

Trust Networks on the Semantic Web

Jennifer Golbeck¹, Bijan Parsia¹, James Hendler¹

¹University of Maryland, College Park
A. V. Williams Building
College Park, Maryland 20742
golbeck@cs.umd.edu, hendler@cs.umd.edu, bparsia@isr.umd.edu

Abstract. The so-called "Web of Trust" is one of the ultimate goals of the Semantic Web. Research on the topic of trust in this domain has focused largely on digital signatures, certificates, and authentication. At the same time, there is a wealth of research into trust and social networks in the physical world. In this paper, we describe an approach for integrating the two to build a web of trust in a more social respect. This paper describes the applicability of social network analysis to the semantic web, particularly discussing the multi-dimensional networks that evolve from ontological trust specifications. As a demonstration of algorithms used to infer trust relationships, we present several tools that allow users to take advantage of trust metrics that use the network.

1 Introduction

"Trust" is a word that has come to have several very specific definitions on the Semantic Web. Much research has focused on authentication of resources, including work on digital signatures and public keys. Confidence in the source or author of a document is important, but trust, in this sense, ignores many important points.

Just because a person can confirm the source of documents does not have any explicit implication about trusting the content of those documents. This project addresses "trust" as credibility or reliability in a much more human sense. It opens up the door for questions like "how much credence should I give to what this person says about a given topic," and "based on what my friends say, how much should I trust this new person?"

In this paper, we will discuss the application of a social network to the semantic web. Section 2 discusses how to build a meaningful social network from the architecture of the semantic web, and how it conveys meaning about the structure of the world. Section 3 will describe the implementation of such a network. We describe a sample ontology, an algorithm for computing trust in a network, and present tools that use this network to provide users with information about the reputation of others.

1.2 Related Work

This paper uses techniques developed in the field of social network analysis, and applies that to the issue of trust on the semantic web. This section describes the most relevant works from each area.

1.2.1 Social Networks

Social networks have a long history of study in a wide range of disciplines. The more mathematical of the studies have appeared in the "Small World" literature. The work on Small Worlds, also commonly known as "Six Degrees of Separation" originated out of Stanley Milgram's work in the 1960s. His original studies indicated that any two people in the world were separated by only a small number acquaintances (theorized to be six in the original work) [18]. Since then, studies have shown that many complex networks share the common features of the small-world phenomenon: small average distance between nodes, and a high connectance, or clustering coefficient [21].

Small world networks have been studied in relation to random graphs [29]. For social systems, both models have been used to describe phenomena such as scientific collaboration networks [20], and models of game theory [29]. The propagation of effects through these types of networks has been studied, particularly with respect to the spread of disease [19, 7]. The web itself has shown the patterns of a small world network, in clustering and diameter [4, 1].

Viewing the current web as a graph, where each page represents a node, and the hyperlinks translate to directed edges between nodes, has produced some interesting results. The main focus of this research has been to improve the quality of search [7,6,5,8,14,26]. Other work has used this structure for classification [9] and community discovery [15].

1.2.2 Trust on the Semantic Web

Yolanda Gil and Varun Ratnakar addressed the issue of trusting content and information sources [12]. They describe an approach to derive assessments about information sources based on individual feedback about the sources. As users add annotations, they can include measures of Credibility and Reliability about a statement, which are later averaged and presented to the viewer. Using the TRELIS system, users can view information, annotations (including averages of credibility, reliability, and other ratings), and then make an analysis.

Calculating trust automatically for an individual in a network on the web is partially addressed in Raph Levin's Advogato project [17]. His trust metric uses group assertions for determining membership within a group. The Advogato [3] website, for example, certifies users at three levels. Access to post and edit website information is controlled by these certifications. On any network, the Advogato trust metric is extremely attack resistant. By identifying individual nodes as "bad" and finding any nodes that certify the "bad" nodes, the metric cuts out an unreliable portion of the network. Calculations are based primarily on the good nodes, so the network as a whole remains secure.

2 Networks on the Semantic Web

Studying the structure of the hypertext web can be used to find community structure in a limited way. A set of pages clustered by hyperlinks may indicate a common topic among the pages, but it does not show more than a generic relationship among the pages. Furthermore, pages with fewer outgoing links are less likely to show up in a cluster at all because their connectance is obviously lower. These two facts make it difficult for a person to actually see any relationship among specific concepts on the web as it currently stands – classification is not specific enough, and it relies on heavy hyperlinking that may not be present.

