Survey of RDF data on the Web

Technical Report

Andreas Eberhart
International University in Germany
eberhart@i-u.de

August 15, 2002

Abstract

The Resource Description Framework (RDF) allows describing metadata in an interoperable way. It is the basis for promising work in the area of the Semantic Web. This paper assesses how much and what kind of RDF data was found on the web in December 2001 and in August 2002. Four search strategies, crawling, scanning URLs obtained from an Internet Directory, a targeted search, and a search at URLs appearing in previously collected facts are described and compared. Results show that RDF is currently hard to find unless a considerable effort and resources are put into the search process.

1 Introduction

After the webs initial phase of being a medium for convenient reading and publishing static information, the popularity of web applications has grown enormously. Today, there hardly is a service or a good that is not available online. Nevertheless, almost all of these services are geared towards human interaction. The electronic data interchange (EDI) community had quite some success in standardizing message formats for application integration; it is however impossible to develop a lightweight standard that serves a variety of application domains. Therefore, EDI solutions are typically very specific to a certain industry.

The Semantic Web aims at bringing the web to its full potential by allowing software agents to communicate in a completely automated fashion. The core idea is to have ontologies provide information about the concepts of different domains. Agreeing on the ontology lays the foundation for integration. It provides a standard vocabulary that can be used by agents to assert facts and to state rules. Just like it is impossible to develop a universal set of API and message formats, it is also impossible to develop a universal ontology fitting everybodys needs. Following the spirit of the web, a set of mark-up languages,

RDF, RDFS, DAML, and RuleML, has been proposed that allows everybody to represent metadata, important concepts of a domain, as well as rules and constraints established by domain experts. The hope is that the ability to reuse and extend existing work as well as a natural selection of ontologies will eventually prove to be the driving force for the Semantic Web.

A whirlwind tour of RDF The Resource Description Framework (RDF) is a framework for metadata and the most basic mark-up language in the context of the Semantic Web. The core idea is that things are treated as URIs. A person could be denoted by her or his homepage. When talking about Ora Lassila, we might use the resource http://www.lassila.org; a desk in some office might be referenced via the companys inventory list as http://xyz.com/inventory#K4622-ERF. In the RDF terminology, these things are called resources. It is then possible to make statements about the resources. If Joe is Peters brother, we could state this as the following subject, predicate, object triple:

```
Subject:   http://www.mit.edu/~joe/
Predicate: http://www.cogsci.princeton.edu/~wn/concept#107127521
Object:    http://www.mit.edu/~peter/
```

Note that the "brother" predicate is a URL pointing to the Wordnet lexical database project at Princeton. The concept "brother", a male with the same parents as someone else, has the ID 107127521. Since further statements about Joe, Peter, and other resources can be made, we eventually end up with a directed labelled graph. The resources are the graph nodes and the statements define the edges.

The above RDF triple is somewhat clumsy, because we wanted to state the "is brother of" relation, nevertheless, due to the wide acceptance and popularity of Wordnet, we can be sure that our statement can be correctly interpreted by many agents. Consider the statement that Joe lives in Boston:

```
Subject:   http://www.mit.edu/~joe/
Predicate: http://www.schema.org/rdf/livesin
Object:    Boston
```

The first difference is that the predicate comes from another namespace. Secondly, the object is a simple string or literal, rather than another resource. If another statement also uses the string "Boston" as its object, it would be up to the application to decide, if the city of Boston, or maybe a project with codename Boston is meant. For further aspects of RDF / RDF Schema (http://www.w3.org/RDF) and information on DAML[1], RuleML[2], and the Semantic Web in general[3] can be found on these websites.

RDF is serialized using an XML syntax. Like any XML document, RDF might only be a stream of bytes travelling from one application to another via

[1] http://www.daml.org
[2] http://www.dfki.de/ruleml
[3] http://www.semanticweb.org

the network. RDF can also be stored in static files, separately or, if statements about an HTML website are being made, within the head tag of the website.

Motivation for the RDF survey Since RDF and the ability to state facts is the foundation of the semantic web, a survey of how much RDF data can be found is of interest. RDF is definitely used a lot by the Semantic Web research community; therefore, the survey is more an indication to what extent the public is starting to adopt the technologies developed. It is also of interest to evaluate typically used predicate namespaces in order to draw conclusions about the application areas.

Section 2 describes how the search was conducted and which tools were used. The results are then presented in section 3 before an evaluation and a summary are provided.

2 Collection of the Survey Data

Some initial experiments were conducted using the RDF crawler developed at the University of Karlsruhe[4]. Given a starting URL, RDF Crawler recursively traverses hyperlinks up to a specified search depth. RDF data found is stored in a file on the local system. The experiments quickly revealed that it is not easy to find RDF data on the web. If the starting points for the search are not selected carefully, a pure crawling approach might require an extensive amount of URLs to be processed before any RDF data is found. Therefore it was decided to pursue four different strategies, which are outlined in the following paragraphs.

