SEARCHING ON THE OPEN SEMANTIC WEB USING A URI IDENTITY MANAGEMENT APPROACH

By
Afraz Jaffri

A mini-thesis submitted for the transfer from MPhil to PhD

School of Electronics and Computer Science,
University of Southampton,
United Kingdom.

August, 2007

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING

SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

SEARCHING ON THE OPEN SEMANTIC WEB USING A URI IDENTITY MANAGEMENT APPROACH

by Afraz Jaffri

The Semantic Web has a vision that involves the production and use of large amounts of RDF data. There have been recent initiatives amongst the Semantic Web community, in particular the Linking Open Data activity and the ReSIST project, to publish large amounts of RDF that are both interlinked and dereferenceable. The proliferation of such data gives rise to millions of URIs for non-information resources such as people, places and abstract things. Frequently, different data providers will mint different URIs for the same resource, leading to the problem of coreference. This thesis describes the phenomenon of coreference, where it occurs in other disciplines and how it is relevant to the Semantic Web. A 'Consistent Reference Service' is proposed for URI identity management and a description of how this is being used in the infrastructure of a scalable Semantic Web system is demonstrated.

One of the features that made the Web so easy to use is the ability to search web pages in a matter of seconds through the use of search engines. The URI management system described in this thesis is being integrated into the open Semantic Web. The CRS and RDF data available on the Web means that the possibility appears to improve the quality of searching by linking the Semantic Web with the 'ordinary' Web. This thesis outlines an architecture for using the Semantic Web to assist and improve searching on the document Web.

Contents

List of Figures

Acknowledgements

This work is partly supported under the ReSIST Network of Excellence, which is sponsored by the Information Society Technology (IST) priority in the EU Sixth Framework Programme (FP6) under contract number IST 4 026764 NOE.

I would also like to thank my supervisor Hugh Glaser for his supervision during the PhD.

Definitions and Abbreviations Used

ACM	Association for Computing Machinery
CRS	Consistent Reference Service
DBLP	Digital Bibliography and Library Project
ECS	School of Electronics and Computer Science
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronic Engineers
OWL	Web Ontology Language
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
ReSIST	Resilience for Survivability in Information Society Technology
RKB	Resilience Knowledge Base
SIOC	Semantically Interlinked Online Communities
SKOS	Simple Knowledge Organisation System
SPARQL	SPARQL Protocol and RDF Query Language
SUMO	Suggested Upper Merged Ontology
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
Web	The World Wide Web
W3C	World Wide Web Consortium

Chapter 1 Introduction

1.1 Overview of Research

Research into the Semantic Web has been continuing for almost a decade. The early days of research concentrated more on the theoretical aspects of ontologies, description logics and how to formally represent knowledge on the Web. Recently, research has become more concentrated on practical applications that can use the vast amount of knowledge that has been published as RDF or OWL. The research presented in this thesis focuses on the latest developments in the Semantic Web community: Linking Open Data and Information Retrieval. This new research area is still in its infancy and the system presented in this thesis attempts to advance the state of the art in large scale knowledge management of open datasets on the Semantic Web.

1.2 Contributions

This thesis documents contributions made to the field of URI management, Coreference and semantic searching of Linked Open Data.

1.3 Document Structure

This thesis describes the work undertaken in order to provide a URI management service on the Semantic Web. The work is motivated by first looking at the problems of searching on the document Web and how existing Semantic Web search engines provide an improved level of functionality for information retrieval. Several semantic search engines are examined and their shortcomings are highlighted. The thesis argues that in order to perform knowledge retrieval on the open Semantic Web, a URI coreference or consistent reference service needs to be implemented. The theory of

coreference is explained along with why it is a problem in the Semantic Web. The last part of the thesis details how the coreference system will be integrated on the open Semantic Web to give improved search functionality and consistency.

Chapter 2 presents an overview of the Semantic Web fundamentals including the basic theory and explanation of the knowledge representation languages RDF and OWL. Two projects that use these formats for knowledge representation are also discussed.

Chapter 3 describes the field of information retrieval or searching. The problems with existing document based searches are given along with the motivation for providing new semantic search engines to help give better search results. Several semantic search engines are described, and systems that represent the state of the art in searching on the open Semantic Web are analysed.

Chapter 4 introduces the concept of coreference and how it is dealt with in the fields of information science and databases. The need to manage coreference in the Semantic Web is argued using example URIs that are currently being used by a number of Semantic Web agents.

Chapter 5 presents the Consistent Reference Service for the management and integration of URIs. The system architecture is detailed and a description of an application that is using the service is also presented.

Chapter 6 outlines the future work that will take place in order to fully integrate CRSes on the Semantic Web. The additional components needed to build the system are described and a prototype application demonstrating the underlying principle of CRS integration is put forward.

1.4 Declaration

This thesis describes the research undertaken by the author while working within a collaborative research environment. This report documents the original work of the author except in Section 5.3.

Chapter 2 Semantic Web Fundamentals

This section provides a general introduction to the Semantic Web and its associated technologies. The main components of the Semantic Web including RDF, OWL and URIs are all discussed. An introduction is given to the Linking Open Data and ReSIST projects that are contributing to the uptake of the Semantic Web by publishing large amounts of RDF data.

2.1 The Semantic Web

The World Wide Web as it exists today consists of many millions of documents accessible through The Internet. Documents can consist of text, audio, video or indeed can be of any format that can be read on a computer. These documents are mainly produced for human consumption, i.e. interaction by a user or simply text that can be read. Therefore the majority of the documents on the Web are meant to be understood by humans and used by humans. Although the Web has provided a useful and sometimes invaluable resource, there is a potential to make the Web even more useful than it is today. A new level of functionality could be added to the Web if, instead of being human oriented, the web or documents on the web could be understood by machines as well as humans. The need to make The Web machine understandable sparked the beginning of The Semantic Web.

The Semantic Web is an effort, to effectively organise all of the knowledge on the Web; make it machine understandable using a common set of standards; make it

universally available to different forms of devices; and provide services that will help us achieve tasks that, at the moment, require some amount of time and effort.

To give us an idea of what The Semantic Web would look like a futuristic scenario was presented in which electronic intelligent agents communicate with each other using the Semantic Web to automatically create doctors appointments and buy medication on prescription (Berners-Lee 2001). There are many such scenarios in which The Semantic Web could fully automate our everyday tasks. When somebody would like to go on holiday abroad, The Semantic Web would enable the flights, travel, accommodation and entertainment to be booked automatically. When someone would need to replace a part of their computer, they would only need to describe that component and then all of the manufacturers and sellers would automatically be contacted and the best priced component would be ordered. This is obviously a very challenging goal and requires a lot of effort and research in order to make it a reality. The approach so far has been to build each part of the 'cake' (see next section) piece by piece until eventually all the different areas of research converge. During the last few years significant progress has been made in laying the foundation of The Semantic Web which has now culminated in W3C approved specifications for knowledge representation (RDF) and ontology definition (OWL). With these specifications in place applications are now being developed that demonstrate the potential power of The Semantic Web (Shadbolt et al. 2004), (Karger et al. 2005), (Noy et al. 2000).

2.2 The Semantic Web Backbone

2.2.1 RDF/RDFS

The Semantic Web is based around a core set of standards that have been developed by the W3C (World Wide Web Consortium). This is commonly described as the 'Semantic Web Layer Cake', (Berners-Lee, T., 2007) and is shown in Figure 2.1.