The Semantic Web changes this. Since the semantic data is machine-understandable, there is no need to use heuristics to relate pages. Concepts in semantically marked up pages are automatically linked, relating both pages and concepts across a distributed web.

By its nature, the semantic web is one large graph. Resources (and literals) are connected by predicates. Throughout the rest of this paper, we will refer to resources as objects, and predicates as properties. Mapping objects to nodes and properties to labeled edges in a graph yields the power to use the algorithms and methods of analysis that have been developed for other manifestations of graphs.

While the graph of the entire Semantic Web itself is interesting, a subgraph generated by restricting the properties, and thus edges, to a subset of interest allows us to see the relationships among distributed data. Applications in this space are vast. Semantic markup means that retrieving instances of specific classes and the set of properties required for a particular project, becomes easy. Furthermore, merging data collected from many different places on the web is trivial.

Generating social networks on the semantic web is a similar task with useful results. Information about individuals in a network is maintained in distributed sources. Individuals can manage data about themselves and their friends. Security measures, like digital signatures of files, go some way toward preventing false information from propagating through the network. This security measure builds trust about the authenticity of data contained within the network, but does not describe trust between people in the network.

There are many measures of "trust" within a social network. It is common in a network that trust is based simply on knowing someone. By treating a "Person" as a node, and the "knows" relationship as an edge, an undirected graph emerges (Figure 1 shows the graph of acquaintances used in this study). If A does not know B, but some of A's friends know B, A is "close" to knowing B in some sense. Many existing networks take this measure of closeness into account. We may, for example, reasonably trust a person with a small Erdos number to have a stronger knowledge of graph theory than someone with a large or infinite number.

Techniques developed to study naturally occurring social networks apply to these networks derived from the semantic web. Small world models describe a number of algorithms for understanding relationships between nodes. The same algorithms that model the spread of disease [19, 7] in physical social networks, can be used to track the spread of viruses via email.

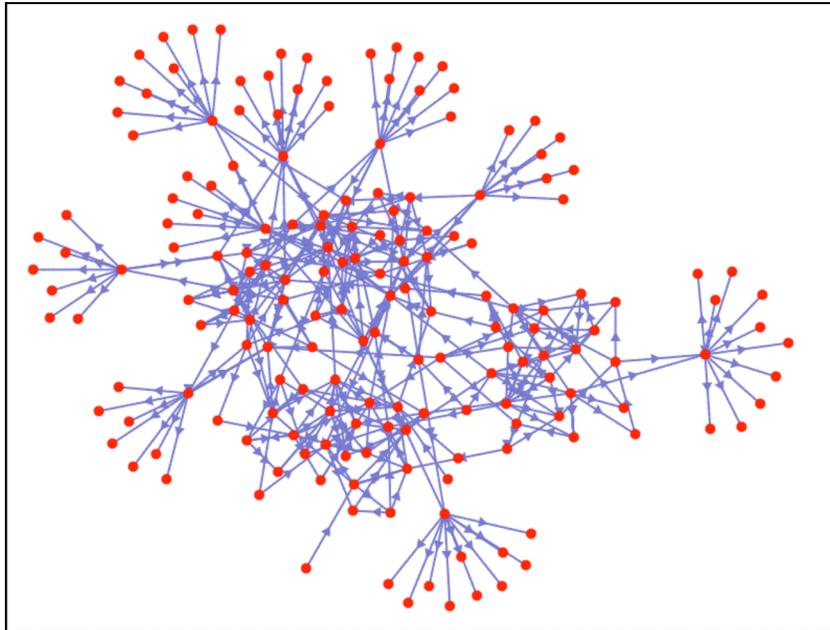


Fig.1. The Trust Network generated for this paper

For trust, however, there are several other factors to consider. Edges in a trust network are directed. A may trust B, but B may not trust A back. Edges are also weighted with some measure of the trust between two people. By building such a network, it is possible to infer how much A should trust an unknown individual based on how much A's friends and friends-of-friends trust that person. Using the edges that exist in the graph, we can infer an estimation of the weight of a non-existent edge. In the next section, we describe one method for making such inferences in the context of two implementations.

3 Implementation

The semantic web of trust requires that users describe their beliefs about others. Once a person has a file that lists who they know and how much they trust them, social information can be automatically compiled and processed.