The first experiment was performed in December of 2001. With the software and the search process in place, we reran the same experiment in August of 2002. The main intention for this was to see if any trend could be observed after the Semantic Web initiative got quite some public exposure lately. We plan to repeat the experiments in the future as well.

2.1 Crawling

According to a study by Lawrence and Giles in 1998 [5], even major search engines that continuously crawl the web only achieve coverage of at most 17% of the static Internet pages. Due to the massive growth of the Internet, this number is likely to have decreased even more [2]. With the limited bandwidth and computing resources available to our study, it would only be possible to cover small islands of URLs. Nevertheless, we applied this approach and chose popular sites within the RDF community as starting points. Table 1 shows these URLs that were used in both search runs. A total of 12507 pages within two hops of these URLs were processed in the first run. Two major RDF collections, namely the Open Directory Project structure and content dumps[5]

[4] http://ontobroker.semanticweb.org/rdfcrawl/
[5] http://dmoz.org/rdf.html

URL
http://www.w3.org/RDF/
http://wilbur-rdf.sourceforge.net/
http://www.daml.org/
http://www.lassila.org/
http://www-db.stanford.edu/~melnik/
http://www-db.stanford.edu/~melnik/rdf/api.html
http://www-db.stanford.edu/~stefan/
http://www-db.stanford.edu/
http://www.semanticweb.org/
http://protege.semanticweb.org/

Table 1: Popular sites within the RDF community were chosen as starting points for crawling

and the RDF version of the Wordnet lexical database project[6], were left out due to their large size. A site related to the RPM software packaging tool[7] also contains a large number of RDF files describing software distributions. These were only scanned in part. Since the choice of starting pages is very restrictive and quite arbitrary, we decided to include the Google directory page on RDF and its fifteen subordinate categories as well for the second run[8]. Similar to the www.semanticweb.org pages, these contain a very complete set of links to all sorts of RDF related sites. 31764 pages of this category were crawled during the second run.

2.2 Open Directory

In order to make sure that the breadth of the web is somewhat captured, we searched URLs from the Open Directory project[9]. This project organizes websites into categories, similar to Yahoo. The content, i.e. the URLs and their descriptions, are available in RDF format. During the first run, 527408 URLs were extracted. Due to the massive growth of the Open Directory project, within eight months, this number increased to 2912434. This comprises all categories, except for the adult pages. The URLs obtained from this source are typically entry- or homepages. Due to the large number of URLs, no more crawling was done from these sites. Obviously this approach will not find standalone RDF data, residing is a separate file. However, we expect to find information about the website encoded in the HTML page itself, as demonstrated by the following

[6] http://www.semanticweb.org/library/

[7] http://rpmfind.net/linux/RDF/

[8] The Google RDF directory can be found http://directory.google.com under Reference > Libraries > Library and Information Science > Technical Services > Cataloguing > Metadata > Resource Description Framework. Note that the Google directory bases on the Open Directory.

[9] http://dmoz.org

example:

```
<head>
  ...
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
           xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about=""
    dc:title="Ora Lassila"
    dc:description="Ora Lassila's professional home page"
  </rdf:RDF>
</head>
```

2.3 Targeted search

Since the initial experiments indicate that RDF data is hard to find, a more targeted search was conducted. The popular Google engine allows searching for pages with a certain string in the URL. Obviously a URL containing RDF is more likely to contain some RDF data. A small parser was developed that extracts the URLs out of a Google HTML result page. During the second run we used the newly available Web Service interface that allows Google to be queries programmatically from a Java or .NET client[10].. Leveraging the extensive Google database, a total of 1256 URLs were obtained during the first run. Table 2 summarizes the number of URLs in each of the three categories. There is a small overlap between the categories. Three URLs where RDF data was found appear in both the RDF community and the Open Directory categories, 63 URLs appear in the RDF community as well as the Google targeted search categories. For the second experiment, less pages containing RDF in the URL could be retrieved and the number decreased to 1079. Note that is by no means an indication that less information on RDF could be found. The most likely scenario is an internal change in the Google database. The search result site actually claims to have found 2,410,000 pages, however only the number specified could be obtained, both via the browser and the Web Service interface.