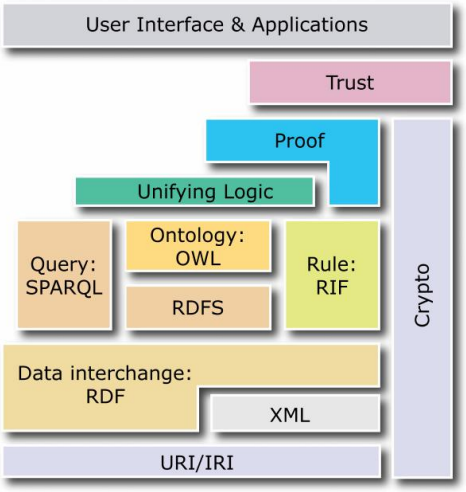

Fig.2.1. The Semantic Web Layer Cake

The most important language shown in figure 2.1 is RDF and its associated vocabulary description framework RDF Schema. The Resource Description Framework is a standard that 'provides interoperability between applications that exchange machine understandable information on the Web' (Lassila & Swick 2002). The specification enables things in the real world to be described in the same way that humans use sentences to describe things.

The basic description takes the form:

```
<subject> <predicate> <object>
```

A subject takes the form of a URI (Uniform Resource Identifier), which is a way of identifying any object in the world. It looks the same as a URL but the address which is given may not be accessible or may not even exist on the Web, instead it is used as a way of linking objects to some definite identifier. The predicate part, which is also a URI, describes the property of the subject. The object is the thing that is 'doing' the predicate. For example the sentence, 'The University of Southampton has a student called Afraz Jaffri', contains a subject, 'The University of Southampton', a predicate, 'student' and an object, 'Afraz Jaffri'. Such a sentence is easy to translate into basic RDF/XML syntax and is expressed as:

```
<rdf:RDF>
  <rdf:Description about="http://www.ecs.soton.ac.uk”>
    <akt:has-student>Afraz Jaffri</akt:has-student>
  </rdf:Description>
</rdf:RDF>
```

This assumes that the entity The University of Southampton is represented by the URI, "http://www.ecs.soton.ac.uk". The '*akt:*' refers to a namespace prefix that is declared in a namespace declaration at the beginning of the RDF document. The declaration contains the identifier where the schema to describe the property 'has-student' can be found. RDF also contains many other features such as containers and ways to make statements about statements.

The RDF specification can express a number of different properties of entities and relationships between those entities. The RDFS specification builds on RDF to allow the creation of classes and subclasses to be used in much the same way as they are used in taxonomic classifications. Together, RDF and RDFS have become the accepted way of describing knowledge of specific domains and information about real-world entities.

2.2.2 Ontologies and OWL

In order for a machine to be able to understand objects in the real world, there needs to be a way to represent and classify these objects. We define and associate objects in the real world by the knowledge that we have about them. An ontology is a means of being able to store knowledge about a particular subject area or for multiple subjects within some specific domain, and process it in some way. Formally they are described by Gruber (1993) as being 'a formal, explicit specification of a shared conceptualisation.' Ontologies promise a 'shared and common understanding of some domain that can be communicated between people and application systems' (Fensel, 2001).

These descriptions are rather vague and there is no way of showing how an ontology can be represented in a form that a computer can read and process. Previous research has led to the development of languages to represent ontologies such as description logics, Frame Logic (Kifer et al. 1995) and Knowledge Interchange Format (KIF) (Genesereth et al. 1992). There have also been full scale applications made from these languages that are in use today (Fensel, 2001).

The main use of an ontology is to describe a domain using an appropriate vocabulary and define relationships between the terms taken from the vocabulary. One of the most natural ways for humans to classify objects is to use hierarchies. The RDFS specification reflects this by using classes and subclasses and has the ability to define properties that can be given to an instance of a class. However, in order to establish more complex relations, such as cardinality constraints, optional constraints or disjointness, a more complex syntax is required. This has led to the development of OWL (Web Ontology Language) which has now become the standard for ontology definition. This language defines syntax for describing classes and properties as well as more complex relationships. The OWL language 'is designed for use by applications that need to process the content of information instead of just presenting information to humans' (McGuiness & Harmelen 2004). There are three types or 'flavours' of OWL: OWL Lite, OWL DL and OWL Full. OWL Lite is a subset of OWL DL and OWL DL is a subset of OWL Full. The main difference between OWL full and the other two flavours of OWL is that in OWL Full instances of a class can be classes, where as in OWL Lite and OWL DL, instances of a class must be individuals. In practice, this means that working with OWL Full is generally too complex for a logic reasoner to use for logical deduction, but OWL DL is both complete and decidable and is therefore easier to reason over and use.

2.3 URIs, URLs and Semantic Web Architecture

The base of the layer cake shown in figure 2.1 is URIs and IRIs. Uniform Resource Identifiers and their corresponding Internationalised Resource Identifiers are central to the design and use of the Semantic Web. A URI provides a common syntax that can be used to name or identify any entity in the world. For example, the URI "http://id.ecs.soton.ac.uk/person/6751" is an identifier used to refer to the person 'Afraz Jaffri' that has been provided by ECS. This URI uses the http naming scheme although other schemes are available such as urn, dav, file and ftp.

The universality of the URI is a fundamental part of Web architecture. The idea of using URIs is that they should be something into which any system's identifiers can be mapped. Providing a name for different entities and resources means that descriptions of objects can be easily made and published so that other agents or systems can combine knowledge from different sources. This relies on the assumption that

every object will have a unique URI. However, anyone who wants to make statements or give knowledge about an object can introduce their own URI to refer to that object. Thus, there maybe many URIs that refer to the same real-world entity. This leads to the problem of coreference that is described in more detail in Chapter 4.

The relationship between a URI and a URL is a subtle one. In the early days of the Web it was thought that the URL was a subclass or special type of URI (W3C, 2001). The contemporary view is that a URL is only an informal concept that provides the address of a location to access a resource. Therefore it can be said that a URI that uses the 'http' scheme is also a URL.

2.3.1 Information and Non-Information Resources

The current Semantic Web architecture divides entities and objects that have URIs into two categories: information resources and non-information resources. Non-Information resources are those entities that cannot be represented as a byte stream or serialised into a character format. This category of objects covers things such as cars, people, places, books and concepts that exist in the real world and not on the Web. The category of information resources on the other hand, consist of resources that can be accessed on the Web such as HTML pages, JPEG images, PDF documents and RDF descriptions. Information resources can be used to describe non-information resources in a way that can be accessed using HTTP. The two categories of resource have been made in order to be able to represent real-world entities on the Web. The URI of an object cannot also contain the description of that object.

In order to use URIs and make statements about the objects that the URIs refer to, a mechanism had to be constructed so that the URI and description stay separated. The way that this is accomplished is to introduce an HTTP 303 redirection when the URI of a resource is resolved or de-referenced on the Web. For example, when the URI for 'Afraz Jaffri' is put into a Web browser, the browser will issue an HTTP GET to the server, who in turn will redirect the browser, via 303, to a description of the URI in the format that was requested by the browser. Each description of the URI will itself have a URI so that the distinction between information and non-information resources is preserved. The architecture is illustrated in Figure 2.2 with the URI for 'Afraz Jaffri' used together with the URIs for a text/html and rdf+xml description of the URI.