3.1 Base Ontology

Friend-Of-A-Friend (FOAF) [10] is one project that allows users to create and interlink statements about who they know, building a web of acquaintances. The FOAF schema [24] is an RDF vocabulary that a web user can use to describe information about himself, such as name, email address, and homepage, as well as information about people he knows. In line with the security mentioned before, users can sign these files so information will be attributed to either a known source, or an explicitly anonymous source. People are identified in FOAF by their email addresses, since they are unique for each person.

In this project, we introduce a schema, designed to extend foaf:Person, which allows users to indicate a level of trust for people they know [28]. Since FOAF is used as the base, users are still identified by their email address. Our trust schema adds properties with a domain of foaf:Person. Each of these new properties specifies one level of trust on a scale of 1-9. The levels roughly correspond to the following:

1. Distrusts absolutely
2. Distrusts highly
3. Distrusts moderately
4. Distrusts slightly
5. Trusts neutrally
6. Trusts slightly
7. Trusts moderately
8. Trusts highly
9. Trusts absolutely

Trust can be given in general, or limited to a specific topic. Users can specify several trust levels for a person on several different subject areas. Users can specify topic specific trust levels to refine the network. For example, Bob may trust Dan highly regarding research topics, but distrust him absolutely when it comes to repairing cars.

Using the trust ontology, the different trust ratings (i.e. "distrustsAbsolutely," "trustsModerately," etc.) are properties of the "Person" class, with a range of another "Person". These properties are used for general trust, and are encoded as follows:

```
<Person rdf:ID="Joe">  
  <mbox rdf:resource="mailto:bob@example.com" />  
  <trustsHighly rdf:resource="#Sue" />  
</Person>
```

Another set of properties are defined for trust in a specific area. They correspond to the nine values above, but are indicated as trust regarding a specific topic (i.e. "distrustsAbsolutelyRe," "trustsModeratelyRe," etc). The range of these topic specific properties is the "TrustsRegarding" class, which has been defined to group a Person and a subject of trust together. The "TrustsRegarding" class has two properties: "trustsPerson" indicates the person being trusted, and "trustsOnSubject" indicates the subject that the trust is about. There are no range restrictions on this latter property, which leaves it to the user to specify any subject from any ontology. Consider the

following example, that shows syntax for trusting one person relative to several different topics:

```
<Person rdf:ID="Bob">

  <mbox rdf:resource="mailto:joe@example.com" />

  <trustsHighlyRe>
    <TrustsRegarding>
      <trustsPerson rdf:resource="#Dan" />
      <trustsOnSubject
        rdf:resource="http://example.com/ont#Research" />
      </TrustsRegarding>
    </trustsHighlyRe>

    <distrustsAbsolutelyRe>
      <TrustsRegarding>
        <trustsPerson rdf:resource="#Dan" />
        <trustsOnSubject
          rdf:resource="http://example.com/ont#AutoRepair" />
        </TrustsRegarding>
      </distrustsAbsolutelyRe>

</Person>
```

With topic-specific information provided, queries can ask for trust levels on a given subject and the network is then trimmed down to include only the relevant edges.

Large networks form when users link from their FOAF file to other FOAF files. In the example above, the can include a reference to the trust file for "Dan". Spidering along these links allow a few seeds to produce a large graph. Many interesting projects have emerged from these FOAF networks, including the Co-depiction project [25], SVG image annotation [2], and FoafNaut [11], an IRC bot that provides FOAF data about anyone in its network [10].

Having specified trust levels as described above, the network that emerges becomes much more useful and interesting. A directed, weighted graph is generated, where an edge from A to B indicates that A has some trust relationship with B. The weight of that edge reflects the specified trust level.

3.2 Computing Trust

Direct edges between individual nodes in this graph contain an explicitly expressed trust value. Beyond knowing that a given user explicitly trusts another, the graph can be used to infer the trust that one user should have for individuals to whom they are not directly connected.

Several basic calculations are made over the network.

- **Maximum and minimum capacity paths:** In the algorithms used in this research, trust follows the standard rules of network capacity – on any given path, the maximum amount of trust that the source can give to the sink is limited to be no larger than the smallest edge weight along that path. The maximum and minimum capacity path functions identify the trust capacity of the paths with the highest and lowest capacity respectively. This is useful in identifying the range of trust given by neighbors of X to Y.
- **Maximum and minimum length paths:** In addition to knowing the varying weights of paths from X to Y, it is useful to know how many steps it takes to reach Y from X. These path length functions return full the chains of weighted edges that make up the maximum and minimum length paths.
- **Weighted average:** In the algorithm described below, this value represents a recommended trust level for X to Y.