2.4 URLs found in the fact triples

Since RDF subjects, predicates, and most objects are URLs themselves, we assume to find RDF data at those URLs. Facts gained from the other categories were extracted first. We only considered URLs that have not appeared in any other category. We chose to implement this restriction due to the expected large overlap with the other categories. After 124374 facts have been found in the first run, a first search process was started, yielding 365 new URLs. Note that URL anchors, i.e. the "#" character in the URL must be ignored, since anchors only identify a certain position within the same document. Therefore it is not necessary to scan such a URL again and only the part left of this sign was considered. The facts of those URLs were loaded again and the process

[10]See http://www.google.com/apis/ for details

	Number of URLs scanned	
Category	**Dec 2001**	**Aug 2002**
RDF Community	12507	31764
URLs from Open Directory	527408	2912434
RDF appears in the URL	1256	1079
URLs from facts	365	6733

Table 2: URLs per category

was repeated in the hope that one can follow the edges of the RDF graph to find new data. The 1923 new facts from the 365 new URLs yielded only 23 new websites and the process was stopped at this point.

This changed in the second run. 139288 facts were found at URLs from the other categories. The subjects, predicates, and resource objects form those facts pointed to 6037 previously unseen URLs. We loaded 54227 new facts from those URLs. This number is promisingly high, however, it turned out that almost all of the facts came from large data repositories that organized their data not within one large file accessible at a single URL, but rather made that data available via several URLs. One example is http://xmlns.com, where an RDF representation of the Wordnet database is hosted. The URL http://xmlns.com/wordnet/1.6/Survivor, for instance, contains several statements about other Wordnet resources located at similar URLs. Nevertheless, we were able to extract 697 new URLs from those new facts. At this point, hardly any URLs could be identified from facts from those sites that were not a simple derivation of previously seen URLs and the process was stopped.

2.5 Architecture of the RDF database

In order to be able to perform further analysis of the data, we decided to load the facts into a relational database system. Figure 1 shows the table layout. The facts table stores subject, predicate, and object triples, along with the id of the URL they were found in. The primary key selection makes sure that data cannot be inserted twice from the same source in case the upload program needs to be run repeatedly. The URLs table has a further uniqueness constraint on the URL attribute to avoid duplicate URLs in the data set. Finally, the URLtype table indicates which of the three categories mentioned above the URL belongs to. The msg field in the URLs table records any error such as network errors, XML parsing errors, etc., that might occur while the data is accessed.

Figure 2 illustrates the overall software design. Any URL to be scanned is first inserted into the URLs table in the database. For each category, a different approach was used: Data from the Open Directory dump was extracted with an XSLT style sheet. The program GetGoogleURLs runs a query against Google with the query parameters encoded in the URL. The encoding mechanism, for instance how to browse through a large result set page by page, was obtained

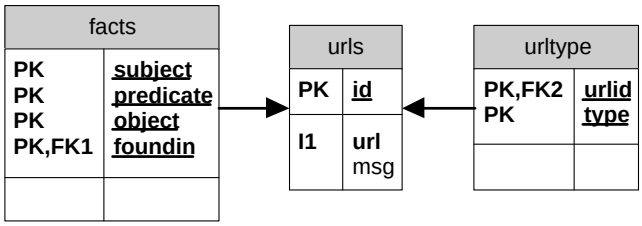

Figure 1: Design of the RDF database

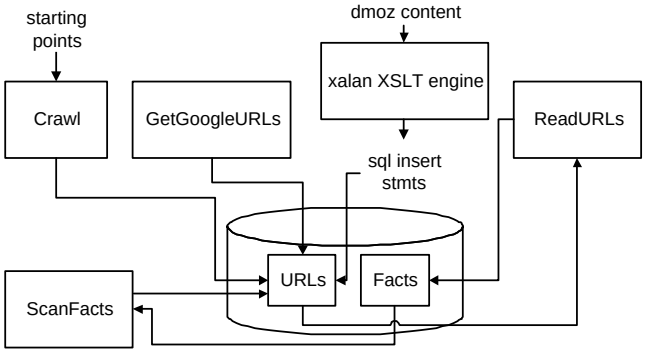

Figure 2: Overall software design

from the advanced search web form. Note that running the automated query repeatedly might cause the requesting IP to be banned by Google. The Crawler program can use multiple threads to spider the hyperlink structure from several starting points.

After the URLs table is filled, the RDFLoad program can be started to scan the URLs for RDF data. It uses Sergey Melnik's RDF API[11] to upload the facts to the database. Any Java exception that is caught is written into the msg field in order to trace, for instance how many pages contain syntactically incorrect RDF, or how many pages could not be reached because of a network outage. Since this is the most widely used RDF API, a URL is considered to contain correct RDF if the RDF API in the version of Jan 19th 2001 parses it without error message and if the resulting RDF triple set is not empty. It is considered to contain incorrect RDF, if an "org.xml.sax.SAXParseException" is thrown, and it is considered to not contain any RDF if a "java.io.EOFException: no more input" exception occurs or if the resulting RDF triple set is empty. Any RDF data found is also written to a file num.rdf for further examination, with num being the id of the URL scanned. Finally, ScanFacts inserts URLs found in

[11] http://www-db.stanford.edu/∼melnik/rdf/api.html

the facts table into the URLs table. Note that ScanFacts can only be run after ReadURLs inserted some facts. Furthermore, ReadURLs must be run again to load facts found in the newly inserted URLs.