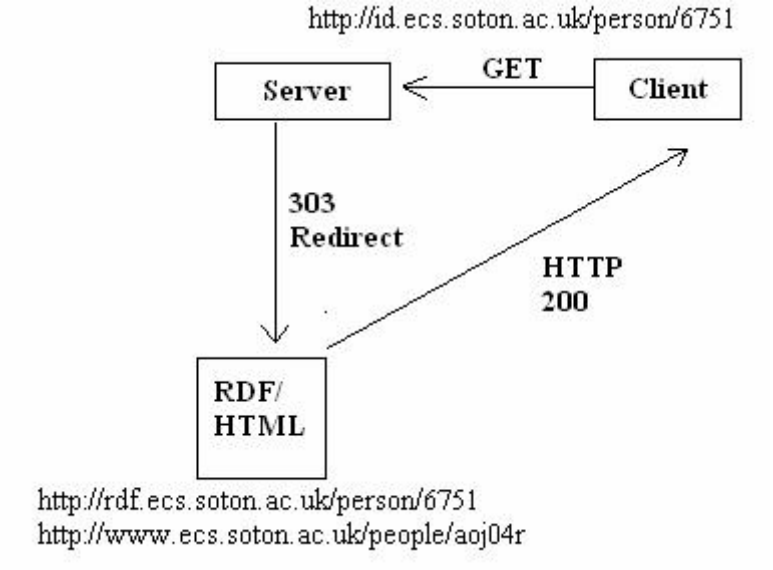

Fig.2.2. Semantic Web architecture for the resolution of URIs of non-information resources

2.4 Linking Open Data

The increased adoption of RDF and OWL as knowledge representation formats is enabling the production of Semantic Web systems that can manage, manipulate and display data in novel ways (Shadbolt et al., 2004, Lei et al., 2006). In order to encourage the adoption of Semantic Web technologies there has been an increased amount of activity in providing a linked data backbone that can be used to bootstrap the Semantic Web (Suchanek et al., 2007). The Linking Open Data project has been a catalyst for such activity, producing data sources that expose their knowledge as RDF and assert links between datasets. Current information sources include Geonames, DBLP, Musicbrainz, The CIA Factbook and US Census data.

The production of the first tutorial on how to link Open Data (Bizer et al., 2007) means that many more information providers are likely to make their knowledge available. Such activity will allow a formidable mass of knowledge to be used by Semantic Web applications. The linked data methodology has also introduced the use of additional techniques to publish Semantic Web data, such as using HTTP 303 redirects to dereference URIs about non-information resources, which have already allowed a new breed of Web browser to be built that can analyse and explore linked data (Berners-Lee et al., 2006).

The first set of data that is being used as a base for all subsequent data linkage is the DBpedia dataset. The DBpedia dataset reportedly contains over 91 million RDF triples and has knowledge covering over one million concepts. The knowledge has been extracted from Wikipedia info boxes that appear on Wikipedia pages. Consequently there have been over one million URIs created corresponding to each Wikipedia page that contains an info box. DBpedia URIs take the form http://dbpedia.org/resource/*resourceName* where *resourceName* is the name of a Wikipedia article. DBpedia has a lightweight property structure that has predicates derived from infobox data such as *name, placeofbirth, placeofdeath* and *capital.* There are also predicates used from other ontologies that link into the dataset including *foaf:page, rdfs:label* and *geonames:featureCode.*

2.5 The ReSIST Project

ReSIST (www.resist-noe.eu) is a network of excellence consisting of 18 partners working in the domain of resilience and scalability in IST systems. One of the deliverables for the project is to provide a Resilience Knowledge Base (RKB) that would aid researchers to find information about people, projects, publications and institutions in the field of resilient systems. The project also hopes to have resilient explicit metadata about systems and components that would aid users in building resilient systems. At present, the RKB contains RDF data from Citeseer, ACM publications, Cordis, DBLP and institutional data provided by each partner. The RKB also captures metadata from the project Wiki (Millard et al., 2007). An interface has been made where users can explore the RKB from the perspective of people, publications, projects or research area. One of the major problems associated with managing large amounts of data is the proliferation of URIs that are used to describe the same non-information resource. Frequently, the same author has different URIs from Citeseer, IEEE, DBLP and other publication repositories. There are often cases where within the same repository different URIs are used to refer to the same author and publication. The management of these URIs and the problem of coreference are given in Chapter 4. An application to search and browse through all of the data in the RKB is detailed in Chapter 5.

Chapter 3 Semantic Search

Current statistics show that the number of Web pages is in excess of one billion (Madhavan et al., 2007). With such a huge amount of data available some means had to be put in place so that exact information, relating to a particular purpose, could be searched for and retrieved. This need for relevant information spawned the production of search engines dedicated to sift through The Web and pick out documents and content that satisfy the requirements given to the engine by a user. However, a successful search for information can only be as good as the ability of the search engine to retrieve that information. Web pages, and any form of information given as text, is only read by a machine but is not understood by a machine. This means that although a text search for a word will be able to find that particular word, the machine cannot attach any meaning to the words it finds or is looking for. This severely limits the ability of a machine to retrieve data according to the needs of a human.

For example, if a person wanted to search a document or group of documents for information about 'men's clothes' they would have to tailor their thinking to extract the words that they thought an article that talked about this subject would contain, such as shirts, ties, coats, sizes, materials, price and so on. If, however, the machine somehow knew that a shirt was an item of clothing, or that coats come in four sizes and knew that ties could be of silk or cotton then the machine could automatically give all this information without the need for multiple searches.

This chapter sets out the problems of searching on the Web and gives some motivation for producing semantic search engines. Some systems already performing semantic search are described together with their limitations. The final section

introduces the problems of making a truly open world semantic search engine, which will provide the focus for the rest of the thesis.

3.1 The Problems of Search and Retrieval on the Web

One of the major uses of the World Wide Web today is to use search engines to find out about information. Due to the increase in the number of Web pages available, finding the desired information can often be an arduous and complex task. There are a number of problems with the current approach to finding results on the web:

- The most basic problem is that the information that you are looking for cannot be found.

  This could happen for a number of reasons:

  1. The information simply does not exist on the web.
  2. The search engine could not find the information that does exist.
  3. The information is in a web page that is not on the first page of returned results and the user does not try and look on through the next few results pages.

Passin (2004, p.107) gives an example where he could not find a replacement part for an old stove, even though there was a shop with a website that sold the part.

- There may be too many irrelevant results returned.

The ratio of the number of relevant results retrieved to the total number of irrelevant and relevant results retrieved is called the *precision* of a search. As a brief example, if a person wanted to know about the Isle of Wight Mosque, then entering this search term into Google produces a total number of 38600 results. However, out of these there are only 2 results that are about the Mosque. This gives a precision value of 0.00005%. The other results are pages that contain the words 'Isle of Wight' and the word 'Mosque' with little or no connection between them.

- Only some of the relevant results to the query are returned.

