An indexing and retrieval engine for the Semantic Web

Tim Finin
University of Maryland, Baltimore County
20 May 2004

(Slides at: http://ebiquity.umbc.edu/v2.1/resource/html/id/26/)
http://swoogle.umbc.edu/

Swoogle is a crawler based search and retrieval system for semantic web documents.
Acknowledgements

• Contributors include Tim Finin, Anupam Joshi, Yun Peng, R. Scott Cost, Joel Sachs, Pavan Reddivari, Vishal Doshi, Rong Pan, Li Ding, and Drew Ogle.

• Partial research support was provided by DARPA contract F30602-00-0591 and by NSF by awards NSF-ITR-IIS-0326460 and NSF-ITR-IDM-0219649.
Swoogle in ten easy steps

(1) Concept and motivation
(2) Swoogle Architecture
(3) Crawling the semantic web
(4) Semantic web metadata
(5) Ontology rank
(6) IR on the semantic web
(7) Current results
(8) Future work
(9) Conclusions
(10) demo…
(1) Concepts and Motivation

- Google has made us all smarter
- Software agents will need something similar to maximize the use of information on the semantic web.
Concepts and Motivation

Semantic web researchers need to understand how people are using the concepts & languages and might want to ask questions like:

– What graph properties does the semantic web exhibit?
– How many OWL files are there?
– Which are the most popular ontologies?
– What are all the ontologies that are about time?
– What documents use terms from the ontology http://daml.umbc.edu/ontologies/cobra/0.4/agent?
– What ontologies map their vocabulary to http://reliant.teknowledge.com/DAML/SUMO.owl?
Concepts and Motivation

Semantic web tools may need to find ontologies on a given topic or similar to another one.

- UMCP’s SMORE annotation editor helps a user add annotations to a text document, an image, or a spreadsheet.
- It suggests ontologies and terms that may be relevant to express the user’s annotations.
- How can it find relevant ontologies?
Concepts and Motivation

• Spire is an NSF supported project exploring how the SW can support science research and education

• Our focus is on Ecoinformatics

• We need to help users find relevant SW ontologies, data, and services

• Without being overwhelmed with irrelevant ones
Related work on Ontology repositories

• Two models: Metadata repositories vs. Ontology Management Systems

• Some examples of web-based metadata repositories
  – http://daml.org/ontologies
  – http://schemaweb.info/
  – http://www.semanticwebsearch.com/

• Ontology management systems
  – Stanford’s Ontolingua (http://www.ksl.stanford.edu/software/ontolingua/)
  – IBM’s Snobase (http://www.alphaworks.ibm.com/tech/snobase/)

• Swoogle is in the first set, but aims to be (1) comprehensive, (2) compute more metadata, (3) offer unique search and browsing components and (4) support web and agent services.
Example Queries and Services

- What documents use/are used (directly/indirectly) by ontology X?
- Monitor any ontology used by document X (directly or indirectly) for changes
- Find ontologies that are similar to ‘http://…’
- Let me browse ontologies w.r.t. the scienceTopics topic hierarchy.
- Find ontologies that include the strings ‘time day hour before during date after temporal event interval’
- Show me all of the ontologies used by the ‘National Cancer Institute’
(2) Architecture

- **Web interface**
- **Apache/Tomcat**
  - php, myAdmin
- **Agent services**
- **Web services**
- **Ontology Analyzer**
- **Ontology Agents**
- **DB**
- **IR engine**
- **SIRE**
- **cached files**
- **Focused Crawler**
- **SWD crawler**
- **Ontology discovery**
- **Google**
- **APIs**

**Languages and Technologies**
- Apache/Tomcat
- php, myAdmin
- MySQL
- Apache/Tomcat
- myAdmin
- MySQL
- Jena
- Jena
- SIRE
- **Web services**
- **Agent services**
- **Web interface**
Database schemata

http://pear.cs.umbc.edu/myAdmin/
Database schemata

~ 10,000 SWDs and counting
Database schemata

SWD relations
Interfaces

• Swoogle has interfaces for people (developers and users) and will expose APIs.
• Human interfaces are primarily web-based but may also include email alerts.
• Programmatic interfaces will be offered as web services and/or agent-based services (e.g., via FIPA).
(3) Crawling the semantic web

Swoogle uses two kinds of crawlers as well as conventional search engines to discover SWDs.