The major advantage of this database-centric architecture is that the search process can be stopped and resumed without any problem. The database provides the necessary persistence and constraints to make sure that data cannot be inserted twice. Since this survey aims at evaluating a snapshot of the current use and acceptance level of RDF, the database currently only accumulates data and no mechanism for deleting or updating the information is implemented.

The application as well as the data sets can be downloaded at http://www.i-u.de/schools/eberhart/rdf/.

3 Search Results

This section outlines the results of the search conducted over 541536 web sites in the first, and 2952010 web sites in the second search.

3.1 How many pages contain RDF data?

Figures 3 and 4 show how many pages contained RDF data by outlining the percentages of the following cases: a general error such as file not found occurred (cyan), page available but no RDF data found (yellow), syntactically incorrect (red), and correct (blue) RDF found. As expected, we see that there are strong variations between the categories. During the first experiment, RDF data was found in only sixteen out of over half a million pages from the Open Directory. This number increased to 180 out of 2.9 million pages in the second run. The density around semantic web portals is higher but still disappointing. About one percent of the URLs that appear in other facts were found to contain RDF in both runs[12]. Finally, the highest success rate was found in pages with "rdf" in the URL, especially pages ending with ".rdf". These URLs contain RDF data with a probability of 17% and 10% in the first and second experiments respectively. Similar percentages occurred in the last URL category, where we followed RDF arcs that were present in the facts found in other categories. We found RDF in 9% and 13% of the pages during the first and second run. Overall, with the categories combined, this translates to 1018 out of 541536 URLs containing RDF, 613 of them with correct and 405 with incorrect RDF for the first run. In the second run, out of 2952010 pages, 1479 contained valid and 2940 contained invalid RDF. Please note that the overall numbers are largely dominated by the Open Directory category, where the bulk of pages were scanned.

[12] Originally we also counted documents on which the RDF parser used yielded an empty RDF result, i.e. a set of zero RDF statements. The exclusion of these pages explains the higher number stated in [4].