The weighted average is designed to give a calculated recommendation on how much the source should trust the sink. The average, according to the formula (1), is the default trust metric used in all applications querying this network. It is designed to be a very simple metric, and it is reasonable that different users would want to use different metrics. In our metric, the maximum capacity of each path is used as the trust value for that path. There is no diminution for long path lengths. Users may want a lower trust rating for someone many links away as opposed to a direct neighbor. Another issue deals with "trust" and "distrust." The maximum capacity calculation does not assign any external value to the trust ratings; they are considered simply as real values. Since there is a social relevance to those numbers, variants of the single path trust value calculation may be desirable. For example, if A distrusts B regarding a specific subject and in turn, B distrusts C on that subject, it is possible that A will want to give C a relatively *higher* rating. That is, if A and B hold opposite views, as do B and C, it may mean that A and C are actually close to one another. Alternately, A may want to distrust C even *more* that A distrusts B, the logic being if A distrusts B, and B cannot even trust C, then C must be especially untrustworthy.

The variants on the trust metric make a long list, and it is unrealistic to provide different functions for each variant. Instead, we have designed an interface where users can create and use their own metrics. This process is described in section 3.3.

By default, trust is calculated according to the following simple function. Using the maximum capacity of each path to the sink, a simple recursive algorithm is applied to calculate the average. For any node that has a direct edge to the sink, we ignore paths and use the weight of that edge as its value. For any node that is not directly connected to the sink, the value is determined by a weighted average of the values for each of its neighbors who have a path to the sink. The calculated trust t from node i to node s is given by the following function:

$$t_{is} = \frac{\sum_{j=0}^n \left\{ \begin{array}{l} (t_{js} * t_{ij}) \text{ if } t_{ij} \geq t_{js} \\ (t_{ij}^2) \text{ if } t_{ij} < t_{js} \end{array} \right\}}{\sum_{j=0}^n t_{ij}} \quad (1)$$

where i has n neighbors with paths to s . In calculating the average, this formula also ensures that we do not trust someone down the line more than we trust any intermediary.

Though in the current implementation, our metric searches the entire graph, a simple modification could easily limit the number of paths searched from each node. Since this is a social network that follows the small world pattern, the average distance across the network increases only logarithmically with the number of nodes. Thus, with a limit on the number of paths searched, the algorithm will scale well.

3.3 Trust Web Service

A web service is a function or collection of functions that can be accessed over the web. Access to the trust network is available as a web service. Web users can provide two email addresses, and the weighted average trust value, calculated as described above is returned. The benefit of a web service is its accessibility. It can be composed with other web services or used as a component of another application. This also makes access to trust values available for agents to use as components in intelligent web applications.

The web service interface also allows users to supply their own algorithms for calculating trust. We provide access to a Java API for querying the trust graph. This includes functions such as retrieving a list of neighbors, getting the trust rating for a given edge, detecting the presence or absence of paths between two individuals, and finding path length. With this information, users can write small Java programs to calculate trust based to their own algorithms. The corresponding class file for this user-defined Java code is passed as an argument to our Trust Web Service, along with the source and sink email addresses, and our service will implement the user defined function.

4 Applications

4.1 TrustBot

TrustBot is an IRC bot that has implemented algorithms to make trust recommendations to users based on the trust network it builds. It allows users to transparently interact with the graph by making a simple series of queries.

At runtime, and before joining an IRC network, TrustBot builds an internal representation of the trust network from a collection of distributed sources. Users can add their own URIs to the bot at any time, and incorporate the data into the graph. The bot keeps a collection of these URIs that are spidered when the bot is launched or called upon to reload the graph. From an IRC channel, the bot can be queried to provide the weighted average, as well as maximum and minimum path lengths, and maximum and minimum capacity paths. The TrustBot is currently running on icr.freenode.net, and can be queried under the nick 'TrustBot'.

4.2 TrustMail

TrustMail is an email client, developed on top of Mozilla Messenger, that provides an inline trust rating for each email message. Since our graph uses email address as a unique identifier, it is a natural application to use the trust graph to rank email. As with TrustBot, users can configure TrustMail to show trust levels for the mail sender either on a general level or with respect to a certain topic.

To generate ratings, TrustMail makes a call to the web service, passing in the email address of the sender and the address of the mailbox to where the message was delivered. It is necessary to use the mailbox email address instead of the "to" address on the email to prevent a lack of data because of cc-ed, bcc-ed, and mailing list messages. Figure 2 shows a sample inbox and message with trust ratings.