– A focused crawler crawls through HTML files for SWD references

– A SWD crawler crawls through SWD documents to find more SWD references.

– Google is used to find likely SWD files using key words (e.g., rdfs) and filetypes (e.g., .rdf, .owl) on sites known to have SWDs.
Priming the crawlers

The crawlers need initial URIs with which to start

– Using global Google queries (Google API)
– Results obtained by scraping sites like daml.org, and schemaweb.info
– URLs submitted by people via the web interface
Priming the Crawler

• Googled for files with the extension of rdf, rdfs, foaf, daml, oil, owl, and n3, but Google returns only the first 1000 results.

<table>
<thead>
<tr>
<th>QUERY</th>
<th>RESULTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>filetype:rdf</td>
<td>rdf 230,000</td>
</tr>
<tr>
<td>filetype:n3</td>
<td>prefix 3220</td>
</tr>
<tr>
<td>filetype:owl</td>
<td>owl 1590</td>
</tr>
<tr>
<td>filetype:owl</td>
<td>rdf 1040</td>
</tr>
<tr>
<td>filetype:rdfs</td>
<td>rdfs 460</td>
</tr>
<tr>
<td>filetype:foaf</td>
<td>foaf 27</td>
</tr>
<tr>
<td>filetype:oil</td>
<td>rdf 15</td>
</tr>
</tbody>
</table>

• The daml.org crawler has ~21K URLs, 75% of which are hosted at teknowledge. Most are HTML files with embedded DAML, automatically generated from wordnet.

• Schemaweb.info has ~100 URLs
SWD Crawler

- We started with the OCRA Ontology Crawler by Jen Golbeck of the Mindswap Lab
- Uses Jena to read URIs and convert to triples.
- When crawler sees an URI, gets date from http header and inserts/uploads Ontology table depending upon whether entry is already present in DB or is a new one.
- Each URI in a triple is potentially a new SWD and, if it is, should be crawled.
Crawler approach

• Then based on the each triple’s subject, object and predicate enters data into ontologyrelation table in DB.

• Relation can be IM, EX, PV, TM or IN depending on predicate.

• Also a count is maintained for same source, destination, relation entries.
  
  – e.g., TM(http://foo.com/A.owl, http://foo.com/B.owl, 19) indicates that A used terms from B 19 times.
Recognizing SWD

• Every URI in a triple potentially references a SWD
  – But many reference HTML documents, images, mailtos, etc.

• Summarily reject
  – URIs in the have seen table
  – URIs with common non-SWD extensions (e.g. .jpg, .mp3)

• Try to read with Jena
  – Does it throw an exception?

• Apply a heuristic classifier
  – To recognize intended SWDs that are malformed
(4) Semantic Web Metadata

- Swoogle stores metadata, not content
  - About documents, classes, properties, servers, …
  - The boundary between metadata and content is fuzzy
- The metadata come from (1) the documents themselves, (2) human users, (3) algorithms and heuristics and (4) other SW sources
  1: SWD3 hasTriples 341, SWD3 dc:creator P31
  2: User54 claims [SWD3 topic:isAbout sci:Biology]
  3: SWD3 endorsedBy User54
  4: P31 foaf:knows P256
Direct document metadata

• OWL and RDF encourage the inclusion of metadata in documents
• Some properties have defined meaning
  – owl:priorVersion
• Others have very conventional use
  – attaching rdf:comment and rdf:label to documents
• Others are rather common
  – Using dc:creator to assert a document’s author.
Some Computed Document Metadata

• Simple
  – Type: SWO, SWI or mixed
  – Language: RDF, DAML+OIL, OWL (lite, DL, Full)
  – Statistics: # of classes, properties, triples defined/used
  – Results of various kinds of validation tests
  – Classes and properties defined/used