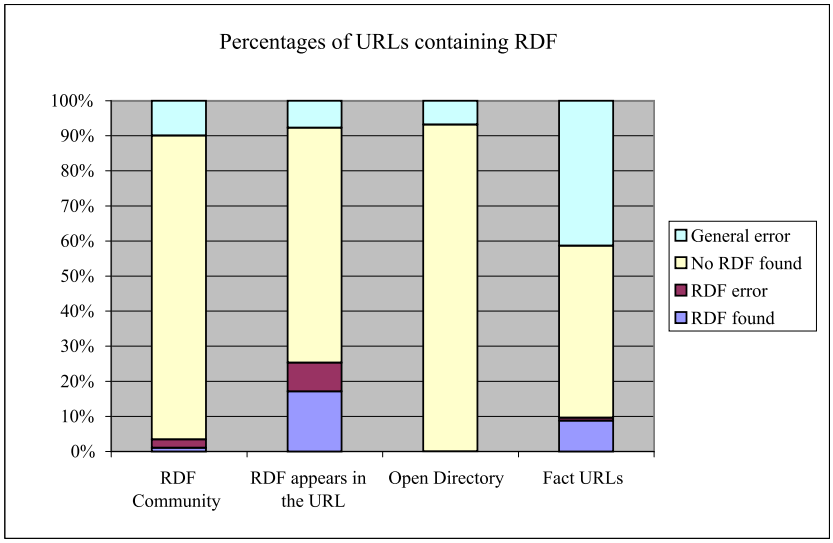

Figure 3: RDF data found per category during the first search Dec. 2001

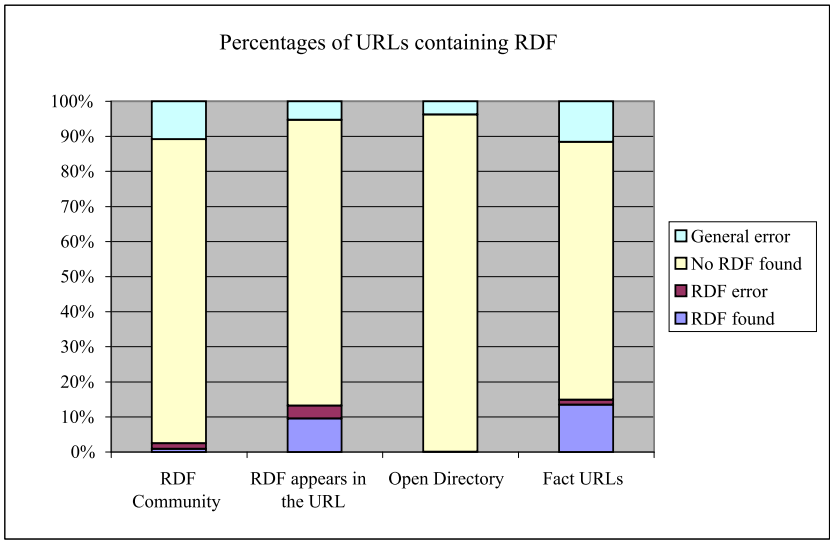

Figure 4: RDF data found per category during the second search Aug. 2002

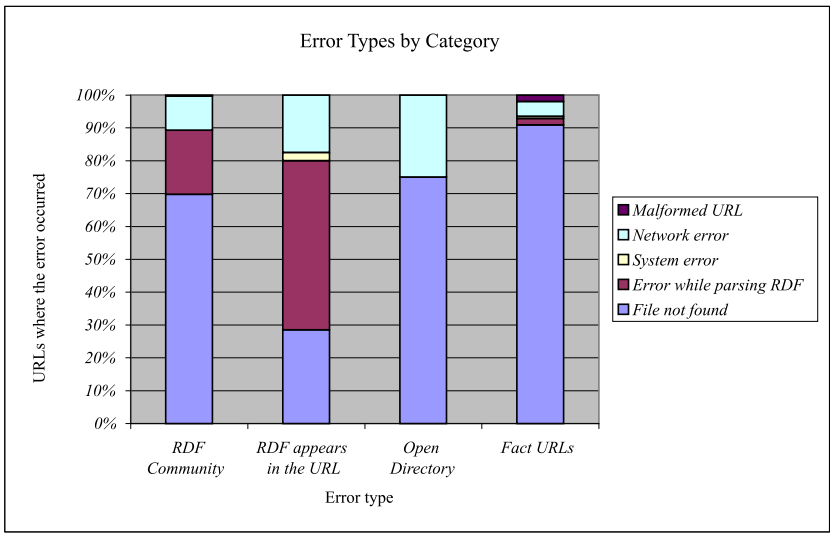

Figure 5: Error types during the first search Dec. 2001

3.2 Error causes

Figures 5 and 6 provide more detailed information about the error causes, i.e. the cyan and red sections from figures 3 and 4. Network errors and URLs that no longer point to any page were expected to be the most frequent sources of errors. In the first experiment, system errors are caused by the ReadURLs component and play a small role for the URLs gained from the targeted search. Five of these failures were recorded, four null pointer exceptions with unknown causes and one out of memory error, caused by a large binary file. During the second run we increased the number of threads used to search especially the large number of Open Directory pages. This resulted in a substantially higher number of 1147 out of memory errors in this category. Considering the total number of 2.9 million URLs scanned, this is definitely not a big concern for the quality of the results.

An interesting issue are the 405 and 2940 RDF parsing errors that might reveal potential problems caused by the use of older RDF versions or frequently occurring mistakes made by RDF authors. In both experiments, over half of these errors are caused by unresolved entity declarations such as the non-breaking space entity   defined in HTML. The remaining error causes are partly XML-related such as missing attribute quotes and partly RDF-related such as nested descriptions. We found that 14 and 24 URLs omitted the RDF namespace and simply placed <RDF> tags in the document. This manifests itself in a "unresolved namespace prefix" error.