The ratio of the number of relevant results retrieved to the total number of relevant results indexed by the search engine is called the *recall* of a search. This value is harder to measure as it is virtually impossible to calculate how many relevant results to a search exist on the web. However, Clarke and Willett (1997) produce a relative estimation of recall based on pooling the results of a number of search engines to the same query. Shafi and Rather (2005) use this method to compare the precision and recall of five search engines in the retrieval of scholarly information in the field of biotechnology.

- The search engine searches for the query term in the wrong context and returns results in that context.

There are many cases where words have more than one meaning. If a user does not correctly add extra keywords to their search to contextualise the topic, they could receive anomalous results. For example, if somebody wanted to know about the Greek deity 'Nike', simply entering this word into a search engine will return results about the sportswear company Nike.

At present, Google has 46.3% (Sullivan 2006) of the search engine market. The reason for Google's success is claimed to be down to the PageRank algorithm (Brin et al. 1998) which ranks pages, not just by keyword frequency, but also according to how many pages link to a site, how many hits a site has and how often a site is updated, as well as other varying criteria (Arnold 2005). However, statistics also show that other search engines return results that are not found by Google (Notess 2002). This all means that the end user cannot make best use of the Web since the information or knowledge that the user requires is either difficult to find or search engine dependent.

The lack of a suitable search engine for the web stems from the fact that all search engines rely heavily on keyword frequency, and in essence, string matching to find their results. When a person enters the term 'football' into a search engine it has no knowledge about the concept of football, it will base its results on the number of times this keyword is mentioned in a document. There are algorithms that enhance this approach somewhat but the underlying principle is the same (Langville & Meyer 2004). What is needed is a system that knows that football is a type of sport and that it has a

World Cup and that it is played by 11 players etc. This kind of knowledge awareness is exactly what The Semantic Web is trying to achieve. If there was an ontology for football then all of the above facts could be easily modelled and used to enhance query results. Then, if someone was to make a query about football, the system would try and attain the concept of the search such as, is the person looking for football tickets? Are they looking for football kit? Or are they interested in a particular team? And so on.

The Semantic Web promises a new generation of World Wide Web infrastructure that will make it possible for machines to 'understand' the data on the web instead of merely presenting it. In order to encourage the increase of semantic web technologies there have been suggestions that a 'killer app' may be needed to convince those that are still unsure about the benefits that semantic web technologies can bring.

The search engine is an example of a potential 'killer app' that has been responsible for increased usage of the current web. As described in this section there are a number of problems with searching the Web today. Despite the improving quality of modern search engines, statistics show that only 17% of people find exactly the information they were looking for (Fallows, 2005). Furthermore, a study has shown that the recall of some search engines can be as low as 18% (Shafi & Rather, 2005). There is therefore a need to improve the quality of search results and user experience. The Semantic Web provides an opportunity to achieve such a goal. The use of RDF and OWL as knowledge representation formats can provide structured content to describe a given domain or set of domains. Using this knowledge, it should be possible to add a sense of 'understanding' to a search engine when searching for results whose knowledge has been partly or fully described in a knowledge representation format.

In the past, such a proposal may not have been viable due to the lack of ontologies and RDF resources available on the web. However, at the present time there are estimated to be more than 5 million RDF or OWL documents available on the Web (Ding, 2006). Even if most of those documents contain knowledge about a limited set of concepts, RDF data from sources such as DBpedia and Wordnet provide a suitable base from which to begin exploring the enhancement that can be made to ordinary Web searches. More importantly, there will be a number of knowledge bases that will be used to store RDF instance data and OWL ontologies that have the ability of being

queried using the SPARQL query language. Therefore, there are a number of components that need to fit together in order to achieve semantically enhanced querying that will be presented in Chapter 6.

3.2 Current Semantic Search Engines

There have been many projects under the general heading of 'Semantic Search' that work towards different goals and objectives. One approach to semantic searching is to restrict the ontology or knowledge base to a specific domain, and this is a much easier task to accomplish. One such system has already been demonstrated by the Royal Institute Elcano in Spain (Rodrigo et al., 2005). The institute's website provides information and commentary on the dealings of Spain with the rest of the world. In order for the site to be semantically enhanced, an ontology of International Affairs was made. The ontology was then populated by a semantic annotation tool that goes through the website and finds instances of the concepts in the ontology. These instances are then linked to the ontology so that when a query is made that matches a concept in the ontology, the instances of that concept are displayed. The display shows the knowledge about that concept and the documents where the concept appears. In this way the user has a choice of information source. This kind of technique is well suited to sites containing information about a particular domain and could be used in the future as a way of semantically improving the search of a website.

The Swoogle search engine attempts to index all semantic web documents (SWD) on the web (Ding et al., 2004). The query a user makes is usually in order to find an ontology that they can use that contains descriptions about their query item. This is a purely semantic web service, i.e. it deals only with SWD and not any other type of document available on the web. The main problem with Swoogle is that there are a few ontologies that have a high rank because they are imported by other ontologies and therefore are returned frequently back to the user. For example, a search in Swoogle for 'The University of Southampton' returns 4 URI's:

1. http://sweet.jpl.nasa.gov/ontology/data_center.owl
2. http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/akt_all_instance.rdf
3. http://triplestore.aktors.org/data/Southampton/southampton-themes.rdf
4. http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/soton_csd_instance.rdf

These URI's are links to ontologies or instance data that has knowledge about the term being searched for. In this case however, the first match is an ontology with a list of universities that is not particularly helpful. The third link is the only one that contains any useful knowledge about the University of Southampton with details of the research areas that the University is interested in and their corresponding URI's. This type of search is restricted by the amount of RDF that is available on the Web at the moment, but even for a relatively simple query the results above are not what would be expected.

The SemSearch search engine integrates ontologies and RDF data to provide a search facility for a departmental university website (Lei et al. 2004). Queries are semi-structured and require users to input a subject keyword as well as free text. This system is a closed world system i.e. it does not interact with the web and it cannot make use of knowledge from other repositories.

The TAP project (Guha & McCool 2003) is both a semantic web application by itself and also has the ability to interact with the Web. Unlike other ontologies or knowledge bases, TAP does not try to model concepts or definitions, but instead concentrates on modelling real world entities such as movies, athletes, musicians, places, people etc. This kind of search is ideal when somebody wishes to know detailed information about a particular instance of any of the categories that TAP contains. However, when more general information is required, the TAP knowledge base is less useful.

For example, a search for Tony Blair in the TAP KB returns a description of Tony Blair and other information such as his date of birth. The description is shown in Figure 3.1.