• Document properties
  – Date modified, crawled, accessibility history
  – Size in bytes
  – Server hosting document

• Relations between documents
  – Versions (partial order)
  – Direct/indirect imports, references, extends,
  – Existence of mapping assertion (e.g., owl:sameClass)
Some Class and Property Metadata

• For a class or property X
  – Number of times document D uses X
  – Which documents (partially) define X

• For classes
  – Subclasses and superClasses

• For properties
  – Domain and range
  – SubProperties and SuperProperties
User Provided Metadata

• We can collect more metadata by allowing users to add annotations about any document
  – To fill in “missing metadata” (e.g., who the author is, what appropriate topics are)
  – To add evaluative assertions (e.g., endorsements, comments on coverage)

• Such information must be stored with provenance data

• A trust model can be employed to decide what metadata to use for a given application
Other Derived Metadata

• Various algorithms and heuristics can be used to compute additional metadata

• Examples:
  – Compute document similarity from statistical similarities between text representations
  – Compute document topics from topics of similar documents, documents extended, other documents by same author, etc.
Relations among SWDs

- **Binary: R(D1,D2)**
  - **IM**: owl:imports
  - **IMstar**: transitive closure of IM
  - **EX**: SWD1 extends D2 by defines classes or properties subsumed by those in D2
  - **PV**: owl:priorVersion or it’s subclasses
  - **TM**: D1 uses terms from D2
  - **IN**: D1 uses an individual defined in D2
  - **MP**: D1 maps some of its terms to D2’s using owl:sameClass, etc.

- **Ternary: R(D1,D2,D3)**
  - D1 maps a term from D2 to D3 using owl:sameClass, etc.
(5) Ranking SWDs

• Ranking pages w.r.t. their intrinsic importance, popularity or trust has proven to be very useful for web search engines.
• Related ideas from the web include Google’s *PageRank* and *HITS*
• The ideas must be adapted for use on the semantic web
Google’s PageRank

- The rank of a page is a function of how many links point to it and the rank of the pages hosting those links.
- The “random searcher” model provides the intuition:
  1. Jump to a random page
  2. Select and follow a random link on the page and repeat (2) until ‘bored’
  3. If bored, go to (1)
- Pages are ranked according to the relative frequency with which they are visited.
PageRank

- The formula for computing page A’s rank is

\[ P(A) = (1 - d) + d \left( \sum_{i=1}^{n} \frac{P(T_i)}{C(T_i)} \right) \]

- Where
  - Ti are the pages that link to A
  - C(A): # of links out of A
  - d is a damping factor (e.g., 0.85)

- Compute by iterating until a fixed point is reached or until changes are very small
HITS

• Hyperlink-Induced Topic Search divides pages relating to a topic into three groups
  – **Authorities**: pages with good content about a topic, linked to by many hubs
  – **Hubs**: pages that link to many good authority pages on a topic (directories)
  – **Others**

• Iteratively calculate hub and authority scores for each page in neighborhood and rank results accordingly
  – Document that many pages point to is a good authority
  – Document that points to many authorities is a good hub, pointing to many good authorities makes for an even better hub

The web, like Gaul, is divided into three parts:
- The regular web (e.g. HTML pages)
- Semantic Web Ontologies (SWOs)
- Semantic Web Instance files (SWIs)
Heuristics distinguish SWOs & SWIs
• SWOs mostly reference other SWOs
• SWIs reference SWOs, other SWIs and the regular web
• There aren’t standards yet for referencing SWDs from the regular web
SWD Rank

Until standards or at least conventions develop for linking from the regular web to SWDs we will ignore the regular web.

- The random surfer model seems reasonable for ranking SWIs, but not for SWOs.
- An issue is whether a SWD’s rank is divided and spread over the SWDs it links to.
- If a SWO imports/extends/refers to N SWOs, all must be read.
- If a SWD uses a SWO’s term, it may be diluted.
- Another issue is whether all links are equal to the surfer.
- The surfer may prefer to click an Extends link rather than an use_INdividual link to learn more knowledge.