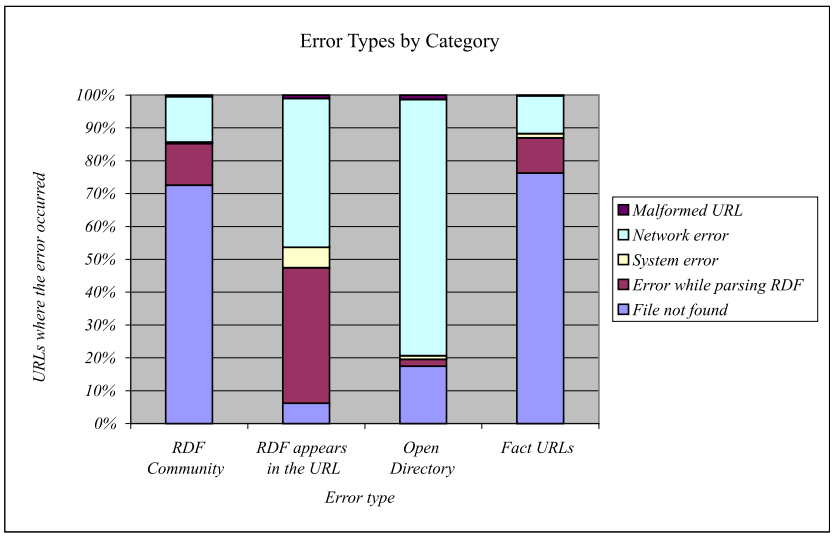

Figure 6: Error types during the second search Aug. 2002

3.3 Size of the RDF data sets

During the first search, a total of 125072 facts were extracted, 104580 came from the targeted search category, 19696 from the RDF community category, 1923 from facts URLs, and only 98 from Open Directory web sites. Figures 7 and 8 illustrate how much data was found at the different URLs. When analyzing the second run with respect to the category's contribution to the total number of 254783 facts, it can be seen that with 115495 facts, the last category contributed more than the RDF community pages with 107308. 29168 facts come from the targeted search and 2812 from Open Directory pages. The first run yielded only three large files with more than 10000 facts were found, namely a list of airports from http://www.megginson.com, an excerpt from the CIA world fact book at http://www.ontoknowledge.org, and a category description file at http://w.moreover.com. The second search tapped into five large repositories, namely the OpenCyc project at http://opencyc.sourceforge.net, part of the WordNet database at http://www.semanticweb.org/library/wordnet/wordnet_hyponyms-20010201.rdf, two military ontologies at http://orlando.drc.com/, and again http://w.moreover.com. Overall, two changes can be observed between the experiments. First and foremost, the last category happened to tap into two highly connected datasets, the xmlns.com version of Wordnet, which is split into many files and the moreover.com directory. This resulted in the large overall increase. The remaining categories actually show little change except for the large number of URLs containing medium sized datasets in the Open Directory category.

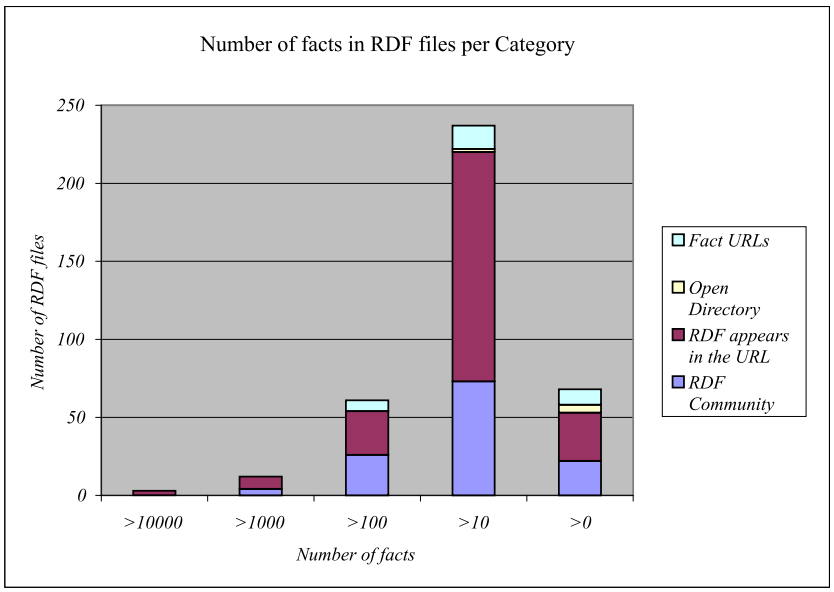

Figure 7: Distribution of the RDF data set sizes during the first search Dec. 2001

A fairly large number of sites contain data in the rich site summary (RSS) format version 0.9. RSS is a format that originally has been proposed by Netscape as a lightweight syndication format for distributing news headlines on the web, for example via Netcenter channels. An RSS example is shown in the following block:

```
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
         xmlns="http://my.netscape.com/rdf/simple/0.9/">
<rss>
    <channel>
        <title>BBspot</title>
        <link>http://www.bbspot.com</link>
        <description>Your Spot for Tech Humor</description>
    </channel>
    ...
```

3.4 Typical namespaces used

After the probability of finding RDF data and the typical data set sizes have been evaluated, we examine the facts further. In order to be able to correctly interpret data, it is crucial that an agent understands or can correctly interpret the predicate used in the triple. One of the most prominent and frequently