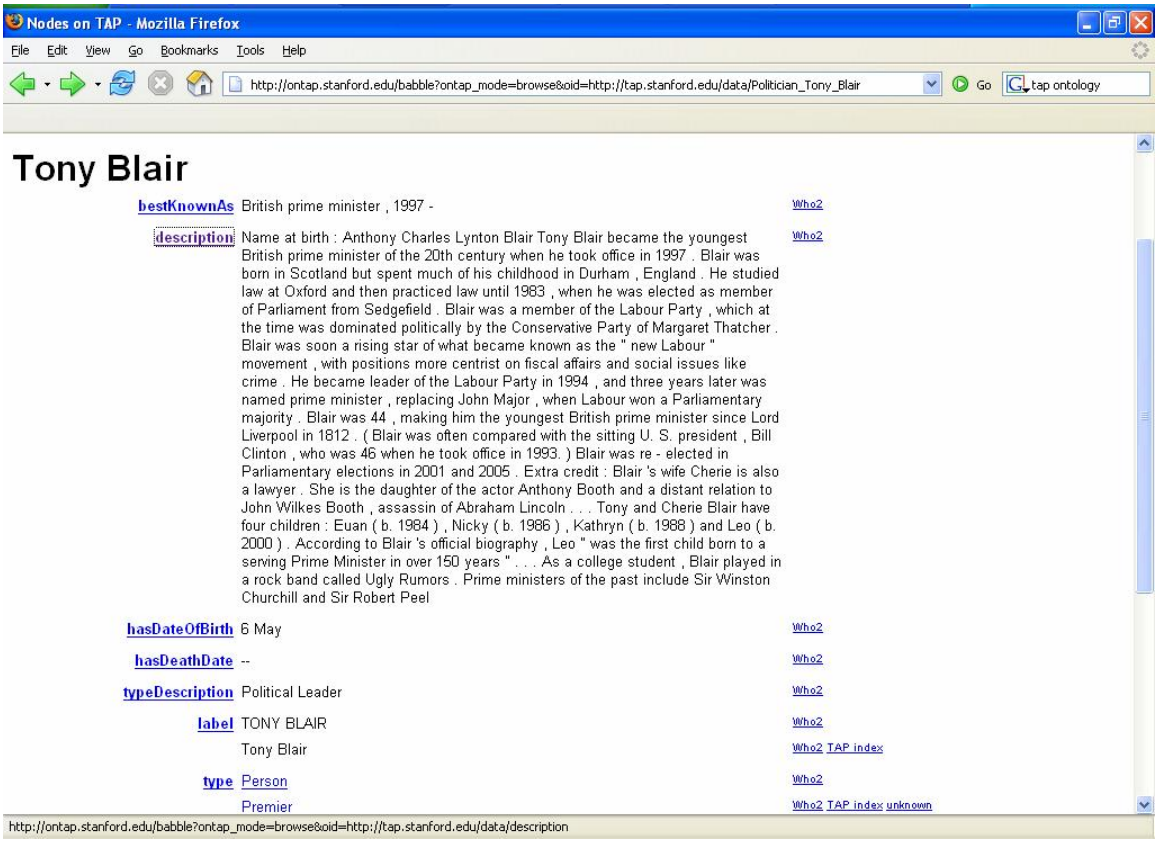

Fig.3.1. Knowledge about Tony Blair in the TAP interface

In a way, this kind of knowledge base acts as an encyclopaedia with the addition of links to news sites about where you can find the latest information on the query. However, when a search is made in the KB for the 'Labour Party' there is no such entry in the knowledge base. Such a query highlights the difficulty of capturing knowledge on such a grand and diverse scale. What is really needed from such a service is that when the term 'Tony Blair' is entered as a search term the system should recognise that Tony Blair is a member of the Labour party and it should know the names of the people in his cabinet, the number of votes he had in the last election, his key policy decisions in the last year, his approval rating, etc. This kind of functionality is being integrated into the Freebase application that is described in Section 3.4.

TAP is both a semantic web application by itself and also has the ability to interact with the WWW. TAP does not try to model concepts or definitions, but instead concentrates on modelling real world entities such as movies, athletes, musicians, places, people etc. The only limitation with the system is that it does not bring in knowledge contained in other repositories. This means that when a user searches for

things that are not inside the TAP knowledge base, very little useful information is returned.

3.3 Searching the Open Semantic Web / Linked Data

The amount of Linked Data that has been produced for use on the Semantic Web has spawned a new generation of search engines that are specifically targeted at utilising the linked structure to improve search results. The search engines given below are not fully operational, but a brief description of each will be given in order to demonstrate potential applications that can be provided for use on the open Semantic Web. A description of a Web 2.0 site, Freebase, is also given in the next section in order to compare the attractiveness and functionality of Semantic Web applications with Web 2.0 applications.

3.3.1 DBpedia

The latest project that is being developed by the Linking Open Data project is DBpedia. Such a large scale repository provides an ideal starting point for implementing a semantic search engine and such a system has already been produced (http://dbpedia.org/search). The search interface consists of a standard text entry box where keywords are entered. On submission, the search application returns a tag cloud of related concepts and an ordered list of articles that match the given keywords. The tag cloud provides a useful means of disambiguating concepts or items that have one or more meanings. The results are also available as RDF and can be integrated into other applications.

However, DBpedia is by no means an exhaustive reference for every concept or entity. The project is also in the initial stages and not everyone can be expected to provide data that has been linked to their own. Searches in which there are a number of different keywords are less useful than searches that have just one keyword relating to one specific topic. A SPARQL endpoint is provided so that searches that cover more than one topic can be easily constructed.

3.3.2 Sindice

Sindice is a Semantic Web Crawler, that indexes and crawls Semantic Web documents in order to find the places where a particular URI is mentioned (Tummarello et al. 2007). It can be likened to the Swoogle search engine, with the difference being that only URIs and not keywords can be entered into the search box. Sindice uses a number of other Semantic Web services in order to discover and index RDF documents. When a particular URI is searched for, Sindice displays a list of those documents in which the URI appears. The primary function of the search is for the results to be incorporated in other applications that may have a 'see also' feature.

The disadvantage with this service is that the data exposed in DBpedia as Linked data has not been crawled or indexed. This makes the available search space very shallow. The problem of indexing the entire Semantic Web maybe even harder than the task of indexing the document web because of the nature of different URIs associated with information resources and non-information resources. If this problem is solved then Sindice could overtake Swoogle as being the standard Semantic Web crawler.

3.3.3 Zitgist

Zitgist will attempt to be for the Semantic Web what Google is for the Document Web (Passant et al., 2007). All linked datasets that have published in RDF will be indexed by Zitgist as well as normal HTML documents. The service aims to give a graphical user interface so that SPARQL queries can be made on Semantic Web data. This would potentially make it easier to search for queries that have a property-like structure. For example, finding the number of players that have won the FA cup playing for Manchester United would involve having to trawl through multiple web sites and issuing different queries to Google. Zitgist would make the process a lot simpler by allowing the user to filter the properties that they require. Therefore the above query could first be for the Manchester United teams that have won the FA cup and then a second search for the players that were in those teams. Zitgist is still not operational and it remains to be seen whether the interface is easy to use and whether the results returned are relevant to the query.

3.3.4 Watson

Watson claims to be a gateway for the Semantic Web and a replacement for Swoogle (D'aquin et al., 2007). The main focus of the search engine is to index ontologies according to some ontological quality metrics and make use of implicit relations between ontologies available on the web. Watson then enables users several different ways of querying the data, from simple keywords to URIs. The main difference between Watson and Swoogle is the way in which each system treats Semantic Web Documents. Swoogle's approach is to use Web based methods for indexing and retrieval, where as Watson takes into account the inherent nature of an ontology and utilises relations such as *owl:imports* and *rdfs:seeAlso* to discover more ontologies. Watson also eliminates duplicate ontologies that are found in multiple locations giving a much improved precision rating over Swoogle.

The main deficiency of Watson is that it is primarily concerned with indexing and searching ontological content, such as *owl* or *rdfs* documents. However, the vast majority of RDF that has been made available by the Linking Open Data project contains no ontology or explicit semantics. This limits the usefulness of Watson to applications that require ontology mediation or mapping. At present, such applications have not matured and are low in number. Nevertheless the services that Watson has to offer will be useful in the future when more ontology based applications are released onto the open Semantic Web.