Flowchart:

- Jump to a random page
  - SWO?
    - yes
      - Explore all linked SWOs
    - no
      - bored?
        - yes
          - Follow a random link
        - no
          - Explore all linked SWOs
Current formula

- Step 1
  \[ rawPR(A) = (1 - d) + d \sum_{i=1}^{n} rawPR(X_i) \frac{flow(X_i, A)}{flow(X_i)} \]

  \[ flow(X_i, A) = \sum_{i \in \text{links}(X_i, A)} \text{weight}(l) \]

  \[ flow(X_i) = \sum_{j=1}^{m} f(X_i, A_j) \]

- Step 2
  - Rank of a SWI: \( PR(A) = rawPR(A) \)
  - Rank of a SWO:
    \[ PR(A) = \sum_{X_i \in TC(A)} rawPR(X_i) \]
  
  where TC(A) is the transitive closure of SWOs

- Each relation has a weight (IM=8, EX=4, TM=2, P=1, ...)
- Step 1 simulates an agent surfing through SWIs.
- Step 2 models the rational behavior of the agent in that all imported SWOs are visited
(6) IR on the semantic web

- Why use information retrieval techniques?
- Several approaches under evaluation:
  - Character ngrams
  - URIs as words
  - Swangling to make SWDs Google friendly
- Work in progress
Why use IR techniques?

• We will want to retrieve over the structured and unstructured parts of a SWD
• We should prepare for the appearance of Text documents with embedded SW markup
• We may want to get our SWDs into conventional search engines, such as Google.
• IR techniques also have some unique characteristics that may be very useful
  – e.g., ranking matches, computing the similarity between two documents, relevance feedback, etc.
Swoogle IR Search

• This is work in progress, not yet integrated into Swoogle

• Documents are put into an ngram IR engine (after processing by Jena) in canonical XML form
  – Each contiguous sequence of N characters is used as an index term (e.g., N=5)
  – Queries processed the same way

• Character ngrams work almost as well as words but have some advantages
  – No tokenization, so works well with artificial languages and agglutinative languages
  => good for RDF!
Why character n-grams?

• Suppose we want to find ontologies for time
• We might use the following query
  “time temporal interval point before after during day month year eventually calendar clock duration end begin zone”
• And have matches for documents with URIs like
  – http://foo.com/timeont.owl#timeInterval
  – http://foo.com/timeont.owl#CalendarClockInterval
  – http://purl.org/upper/temporal/t13.owl#timeThing
Another approach: URIs as words

• Remember: ontologies define vocabularies
• In OWL, URIs of classes and properties are the words
• So, take a SWD, reduce to triples, extract the URIs (with duplicates), discard URIs for blank nodes, hash each URI to a token (use MD5Hash), and index the document.
• Process queries in the same way
• Variation: include literal data (e.g., strings) too.
Harnessing Google

- Google started indexing RDF documents some time in late 2003
- Can we take advantage of this?
- We’ve developed techniques to get some structured data to be indexed by Google
- And then later retrieved
- Technique: give Google enhanced documents with additional annotations containing Swangle Terms™
Swangle definition

swan·gle

Pronunciation: ‘swa[ng]-g&l
Function: transitive verb
Inflected Forms: swan·gled; swan·gling /-g(&-)li[ng]/
Etymology: Postmodern English, from C++ mangle,
Date: 20th century

1: to convert an RDF triple into one or more IR
   indexing terms

2: to process a document or query so that its content
   bearing markup will be indexed by an IR system

Synonym: see tblify

- swan·gler  /-g(&-)l&r/ noun
Swangling

• Swangling turns a SW triple into 7 word like terms
  – One for each non-empty subset of the three components with the missing elements replaced by the special “don’t care” URI
  – Terms generated by a hashing function (e.g., MD5)

• Swangling an RDF document means adding in triples with swangle terms.
  – This can be indexed and retrieved via conventional search engines like Google

• Allows one to search for a SWD with a triple that claims “Ossama bin Laden is located at X”
A Swangled Triple