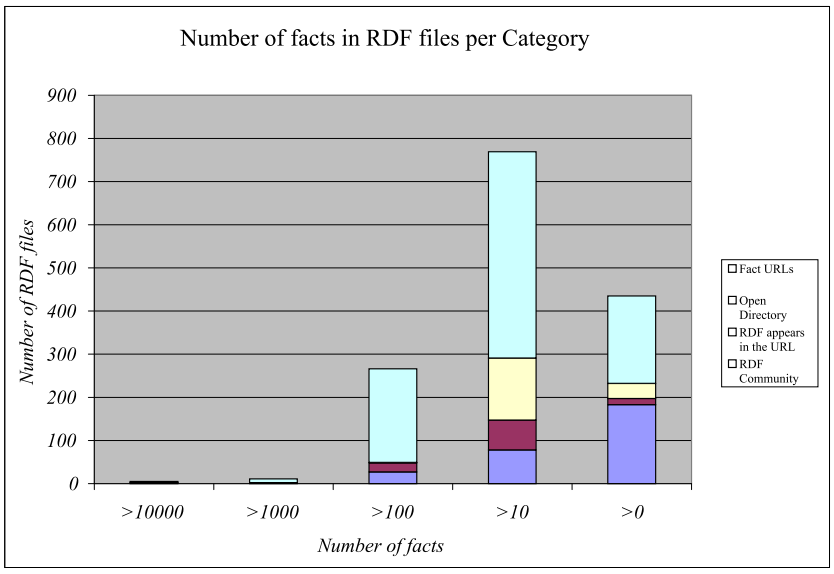

Figure 8: Distribution of the RDF data set sizes during the second search Aug. 2002

cited examples is the Dublin Core metadata vocabulary. Tables 3 and 4 show how often a certain namespace prefix occurs in the facts gathered, along with information at how many distinct URLs this namespace prefix was found.

In the data of the first experiment, we can see that some large data sets like the Ontoknowledge case study and David Megginson's airport example rank among the top namespaces but are only used by one web site. The largest number of other sites references the W3C and the Dublin Core namespace, with the RDF type relationship occurring very often. We really should have counted the distinct hosts rather than distinct URLs, since a large data set being split into several files, like in the rpmfind.net example, would create the wrong impression. This was not done since the data model does not support this specific query. A manual check confirmed though, that the URLs really are located on a large number of different hosts. No Wordnet or Open Directory predicates were found.

The second experiment shows a similar picture. Again, we can see some of the large datasets that contribute many facts but occur only within a very limited number of documents. Dublin Core remains the most frequently used non-W3C namespace, however, the Adobe namespace is a prominent newcomer in this list of namespaces found at several different sources. These pages follow the Adobe eXtensible Metadata Platform (XMP) [1]. XMP builds on top of RDF and is designed to embed metadata into application files. The fact that a major IT company embraces RDF is obviously a very encouraging sign for the

Predicate namespace prefix	in # of URLs	in # of facts
http://www.ontoknowledge.org/oil/case-studies	1	23259
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	326	21011
http://www.w3.org/1999/02/22-rdf-syntax-ns#	326	17298
http://www.megginson.com/exp/ns/airports#	2	13589
http://alchemy.openjava.org/ocs/directory#	1	7014
http://www.w3.org/2000/01/rdf-schema#	62	6182
http://purl.org/	123	5198
http://interdataworking.com/vocabulary/	27	4698
http://www.trustix.net/schema/rdf/spi-0.0.1#	2	3012
http://my.netscape.com/rdf/simple/	93	2446
Other http://www.w3.org	331	2212
http://www.daml.org	27	2032
http://www.rpm.org	7	1716
http://metainfo.hauN.org	1	1351
http://home.netscape.com/	1	801
Other	164	13253

Table 3: Predicate namespace prefixes used by the RDF data found during the first search Dec. 2001

Semantic Web community.

Like the predicates, commonly referenced objects are also important for agents to understand RDF facts. An example would be metadata about a web site referring to an Open Directory category. This would allow any agent aware of the Open Directory to draw conclusions about the content of the site for example. Table 5 shows the results of this test. In both experiments we found about 57% of the objects to be literals, mostly numbers and the frequently occurring strings "en", "text/plain". As the large number of RDF type predicates suggests, the objects are mostly RDFS classes. We could not find any non-class object that is referenced frequently from many different sites. Hardly any references to prominent repositories such as Wordnet or Open Directory objects were found.

3.5 Comparison of the two Experiments

Before we will give an evaluation, we want to analyze if any trend can be observed when the runs from December of 2001 and August of 2002 are compared. Overall, the results do not show any drastic changes except for the much larger number of URLs and facts found in the last category, which comprises the sites referenced by the other facts found. This hints at a higher lever of interconnectivity among the RDF facts. However, a closer analysis shows that most of these come from a small set of sources. During the first run, URLs from 152

Predicate namespace prefix	in # of URLs	in # of facts
http://www.cogsci.princeton.edu/	1	78445
http://www.w3.org/2000/01/rdf-schema	693	57132
http://www.w3.org/1999/02/22-rdf-syntax-ns#type	1205	37926
http://orlando.drc.com/	19	27773
Other http://www.w3.org	435	11454
http://alchemy.openjava.org/	2	9793
http://purl.org/	463	9411
http://interdataworking.com/	16	5247
http://www.daml.org/	53	4490
http://ilrt.org/	9	2124
http://opencyc.sourceforge.net/	1	1630
http://ns.adobe.com/	152	1589
http://my.netscape.com/	34	902
http://www.rpm.org/	3	734
http://www.ontoknowledge.org/	2	645
http://dublincore.org/	82	544
http://www.omg.org/	3	523
http://www.semanticweb.org/	41	466
http://annotation.semanticweb.org/	5	375
http://xmlns.com/	48	351
http://example.org/	95	121
http://www.nesstar.org/	6	106
Other	129	3002

