name>Peter Parker</foaf:name>
    <rel:friendOf rdf:resource="#harry"/>
</foaf:Person>
```

```
<foaf:Person rdf:ID="harry">
    <foaf:name>Harry Osborn</foaf:name>
```



```
<rel:friendOf rdf:resource="#peter"/>
```

```
<rel:childOf rdf:resource="#norman"/>
```

```
</foaf:Person>
```

```
<foaf:Person rdf:ID="norman">
```

```
<foaf:name>Norman Osborn</foaf:name>
```

```
<rel:parentOf rdf:resource="#harry"/>
```

```
</foaf:Person>
```

```
</rdf:RDF>
```

While it is possible to model quite fine-grained relationships using this method, the most interesting applications will be those that can infer relationships between people based on other metadata. For example, have they collaborated on the same project, worked for the same company, or been pictured together in the same image? Which brings us to the other commonly used FOAF class, Image (http://xmlns.com/foaf/0.1/#term_Image).

Image is Everything

Digital cameras being all the rage these days, it's not surprising that many people are interested in capturing metadata about their pictures. FOAF provides for this use case in several ways. First, using the `foaf:depiction` property we can make a statement that says "this person (Resource) is shown in this image". FOAF also supports an inverse of this property (`foaf:depicts`) that allows us to make statements of the form: "this image is a picture of this Resource". The following example illustrates both of these properties.

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

```
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
```

```
    xmlns:dc="http://purl.org/dc/elements/1.1/">
```

```
<foaf:Person rdf:ID="peter">
```

```
    <foaf:name>Peter Parker</foaf:name>
```

```
    <foaf:depicts rdf:resource="http://www.peterparker.com/peter.jpg"/>
```

```
</foaf:Person>
```

```
<foaf:Person rdf:ID="spiderman">
```

```
    <foaf:name>Spiderman</foaf:name>
```

```
</foaf:Person>
```

```
<foaf:Person rdf:ID="green-goblin">
```

```
    <foaf:name>Green Goblin</foaf:name>
```

```
</foaf:Person>
```

```
<!-- codepiction -->
```

```
<foaf:Image rdf:about="http://www.peterparker.com/photos/spiderman/statue.;
```

```
<dc:title>Battle on the Statue Of Liberty</dc:title>
```

```
<foaf:depicts rdf:resource="#spiderman"/>
```

```
<foaf:depicts rdf:resource="#green-goblin"/>
```

```
<foaf:maker rdf:resource="#peter"/>
```

```
</foaf:Image>
```

```
</rdf:RDF>
```



This RDF instances says that the image at <http://www.peterparker.com/peter.jpg> is a picture of Peter Parker. It also defines a `foaf:Image` resource, i.e. an image which can be found at a specific URI, that depicts both Spiderman and the Green Goblin. Elements from the [Dublin Core](http://www.dublincore.org/) (<http://www.dublincore.org/>) namespace are often added to FOAF documents to title images, documents, etc.

Notice also that Peter Parker is defined as the author of the image using the `foaf:maker` property which is used to relate a resource to its creator. The `dc:creator` term isn't used here due to some [issues with its loose definition](http://rdfweb.org/topic/UsingDublinCoreCreator) (<http://rdfweb.org/topic/UsingDublinCoreCreator>).

Publishing FOAF Data

Having created an RDF document containing FOAF terms and copied it to the Web, the next step is to link the new information into the existing web of FOAF data. There are several ways of doing this:

- *through foaf:knows* – ensuring that people who know you link to your FOAF data via `rdfs:seeAlso` link will make the data discoverable
- *through the FOAF Bulletin Board* – [a wiki page](http://rdfweb.org/topic/FOAFBulletinBoard) (<http://rdfweb.org/topic/FOAFBulletinBoard>) that links to dozens of FOAF files. FOAF harvesters generally include [the RDF view](http://jibbering.com/2002/10/rdf-wiki.1) (<http://jibbering.com/2002/10/rdf-wiki.1>) of this page as one of their starting locations.
- *through auto-discovery* – the FOAF project has defined a means to [link to a FOAF document from an HTML page](http://rdfweb.org/topic/Autodiscovery) (<http://rdfweb.org/topic/Autodiscovery>) using the link element; several tools now support this mechanism

Having covered the basics of the FOAF vocabulary and published some data, it's time to consider what applications are out there making use of it.

FOAF Applications

The FOAF application most immediately useful to the owner of a freshly published FOAF description is Morten Frederikson's FOAF Explorer (<http://xml.mfd-consult.dk/foaf/explorer/>) which can generate a HTML view of FOAF data, complete with referenced images and links to other data. For example here is a view of my FOAF description (<http://xml.mfd-consult.dk/foaf/explorer/?foaf=http://www.ldodds.com/ldodds.rdf>).

FOAF Explorer provides an effective way to browse the network of FOAF data. With the addition of a Javascript bookmarklet (<http://www.ldodds.com/blog/archives/000026.html>) to perform auto-discovery, it's easy to jump from a blog posting to a description of that person and their interests.

However the most elegant way to browse the relationships to be found in the network of FOAF data is by using Jim Ley's foafnaut (<http://www.foafnaut.org/>): an SVG application that provides a neat visualiaation of `foaf:knows` relationships. Here's the foafnaut view starting from my description (<http://blub.foafnaut.org/?sha1=1bca73e5c6916c738d6ec7cc0597ad0e395e7ace>).

There are a number of other interesting FOAF applications. plink (<http://www.plink.org>) is a social networking site. foafbot (<http://usefulinc.com/foaf/foafbot>) and whwhwhwh (<http://swordfish.rdfweb.org/discovery/2003/10/whwhwhwh/>) are IRC bots that provide conversational interfaces onto FOAF data. Libby Miller's codepiction (<http://swordfish.rdfweb.org/discovery/2001/08/codepict/>) experiments demonstrate a novel way to explore FOAF image metadata.

Beyond these initial developments, FOAF has potential in many other areas. For example, painful web site registrations can become a thing of the past: just indicate the location of your FOAF description. Throw in the relationships and FOAF can be used as an interchange format between social networking sites, building an open infrastructure that allows end users to retain control over their own data.

As an example consider ecommerce sites like Amazon, which have become successful because of their high levels of personalization. Getting the most from these sites involves a learning process where they discover your interests either through explicit preference setting or adapting product suggestions based on a purchase history. Using FOAF there's the potential to capture this information once, in a form that can be used by not one site, but many. The user could then move freely between systems.

To summarize, FOAF is a lively project exploring the role that person-centric metadata can help in delivering on the promise of the Semantic Web. Whether or not this grand vision is ultimately realized, FOAF has many immediate applications. And what's more, it's fun.

Content licensed from and © 1998 - 2008 O'Reilly Media, Inc.