<rdf:RDF
  xmlns:s="http://swoogle.umbc.edu/ontologies/swangle.owl#"
</rdf>

<s:SwangledTriple>
  <s:swangledText>N656WNTZ36KQ5PX6RFUGVKQ63A</s:swangledText>
  <rdfs:comment>Swangled text for
  [http://www.xfront.com/owl/ontologies/camera/#Camera,
  http://www.w3.org/2000/01/rdf-schema#subClassOf,
</s:SwangledTriple>

<s:SwangledText>M6IMWPWIH4YQI4IMGZYBGPYKEI</s:SwangledText>
<s:SwangledText>HO2H3FOPAEM53AQIZ6YVPFQ2XI</s:SwangledText>
<s:SwangledText>2AQEUJOYPMXWKHZTENIJS6PQ6M</s:SwangledText>
<s:SwangledText>IIVQRXOAYRH6GGRZDFXKEEB4PY</s:SwangledText>
<s:SwangledText>75Q5Z3BYAKRPLZDLFNS5KKMTOY</s:SwangledText>
<s:SwangledText>2FQ2Y17SNJ7OMXOXIDEEE2WOZU</s:SwangledText>
What’s the point?

• We’d like to get our documents into Google
• The Swangle terms look like words to Google and other search engines.
• We use *cloaking* to avoid having to modify the document
  – Add rules to the web server so that, when a search spider asks for document X the document swangled(X) is returned
• Caching makes this efficient
(7) Current status (5/19/2004)

• Swoogle’s database
  ~11K SWDs (25% ontologies), ~100K document relations, 1 registered user

• Swoogle 2’s database
  ~58K SWDs (10% Ontologies), ~87K classes, ~47K properties, 224K individuals, …

• FOAF dataset
  ~1.6M foaf rdf documents identified, ~800K analyzed

• Web site is functional and usable, though incomplete
• Some bugs (e.g., #triples etc reported wrongly in some cases)
• IR component is not yet integrated in
• Please use and provide feedback
• Submit URLs
(8) Future work

• Swoogle 2 (summer 2004)
  – More metadata about more documents
  – Scaling up requires more robustness
  – Document topics
• FOAF dataset (summer 2004)
• From our todo list…(2004-2005)
  – Add non RDF ontologies (e.g., glossaries)
  – Publish a monthly one-page state of the semantic web report
  – Add a trust model for user annotations
  – Implement web and agent services and build into tools (e.g., annotation editor)
  – Visualization tools
**Swoogle\(^2\)**

- Prototype exists with minimal interfaces
- Goals: more metadata, millions of documents
- More heuristics for finding SWDs
- More objects (e.g., sites) and relations
- Records unique classes and properties and their metadata and relations e.g.,
  - property: domain, range, …
  - definesProperty(SWD, property)
  - usesProperty(SWD, property, N)
Studying FOAF files

- FOAF (Friend of a Friend) is a simple ontology for describing people and their social networks.
  - See the foaf project page: http://www.foaf-project.org/
- We recently crawled the web and discovered ~1.6M RDF FOAF files.
  - Most of these are from the http://liveJournal.com/ blogging system which encodes basic user info in foaf
  - See http://apple.cs.umbc.edu/semdis/wob/foaf/

```xml
<foaf:Person>
  <foaf:name>Tim Finin</foaf:name>
  <foaf:mbox_sha1sum>2410..37262c252e</foaf:mbox_sha1sum>
  <foaf:homepage rdf:resource="http://umbc.edu/~finin/" />
  <foaf:img rdf:resource="http://umbc.edu/~finin/images/passport.gif" />
</foaf:Person>
```
## FOAF Vocabulary

### Basics
- Agent
- Person
- name
- nick
- title
- homepage
- mbox
- mbox_sha1sum
- img
- depiction
- surname
- family_name
- givenname
- firstName

### Personal Info
- weblog
- knows
- interest
- currentProject
- pastProject
- plan
- based_near
- workplaceHomepage
- workInfoHomepage
- schoolHomepage
- topic_interest
- publications
- geekcode
- myersBriggs
- dnaChecksum

### Documents & Images
- Document
- Image
- PersonalProfileDocument
- topic (page)
- primaryTopic
- tipjar
- sha1
- made (maker)
- thumbnail
- logo

### Projects & Groups
- Project
- Organization
- Group
- member
- membershipClass
- fundedBy
- theme

### Online Accts
- OnlineAccount
- OnlineChatAccount
- OnlineEcommerceAccount
- OnlineGamingAccount
- holdsAccount
- accountServiceHomepage
- accountName
- icqChatID
- msnChatID
- aimChatID
- jabberID
- yahooChatID
FOAF: why RDF? Extensibility!