Table 4: Predicate namespace prefixes used by the RDF data found during the second search Aug. 2002

RDF Object	in number of Facts Dec. 2001	in number of facts Aug. 2002
Other literals	58949	237163
Other resources	44562	175110
http://www.w3.org/1999/ 02/22-rdf-syntax-ns#	7646	7947
Numbers	8115	9667
en	2414	3278
hourly	2361	3265
text/plain	1002	1410

Table 5: Overview over RDF Objects

distinct hosts were added. This is within the same order of magnitude than the 269 hosts obtained during the second experiment.

Overall more pages were scanned, largely due to the big increase in the size of the Open Directory, and naturally also more pages containing RDF surfaced, although the percentage of pages containing RDF actually declined. The changes are not too big so we advise against trying to draw any conclusion from this.

4 Evaluation

The results of this survey suggest that RDF has not caught on with a large user community. Obviously the search was not very extensive. Therefore it is possible, that some large RDF islands were not found. Much RDF data might also not be publicly available on the web. In a way we are seeing a situation that is similar to the adoption of web services. There are millions of data sources that could easily be made available via both web service and RDF interfaces on a technical level. Without a doubt, web services have covered more ground in terms of public acceptance, however, except for some highly visible services like the Google web service API or Microsoft's Map Point service[13], most of the services that can be found in the UDDI registries today have a clear test prototype character. The most likely explanation for this situation is that the automation of the web be it through web services or the Semantic Web, brings about a radical change from a business perspective. Advertising largely finances today's web with its free offerings. This must change when machines and no longer humans access the sites. Several payment methods such as micro payment and bulk subscriptions are being considered, however, it is too early to see clear trends or even standards in this direction. Once this shift towards a more automated web begins, we will probably also see more data being exposed in RDF format.

The use of RDF as a simple metadata format for HTML pages does not make much sense at the moment, since HTML meta tags do the job just fine. The strengths of RDF, namely its extensibility and the possibility to refer to widely accepted standard vocabularies and global identifiers, are not being used. The very poor search results among regular web sites taken from the Open Directory clearly supports this observation. Furthermore, the nature of facts found indicates that the level of interconnection is quite low, i.e. most objects are literals or belong to RDF schemas. Apart from the Dublin Core and the Adobe XMP namespaces, hardly any other non-W3C vocabulary is used by many different authors. Specifically Adobe's support of RDF is a very promising sign, however.

Nevertheless, we believe that RDF has a lot of potential. The popularity of the NEC CiteSeer [3] research index for example, is a clear indication that there is a need for metadata and better, more targeted search on the web. This application extracts the information which other papers are cited by a certain publication. The number of citations is used as an indication for the quality

[13] http://www.microsoft.com/mappoint/net/

of a publication. If RDF metadata had been used to describe publications, a system like this would be quite easy to implement. One can only imagine the various RDF applications that could be implemented.

We believe that it is crucial for the success of the Semantic Web that the research community starts working on some of these applications in order to get a large user community excited about the ideas and possibilities. Only then will it be possible to resolve what seems to be a chicken and egg problem: data will only be marked up if there is an application, and an application only is successful if it operates on a large data set.

Possible sources of errors Apart from the limited search that might not have revealed large amounts of RDF data, another source of error is not recognizing RDF data when a page is scanned. We tested several cases of incorrectly formatted RDF and missing RDF namespace definition. It turned out that the RDF API we used for the search reacts in a very robust manner by indicating the problem with an error message. If an XML or XHTML document is scanned by the RDF API, an empty dataset is returned. Some random samples of these cases were examined manually and no malfunction of the RDF API could be detected there, i.e. RDF data that was omitted by the RDF API.

References

[1] Adobe Inc. A managers introduction to adobe extensible metadata platform, the Adobe XML metadata framework. http://www.adobe.com/products/xmp/pdfs/whitepaper.pdf.

[2] M. Bergman. The deep web: Surfacing hidden value, 2001.

[3] K. Bollacker, S. Lawrence, and C. L. Giles. CiteSeer: An autonomous web agent for automatic retrieval and identification of interesting publications. In K. P. Sycara and M. Wooldridge, editors, *Proceedings of the Second International Conference on Autonomous Agents*, pages 116–123, New York, 1998. ACM Press.

[4] A. Eberhart. Survey of RDF data on the web. In *Proc. of the 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*, July 2002.

[5] S. Lawrence and C. L. Giles. Searching the World Wide Web. *Science*, 280(5360):98–100, 1998.