There are two points to note about the figure below. First is that trust ratings are calculated with respect to the topic of email (using the TrustsRegarding framework described above). If a user has a trust rating with respect to email, that value is used. If there is no trust rating specifically with respect to email, but a general trust rating is available, the latter value is used.

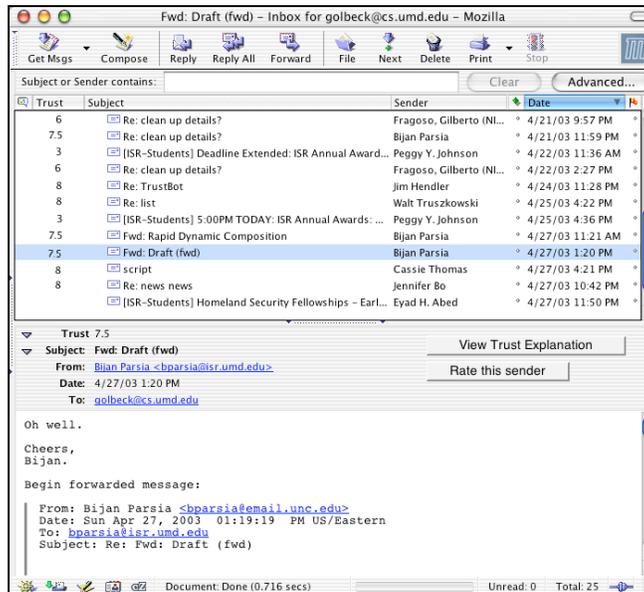


Fig. 2: Trust Mail with trust ratings

Consider the case of two research groups working on a project together. The professors that head each group know one another, and each professor knows the students in her own group. However, neither is familiar with the students from the other group. If, as part of the project, a student sends an email to the other group's professor, how will the professor know that the message is from someone worth paying attention to? Since the name is unfamiliar, the message is not distinguishable from other, not-so-important mail in the inbox. This scenario is exactly the type of situation that TrustMail improves upon. The professors need only to rate their own students and the other professor. Since the trust algorithms looks for *paths* in the graph (and not just direct edges), there will be a path from professor to professor to student. Thus, even though the student and professor have never met or exchanged correspondence, the student gets a high rating because of the intermediate relationship. If it turns out that one of the students is sending junk type messages, but the network is producing a high rating, the professor can simply add a direct rating for that sender, downgrading the trust level. That will not override anyone else's direct ratings, but will be factored into ratings where the professor is an intermediate step in the path.

The ratings alongside messages are useful, not only for their value, but because they basically replicate the way trust relationships, and awareness thereof, work in social settings. For example, today, it would sensible and polite for our student from above to start of the unsolicited email with some indication of the relationships between the student and the two professors, e.g., "My advisor has collaborated with you on this topic in the past and she suggested I contact you." Upon receiving such a note, the professor might check with her colleague that the student's claims were correct, or

just take those claims at face value, extending trust and attention to the student on the basis of the presumed relationship. The effort needed to verify the student by phone, email, or even walking down the hall weighed against the possible harm of taking the student seriously tends to make extending trust blindly worthwhile. TrustMail lowers the cost of sharing trust judgments even in across widely dispersed and rarely interacting groups of people, at least in the context of email. It does so by gathering machine readably encoded assertions about people and their trustworthiness, reasoning about those assertions, and then presenting those augmented assertions in an end user friendly way.

Extending this partial automation of trust judgments to other contexts can be as simple as altering the end user interface appropriate, as we did in earlier work where we presented the trust service as Internet Relay Chat “bot”[30]. However, one might expect other situations to demand different inference procedures (which our service supports), or even different networks of relations. In the current Web, the latter would be difficult to achieve as there is only one sort of link. On the Semantic Web, we can overlay many different patterns of trust relation over the same set of nodes simple by altering the predicate we focus on.

5 Conclusions and Future Work

This paper illustrates a method for creating a trust network on the semantic web. By introducing an ontology, an algorithm for finding trust from the resulting network, and different methods for accessing the network through a web service or applications, this is a first step for showing how non-security based efforts can become part of the foundation of the web of trust.

As work in this area progresses, developers should consider more in depth investigation of algorithms for calculating trust. The algorithm here was designed to be a simple proof-of-concept. Studies that look into algorithmic complexity (as graph size increases), as well more social issues like path length considerations and which averages to use for recommendations will become increasingly important.