- FOAF vocabulary provides 50+ basic terms for making simple claims about people
- FOAF files can use other RDF terms too: RSS, MusicBrainz, Dublin Core, Wordnet, Creative Commons, blood types, starsigns, …
- RDF guarantees freedom of independent extension
  - OWL provides fancier data-merging facilities
- **Result:** Freedom to say what you like, using any RDF markup you want, and have RDF crawlers merge your FOAF documents with other’s and know when you’re talking about the same entities.

*After Dan Brickley, danbri@w3.org*
No free lunch!

- We must plan for lies, mischief, mistakes, stale data, slander
- The data is out of control, distributed, dynamic
- Importance of knowing who-said-what
  - Anyone can describe anyone
  - We must record data provenance
  - Modeling and reasoning about trust is critical
- Legal, privacy and etiquette issues emerge
- Welcome to the real world

After Dan Brickley, danbri@w3.org
Swoogle 2 FOAF dataset

- As of May 19, 2004 ~1.6M FOAF documents identified and about 1/2 analyzed
  - Using 3353 unique classes
  - Using 5618 unique properties
  - From 6066 unique servers
  - Defining ~2M individuals
A subset of 1000 FOAF files
## Popular FOAF Properties

This table is based on the statistics of 453867 foaf files (Sun, 2 May 2004 22:07:12 -0400)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Popularity (Number of occurrence)</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://xmlns.com/foaf/0.1/nick">http://xmlns.com/foaf/0.1/nick</a></td>
<td>452122</td>
<td>0.996155</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://xmlns.com/foaf/0.1/inbox_sha1sum">http://xmlns.com/foaf/0.1/inbox_sha1sum</a></td>
<td>448950</td>
<td>0.989166</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://xmlns.com/foaf/0.1/weblog">http://xmlns.com/foaf/0.1/weblog</a></td>
<td>447703</td>
<td>0.986419</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://xmlns.com/foaf/0.1/page">http://xmlns.com/foaf/0.1/page</a></td>
<td>436881</td>
<td>0.962575</td>
</tr>
<tr>
<td>5</td>
<td><a href="http://xmlns.com/foaf/0.1/knowns">http://xmlns.com/foaf/0.1/knowns</a></td>
<td>432278</td>
<td>0.952433</td>
</tr>
<tr>
<td>6</td>
<td><a href="http://xmlns.com/foaf/0.1/interest">http://xmlns.com/foaf/0.1/interest</a></td>
<td>355204</td>
<td>0.782617</td>
</tr>
<tr>
<td>7</td>
<td><a href="http://xmlns.com/foaf/0.1/dateOfBirth">http://xmlns.com/foaf/0.1/dateOfBirth</a></td>
<td>327835</td>
<td>0.722315</td>
</tr>
<tr>
<td>8</td>
<td><a href="http://xmlns.com/foaf/0.1/aimChatID">http://xmlns.com/foaf/0.1/aimChatID</a></td>
<td>179200</td>
<td>0.394829</td>
</tr>
<tr>
<td>9</td>
<td><a href="http://xmlns.com/foaf/0.1/homepage">http://xmlns.com/foaf/0.1/homepage</a></td>
<td>163774</td>
<td>0.360841</td>
</tr>
<tr>
<td>10</td>
<td><a href="http://xmlns.com/foaf/0.1/yahooChatID">http://xmlns.com/foaf/0.1/yahooChatID</a></td>
<td>75496</td>
<td>0.166339</td>
</tr>
<tr>
<td>11</td>
<td><a href="http://xmlns.com/foaf/0.1/msnChatID">http://xmlns.com/foaf/0.1/msnChatID</a></td>
<td>64300</td>
<td>0.141671</td>
</tr>
<tr>
<td>12</td>
<td><a href="http://xmlns.com/foaf/0.1/icqChatID">http://xmlns.com/foaf/0.1/icqChatID</a></td>
<td>38955</td>
<td>0.085829</td>
</tr>
<tr>
<td>13</td>
<td><a href="http://xmlns.com/foaf/0.1/name">http://xmlns.com/foaf/0.1/name</a></td>
<td>11087</td>
<td>0.024428</td>
</tr>
<tr>
<td>14</td>
<td><a href="http://xmlns.com/foaf/0.1/jabberID">http://xmlns.com/foaf/0.1/jabberID</a></td>
<td>7206</td>
<td>0.015877</td>
</tr>
<tr>
<td>15</td>
<td><a href="http://xmlns.com/foaf/0.1/firstName">http://xmlns.com/foaf/0.1/firstName</a></td>
<td>6931</td>
<td>0.015271</td>
</tr>
<tr>
<td>16</td>
<td><a href="http://xmlns.com/foaf/0.1/surname">http://xmlns.com/foaf/0.1/surname</a></td>
<td>6926</td>
<td>0.015260</td>
</tr>
<tr>
<td>17</td>
<td><a href="http://xmlns.com/foaf/0.1/title">http://xmlns.com/foaf/0.1/title</a></td>
<td>6635</td>
<td>0.014619</td>
</tr>
</tbody>
</table>
FOAF dataset in Swoogle 2