3.4 Freebase

Freebase is not a Semantic Web application in the traditional sense. The idea behind freebase is to create a structured Wikipedia type site, where users can add properties and values for different entities in a particular domain. This structured information can then be queried using MQL (Metaweb Query Language) in order to extract structured information about particular topics and use them in applications or display them on a website.

Freebase consists of a number of domains such as Arts and Entertainment, Society, Sports and Money. Within a domain there are a number of types that correspond to entities within the domain. For example, the Special Interests domain

consists of a number of sub domains, one of which is 'Automotive'. The automotive domain then has types for company, make, model etc. Each type is then further broken down into instances of a type, which are called topics. The company type has topics for BMW, Ford, Mercedes etc. A type has associated with it a type definition that consists of a number of properties that are common for that type. The automotive company type has a definition which consists of properties for name, founded date, founder name, operating income, net income etc. There is also an 'included type' property that enables multiple types to be added. An automobile company is also a 'company' and a 'legal entity' so they are included on its included type property. This association means that the type definitions for 'company' and 'legal entity' are also added to the type definition for 'Automobile Company'.

Although Freebase currently does not use standard RDF or OWL to encapsulate the knowledge that is being gathered, the structured information could easily be processed for inclusion on the Semantic Web. Domains and Types closely correspond to classes and subclasses in an ontology, type definitions can be mapped as properties or slots and Topics can become instances of a class. It is also interesting to note the way in which the structured data is being collected. Freebase is following a bottom up approach, with all domains, types and properties being constructed form scratch. This is in contrast to DBpedia, which is taking a top down approach by extracting metadata from an already existing source. In theory this should mean that Freebase will provide a better structure and schema for their data than DBpedia, although presently the quality of information cannot be judged as the data gathered by Freebase is still sparse. Furthermore, the considerable amount of interest in Freebase has meant that there have already been applications that have developed using the Freebase API that greatly outnumber the application that have been built using DBpedia data, even though Freebase is currently only at the alpha release stage.

Each topic in Freebase has its own URI, although the distinction between information and non-information resources is not made. Therefore, the page URI for BWM, http://www.freebase.com/view/?id=%239202a8c04000641f8000000000009faf, which is depicted in Figure 3.2 is also the same as the URI for the company BMW itself. Freebase is seen as more of a Web 2.0 site than a Semantic Web site and therefore the developer community is a lot more active. This has implications for the

way that the Semantic Web will grow, given that eventually Freebase may turn itself into a Semantic Web site. The more people that are attracted to Freebase, the less will be attracted to projects such as DBpedia and Linking Open Data even though at some stage the two may be integrated together.

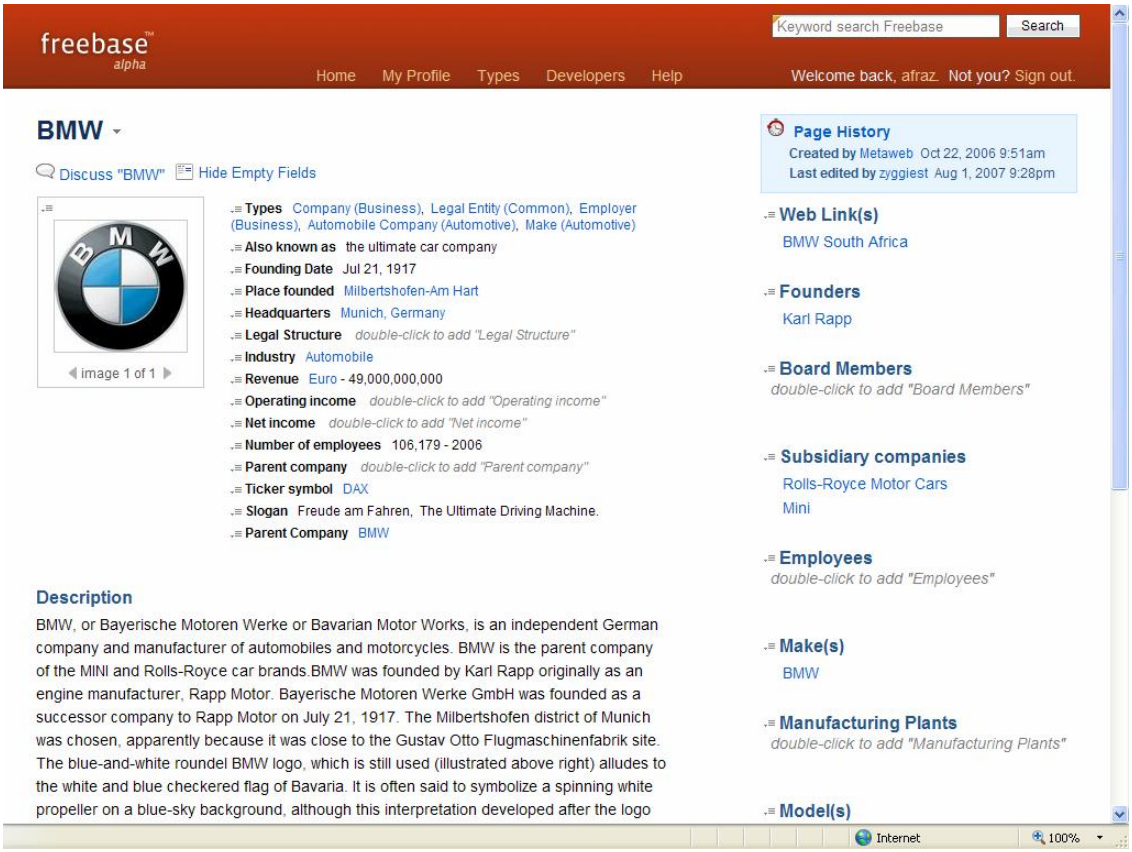

Fig.3.2. The Freebase page for the Topic 'BMW'

One component of Semantic Searching that has been largely overlooked by existing applications is the need to manage the huge amount of URIs that have been created and the overlap between them. There are many non-information resources that are being given URIs for which there are already existing URIs present. The motivation for studying this overlap between URI identities and how they can be managed will be presented in the next chapter.

Chapter 4 Coreference

4.1 The Problem of Coreference

The explosion in the number of information sources being exposed as RDF has also led to an explosion in the number of URIs used to identify different entities. It is often the case that data in different repositories will hold information regarding identical resources. For example, DBpedia, Geonames, the CIA Factbook and Eurostat all have different URIs for the same country. In another context, Citeseer, DBLP, IEEE and the ACM have different URIs for the same authors and papers. The responsibility of giving an information or non-information resource a URI lies with the data provider and they will assign URIs based on the web domain over which they have control.

The multiplicity of URIs leads to the problem of *coreference*, where different URIs are used to describe the same entity. On an open Semantic Web this presents a problem when there is a need to link together knowledge from disparate information providers. The present approach, used by the Linking Open Data community, is to use various equivalence mining techniques in order to assert *owl:sameAs* relations between entities that are considered to be the same. DBpedia has, for example, made an assertion that: <*http://dbpedia.org/resource/Berlin*> is <*owl:sameAs*> <*http://sws.geonames.org/2950159/*>. In this chapter it will be argued why this is not the best approach for dealing with coreference, and a system will be proposed for dealing with consistent reference across multiple knowledge bases. The next section describes the problem of coreference and where the problem occurs in other domains. Section 4.2.3 shows how coreference is a problem on the Semantic Web.