6 Acknowledgements

This work was supported in part by grants from DARPA, the Air Force Research Laboratory, and the Navy Warfare Development Command. The Maryland Information and Network Dynamics Laboratory is supported by Industrial Affiliates including Fujitsu Laboratories of America, NTT, Lockheed Martin, and the Aerospace Corporation.

The applications described in this paper are available from the Maryland Information and Dynamics Laboratory, Semantic Web Agents Project (MIND SWAP) at <http://www.mindswap.org>.

References

1. Adamic, L., "The Small World Web". *Proceedings of ECDL*, pages 443-- 452, 1999.
2. Adding SVG Paths to Co-Depiction RDF, <http://Jibbering.com/svg/codepiction.html>
3. The Advogato Website: <http://www.advogato.org>
4. Albert, R., Jeong, H. AND Barabasi, A.-L. "Diameter of the world-wide web." *Nature* 401, 130–131, 1999
5. Bharat, K and M.R. Henzinger. "Improved algorithms for topic distillation in a hyperlinked environment," *Proc. ACM SIGIR*, 1998.
6. Brin, S and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Proc. 7th WWW Conf.*, 1998.
7. Broder, R Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. "Graph structure in the web. " *Proc. 9th International World Wide Web Conference*, 2000.
8. Carriere, J and R. Kazman, "WebQuery: Searching and visualizing the Web through connectivity," *Proc. 6th WWW Conf.*, 1997.
9. Chakrabarti, S, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan, "Automatic resource compilation by analyzing hyperlink structure and associated text," *Proc. 7th WWW Conf.*, 1998.
10. Dumbill, Ed, "XML Watch: Finding friends with XML and RDF." IBM Developer Works, <http://www-106.ibm.com/developerworks/xml/library/x-foaf.html>, June 2002.
11. FOAFNaut: <http://foafnaut.org/>
12. Gil, Yolanda and Varun Ratnakar, "Trusting Information Sources One Citizen at a Time," *Proceedings of the First International Semantic Web Conference (ISWC)*, Sardinia, Italy, June 2002.
13. Kleczkowski, A. and Grenfell, B. T. "Mean-fieldtype equations for spread of epidemics: The 'small-world' model." *Physica A* 274, 355–360, 1999.
14. Kleinberg, J, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, 1999.
15. Kumar, Ravi, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tomkins, and Eli Upfal. "The web as a graph". *Proceedings of the Nineteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, May 15-17, 2000.
16. Labalme, Fen, Kevin Burton, "Enhancing the Internet with Reputations: An Openprivacy Whitepaper," <http://www.openprivacy.org/papers/200103-white.html>, March 2001.
17. Levien, Raph and Alexander Aiken. "Attack resistant trust metrics for public key certification." *7th USENIX Security Symposium*, San Antonio, Texas, January 1998.
18. Milgram, S. "The small world problem." *Psychology Today* 2, 60–67, 1967.
19. Moore, C. and Newman, M. E. J. "Epidemics and percolation in small-world networks." *Physical Review E* 61, 5678–5682, 2000.

20. Newman, Mark, "The structure of scientific collaboration networks," *Proc. Natl. Acad. Sci. USA* 98, 404-409 (2001).
21. Newman, Mark, "Models of the small world", *J. Stat. Phys.* 101, 819-841 (2000).
22. Open Privacy Initiative: <http://www.openprivacy.org/>
23. Mutton, Paul and Jennifer Golbeck, "Visualization of Semantic Metadata and Ontologies," *Proceedings of Information Visualization 2003*, London, England, July 2003.
24. RDFWeb: FOAF: 'the friend of a friend vocabulary', <http://rdfweb.org/foaf/>
25. RDFWeb: Co-depiction Photo Meta Data: <http://rdfweb.org/2002/01/photo/>
26. Spertus, E, "ParaSite: Mining structural information on the Web," *Proc. 6th WWW Conf.*, 1997.
27. Szalay, A. S. 2001, "Astronomical Data Analysis Software and Systems X," in *ASP Conf. Ser.*, Vol. 238, eds. F. R. Harnden, Jr., F. A. Primini, & H. E. Payne (San Francisco: ASP), 3.
28. The Trust Ontology: <http://www.mindswap.org/~golbeck/web/trust.daml>
29. Watts, D. and S. H. Strogatz. "Collective Dynamics of Small-World' Networks", *Nature* 393:440-442 (1998).