### Sample Foaf files

This table is generated at Thu, 20 May 2004 14:04:13 -0400

<table>
<thead>
<tr>
<th>No</th>
<th>Foaf URL</th>
<th>Optic</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td><a href="http://cenc.cconu.edu.cn/lb/foaf">http://cenc.cconu.edu.cn/lb/foaf</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td><a href="http://cyber.law.harvard.edu/blog/foaf">http://cyber.law.harvard.edu/blog/foaf</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td><a href="http://ksc.w3.org/Archives/Public/foaf/">http://ksc.w3.org/Archives/Public/foaf/</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td><a href="http://kse.w3.org/Archives/Public/foaf/">http://kse.w3.org/Archives/Public/foaf/</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td><a href="http://kse.w3.org/Archives/Public/foaf/">http://kse.w3.org/Archives/Public/foaf/</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td><a href="http://kse.w3.org/Archives/Public/foaf/">http://kse.w3.org/Archives/Public/foaf/</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td><a href="http://people.msu.edu/~alexei/foaf">http://people.msu.edu/~alexei/foaf</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td><a href="http://student.van.abeka.edu/foaf">http://student.van.abeka.edu/foaf</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td><a href="http://swolfish.rdfweb.org/people">http://swolfish.rdfweb.org/people</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td><a href="http://www.libr.alberta.ca/foaf">http://www.libr.alberta.ca/foaf</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Popular Properties

This table is based on the statistics of 5238 foaf files, which use properties 169369 times (rdfs:type is excluded) (generated Thu, 20 May 2004 14:07:31 -0400)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name</th>
<th>Popularity (Number of references)</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://xmlns.com/foaf:0.1/name">http://xmlns.com/foaf:0.1/name</a></td>
<td>22542</td>
<td>0.133094</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://xmlns.com/foaf:0.1/known">http://xmlns.com/foaf:0.1/known</a></td>
<td>21250</td>
<td>0.125466</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://xmlns.com/foaf:0.1/hasPage">http://xmlns.com/foaf:0.1/hasPage</a></td>
<td>16002</td>
<td>0.099794</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://xmlns.com/foaf:0.1/hasPrimaryPage">http://xmlns.com/foaf:0.1/hasPrimaryPage</a></td>
<td>13773</td>
<td>0.081319</td>
</tr>
<tr>
<td>5</td>
<td>foaf:schmidtName.Also</td>
<td>10000</td>
<td>0.062819</td>
</tr>
<tr>
<td>6</td>
<td>hasBlog</td>
<td>9891</td>
<td>0.053143</td>
</tr>
<tr>
<td>7</td>
<td>hasWeblog</td>
<td>4720</td>
<td>0.027921</td>
</tr>
<tr>
<td>8</td>
<td>hasWeblog</td>
<td>4000</td>
<td>0.022664</td>
</tr>
<tr>
<td>9</td>
<td>hasBlog</td>
<td>3387</td>
<td>0.019998</td>
</tr>
</tbody>
</table>