4.2 Coreference and URI Identity

The term ‘coreference’ is used in the field of linguistics to define the situation where different terms are used to describe the same referent. This is often done using words such as ‘he’, ‘she’, ‘we’, ‘them’ or ‘it’. On the Semantic Web we use the term coreference to define the situation where different URIs are used to describe the same non information resource. This section gives a brief description of coreference in information science and databases and then goes on to discuss the meaning of a URI and the necessity of giving coreference due importance in Web architecture.

4.2.1 Coreference in Information Science

The problem of coreference within the field of information science has existed for many years and the solution to the problem is based around the use of controlled vocabularies. Such a solution is possible because of the closed world nature and human processing characteristics of a library system. The most popular closed vocabulary is the Library of Congress Subject Headings (LCSH). This vocabulary gives a defined and precise meaning to each subject in the vocabulary which contains over 280 000 terms. Thus it is not possible for people to make their own subject headings, keyword descriptions or tags as is prevalent on the Web today.

A more relevant case of coreference occurs in digital libraries when the author of a publication has to be disambiguated. There are many authors who share the same name and matters are made more complex by the use of initials, different naming formats and spelling errors. For example, the author ‘Hugh Glaser’ could be represented with his full name or by using ‘H. Glaser’, or ‘Glaser, H.’. The task of author disambiguation is an active area of research in information science and many solutions have been proposed. Some solutions use Web based searches in order to determine if one author is the same as another (Yang et al., 2006, Tan et al. 2005). Such techniques will be needed on the Semantic Web if a consistent web of data is to be created.

The ReSIST project has published RDF data from institutions such as the ACM, Citeseer, IEEE and DBLP. Amongst these datasets the problem of author, as well as paper, disambiguation has proved a challenging task. Each repository has its own

naming scheme for authors and publications and the overlap between each repository is significant. In order to have a Semantic Web that provides the scalability to cope with such inconsistencies the issue of coreference will have to be addressed. Our proposed solution for this problem is given in Chapter 5.

4.2.2 Coreference in Databases

Within the database community the problem of coreference is referred to as record linkage. The need for record linkage arises when records or files from different databases need to be joined or merged. Each database could have duplicate records of the same person or thing which, when amalgamated, would make the data inconsistent or 'dirty' (Hernandez & Stolfo).

Record linkage has a well defined mathematical theory as proposed by Fellegi and Sunter (1969). The theory is based on records referring to the same entity having a number of characteristics in common. If *a* and *b* are elements from populations A and B and some elements are common to A and B then two disjoint sets can be created. The first set, M, is the set of elements that represent identical entities and the second set, U, is the set of elements representing different entities. Now if α(a) and β(b) refer to records from databases A and B respectively and each record has *k* characteristics then a comparison vector, γ is defined that contains the coded agreements and disagreements on each characteristic:

$$\gamma[\alpha(a),\beta(b)] = \{\gamma^1[\alpha(a),\beta(b)],\ldots,\gamma^k[\alpha(a),\beta(b)]\} \quad \textbf{(1)}$$

The theory then gives the probability of observing a specific vector given (a,b)ЄM as:

$$M(\gamma) = \sum_{(a,b)\in M} P\{\gamma[\alpha(a),\beta(b)]\} \cdot P[(a,b) \mid M] \quad \textbf{(2)}$$

This theory can also be used for identifying the probability of having identical resources on the Semantic Web. If A and B are two RDF graphs which have URIs *a* and *b* identifying the same resource such that *a*ЄA and *b*ЄB, then when the following graphs of each URI are made:

CONSTRUCT {<URI-1> ?ap ?ao} WHERE {<URI-1> ?ap ?ao} } **(3)**

CONSTRUCT {<URI-2> ?bp ?bo} WHERE {<URI-2> ?bp ?bo} } **(4)**

α(a) can be substituted by the result ?ao and β(b) can be substituted by the result ?bo. The characteristics k^i can then be substituted by the union of the result of ?ap and ?bp. The equations (1) and (2) will then hold with these substituted values. The result only shows the probability of two URIs referring to the same entity and can only be used as a basis for coreference resolution as it has been used in the database community for the same purpose.

The problem with this theory is that it is assumed that there is at least one characteristic that is in common across the different graphs. In the Semantic Web this is not always the case as different graphs will have different predicates for a particular resource and there may be little or no overlap between predicates. For example the URIs <http://sws.geonames.org/2950159> and <http://dbpedia.org/resource/Berlin> are URIs for Berlin that each have over 100 predicates, yet only one of them is in common. There is also the problem that predicates in the Semantic Web are not like database field names that have common names. Therefore, ontology matching techniques need to be used in order to match predicates between different resources, so that characteristics can then be matched.

There are thus more advanced algorithms that need to be developed in order to identify identical URIs on the Semantic Web. However, the problem is not just of finding equivalences, but what to do when candidate equivalences are found. This problem is also discussed in Chapter 5.

4.2.3 Coreference in the Semantic Web

The subject of coreference on the Semantic Web has been raised previously (Alani et al., 2002), but it was not pressing or therefore studied for many years because of the lack of real scalable RDF data that was freely available. However, the Linking Open Data project and our own ReSIST project are highlighting the need to have some form of URI management system. For example, the following are all URIs for Spain:

```
http://dbpedia.org/resource/Spain
http://www4.wiwiss.fu-berlin.de/factbook/resource/Spain
http://sws.geonames.org/2510769/
```

```
http://www.daml.org/2001/09/countries/fips#SP
http://www4.wiwiss.fu-
berlin.de/eurostat/resource/countries/Espa%C3%B1a
```

These URIs come from 5 different sources. There are also at least 9 URIs for Hugh Glaser that originate from 6 different sources:

```
http://acm.rkbexplorer.com/rdf/resource-P112732
http://citeseer.rkbexplorer.com/rdf/resource-CSP109020
http://citeseer.rkbexplorer.com/rdf/resource-CSP109013
http://citeseer.rkbexplorer.com/rdf/resource-CSP109011
http://citeseer.rkbexplorer.com/rdf/resource-CSP109002
http://dblp.rkbexplorer.com/rdf/resource-27de9959
http://europa.eu/People/#person-0ff816fa
http://resist.ecs.soton.ac.uk/wiki/User:hugh_glaser
http://www.ecs.soton.ac.uk/info/#person-00021
```

These URIs have been grouped together because we believe they all refer to the same non-information resource. However, the standard way of dealing with such a plethora of URIs is to use *owl:sameAs* to link between them. The semantics of *owl:sameAs* mean that all the URIs linked with this predicate have the same identity (Bechofer et al., 2004), this means that the subject and object must be the same resource. The major disadvantage with this approach is that the two URIs become indistinguishable even though they may refer to different entities according to the context in which they are used. For example, consider the case where a person has a URI at one institution and then moves to another institution that provides another URI. If the person makes an *owl:sameAs* link between them then it will not be possible to differentiate between the person as they were at the first institution and the person as they are at the second institution. The knowledge about the person at institution 1 and institution 2 effectively become merged so, for example, the addresses would not be able to be separated.