### Popular Classes

This table is based on the statistics of 5238 files, which have 55667 instances (ontology classes are excluded) (Thu, 20 May 2004 14:06:01 -0400)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Class Name</th>
<th>Popularity (Number of instances)</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://xmlns.com/foaf:0.1/Person">http://xmlns.com/foaf:0.1/Person</a></td>
<td>20396</td>
<td>0.755208</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://purl.org/0.1/912041">http://purl.org/0.1/912041</a></td>
<td>1559</td>
<td>0.047210</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://xmlns.com/foaf:0.1/Document">http://xmlns.com/foaf:0.1/Document</a></td>
<td>1330</td>
<td>0.036102</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://musicbrainz.org/maximum-2.1/Track">http://musicbrainz.org/maximum-2.1/Track</a></td>
<td>940</td>
<td>0.026697</td>
</tr>
<tr>
<td>5</td>
<td><a href="http://musicbrainz.org/maximum-2.1/Track">http://musicbrainz.org/maximum-2.1/Track</a></td>
<td>940</td>
<td>0.026697</td>
</tr>
<tr>
<td>6</td>
<td><a href="http://xmlns.com/foaf:0.1/Image">http://xmlns.com/foaf:0.1/Image</a></td>
<td>552</td>
<td>0.014576</td>
</tr>
<tr>
<td>7</td>
<td><a href="http://xmlns.com/foaf:0.1/Person">http://xmlns.com/foaf:0.1/Person</a></td>
<td>415</td>
<td>0.010635</td>
</tr>
<tr>
<td>8</td>
<td><a href="http://purl.org/0.1/channel">http://purl.org/0.1/channel</a></td>
<td>311</td>
<td>0.008720</td>
</tr>
<tr>
<td>9</td>
<td><a href="http://purl.org/0.1/channel">http://purl.org/0.1/channel</a></td>
<td>292</td>
<td>0.008187</td>
</tr>
<tr>
<td>10</td>
<td><a href="http://purl.org/0.1/channel">http://purl.org/0.1/channel</a></td>
<td>215</td>
<td>0.006028</td>
</tr>
</tbody>
</table>
What are SWDs about?

• We might want to browse SWDs via a topic hierarchy, a la Yahoo (Swahoo?)
• Users doing searches might want to restrict their search to ontologies about, say, *Biology*
• Idea: build topic hierarchies using a simple topic ontology, e.g., see
  – http://swoogle.umbc.edu/ontologies/sciences.owl
• Associate SWDs with one or more topics drawn from appropriate topic hierarchies
Who’s going to add those associations?

• People will assert some initially, e.g.,
  – SWD X is about sciences:microbiology and sciences:genomics
  – All SWDs on http://lisp.com/ontologies/ are about it:computer programming and about it:lisp

• And heuristics can infer or learn more associations
  – If A extends B, then A is about whatever B is about
  – All SWDs authored by X are about sciences:space

• A trust model might be needed here
(9) Conclusions

• Search engines have taken the web to a new level
• The semantic web will need them too.
• SW search engines can compute richer meta data and relations
• Working on Swoogle is a lot of fun
• We think it will be useful
• It should be a good testbed for more research
What will Google do?

• The web search companies are tracking the SW
• But waiting until there is significant use before getting serious
  – Significant for Google probably means $10^{7}$ pages
  – Google did recently started indexing XML encoded documents, albeit in a simple way
• Caution: processing SWDs is inherently more expensive
(10) Demo

http://swoogle.umbc.edu/