Even worse, an incorrect equivalence can cause other incorrect equivalences to be inferred. For example, it was found that one of the ReSIST project investigators (Tom Anderson) had extra information which appeared plausible, but was not correct. The information was finally tracked down to DBLP, where the two Tom Andersons had been conflated.

We subscribe to the belief that the meaning of a URI may change according to the context in which it is used (Booth, 2006). For example the URIs that refer to Spain given above could refer to 'Spain the political entity', or 'Spain the geographic location', or 'Spain the football team'. Some people would be happy to use each URI interchangeably because they do not care about the precise definition, whereas others will want a URI that specifically matches their intended meaning. There is a requirement to have a system that deals with URIs about the same resource that are not exactly identical. The semantics of *owl:sameAs* are too strong and other alternatives like *rdfs:seeAlso* do not fit the intended purpose. Such a requirement is vital if data is to be cleanly linked together in a consistent fashion. The next section details an initial attempt to handle URI management called the Consistent Reference Service (CRS).

Chapter 5 The Consistent Reference Service

The Consistent Reference Service (CRS) has been created in order to manage coreference between the millions of URIs that are accumulating on the Semantic Web. This section will describe the concept of a *bundle* that groups together URIs referring to the same resource, and also describe the implementation and architecture of the CRS.

5.1 URIs and Bundles

The CRS service has been implemented as both an RDF knowledge base and a relational database with RDF export. The CRS sits in the Semantic Web as any other knowledge base or database would. Each data provider maintains one or more CRSes for their own knowledge. In the ReSIST project there are over 15 repositories each with their own CRS.

The CRS introduces the concept of a *bundle* to group together resources that have been deemed to refer to the same concept within a given context. Different bundles may be used to group together URIs of the same resource in different contexts. For example, there may be a bundle containing all of the URIs about a person in the context of institution 1; and another bundle containing all of the URIs about the same person in the context of institution 2. Each CRS can use different algorithms to identify equivalent resources. For example, the algorithms to detect equivalence amongst authors are different from the algorithms used to detect equivalence between countries. To begin with, each URI in a repository has its own bundle in the CRS. When an equivalence is detected the bundles containing the URIs are merged together to create a

new bundle. In this way successive iterations group together larger bundles, with each bundle having an anonymous URI.

The concept of a bundle is defined as a class in a coreference ontology used by the CRS. There is also a database schema that maps onto the ontology. Every resource that is defined as being of *rdf:type coref:Bundle* can have the following properties:

coref:hasCanonicalReference – One URI in a bundle can be made to be the canonical representation i.e. the preferred URI that one should use.

coref:hasEquivalentReference – The URIs in a bundle are grouped together using this predicate.

coref:updatedOn – The date of the last update to the bundle.

To illustrate let us take the example of the URIs referring to Hugh Glaser in the previous section. If we assume that we want to group together all the URIs that Citeseer has referring to Hugh then the triples asserted in RDF/XML format would be the following:

```
<rdf:RDF xmlns:coref=http://www.resist.ecs.soton.ac.uk/ontology/coref#
         xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <coref:Bundle rdf:about="http://www.rkbexplorer.com/crs/coref#bundle1">
       <coref:hasEquivalentReference rdf:resource=
       "http://citeseer.rkbexplorer.com/rdf/resource-CSP109020"/>
       <coref:hasEquivalentReference rdf:resource=
       "http://citeseer.rkbexplorer.com/rdf/resource-CSP109013"/>
       <coref:hasEquivalentReference rdf:resource=
       "http://citeseer.rkbexplorer.com/rdf/resource-CSP109011"/>
       <coref:hasEquivalentReference rdf:resource=
       "http://citeseer.rkbexplorer.com/rdf/resource-CSP109002"/>
       <coref:hasCanonicalReference rdf:resource=
       "http://citeseer.rkbexplorer.com/rdf/resource-CSP109002"/>
      </coref:Bundle>
</rdf:RDF>
```

The bundle mechanism provides an easy method to manage URI identities without having to incorporate expensive inference mechanisms. When dereferencing a resolvable URI the RDF document returned contains additional predicates identifying CRS services that may provide further information regarding the resource. If the user wishes, then they can assert explicitly *owl:sameAs* or *rdfs:seeAlso* links between the

equivalent URIs. The next section will look at how the CRS is used in conjunction with multiple knowledge bases and how bundles can be linked to other open data.

5.2 The CRS and Web Architecture

There have been many discussions in the Semantic Web community regarding the actual meaning of a URI. Does it refer to a sequence of bits? A Web page? A concept? These questions and others arising from the URI identity crisis (Halpin, 2006) are outside the scope of this thesis. We will use the definitions as given by the W3C Technical Architecture Group (TAG) to show how a coreference mechanism can be included in the current Semantic Web infrastructure.

The CRS can be treated as any other knowledge base, in that it contains knowledge about a particular URI. Our infrastructure implements the current best practice on how to serve linked data and uses cool URIs (Berners-Lee, 2003). As an example we will use the URI http://southampton.rkbexplorer.com/id/person-21 to represent the non-information resource, 'Hugh Glaser'. When a request is given to the server for a description of the URI, an HTTP 303 redirect is issued to one of two locations, depending on the accept headers sent by the client. If the requested content is application/rdf+xml then the server will generate an RDF description detailing the properties of the requested URI by issuing SPARQL CONSTRUCT queries to the appropriate knowledge base. The resulting description is cached and the 303 is issued to http://southampton.rkbexplorer.com/description/person-21. However, if the accept header is set to text/html then a 303 'See Other' is returned identifying an html description of Hugh Glaser at http://southampton.rkbexplorer.com/browse/peson-21. The server architecture conforms to the latest http-Range-14 (Fielding, R. 2007) recommendation of the TAG that involves HTTP 303 Redirects from the URI of a non-information source to an RDF or HTML information resource.

Each URI that is maintained by an institution will also have its own CRS. This CRS can be termed the 'home' CRS of a URI. The home CRS will provide a level of trust over what URIs it considers to be the same because it is the sole provider of that URI. To find all possible equivalences for a URI the following algorithm can be performed:

```
findEquivalence(URI u) {
      Dereference u;
      while (u coref:hasCRS a) {
            add a to equivalences;
            if(equivalences contains a)
                  break;
            findEquivalence(a);
      }
}
```

Finding all equivalences is entirely at the discretion of the application wishing to process the results of the search. If only one CRS is required, then only one iteration is necessary. Computing equivalences in this manner gives a considerable amount of flexibility in choosing duplicate URIs for a resource. Taking the URI management into a separate layer without fixing *owl:sameAs* links is an efficient and controllable way to manage coreference between URIs. The next section will describe an application that has been built using the CRS infrastructure.

5.3 A CRS Application: The Resilience Knowledge Base Explorer

Resilience Knowledge Base (RKB) Explorer is a Semantic Web application that is able to present unified views of a significant number of heterogeneous data sources regarding a given domain. An underlying information infrastructure has been developed that utilises the CRS architecture outlined in this chapter. The current dataset totals many tens of millions of triples, and is publicly available through both SPARQL endpoints and resolvable URIs. To realise the synergy of disparate information sources we are using the CRS system and have devised an architecture to allow the information to be represented and used.

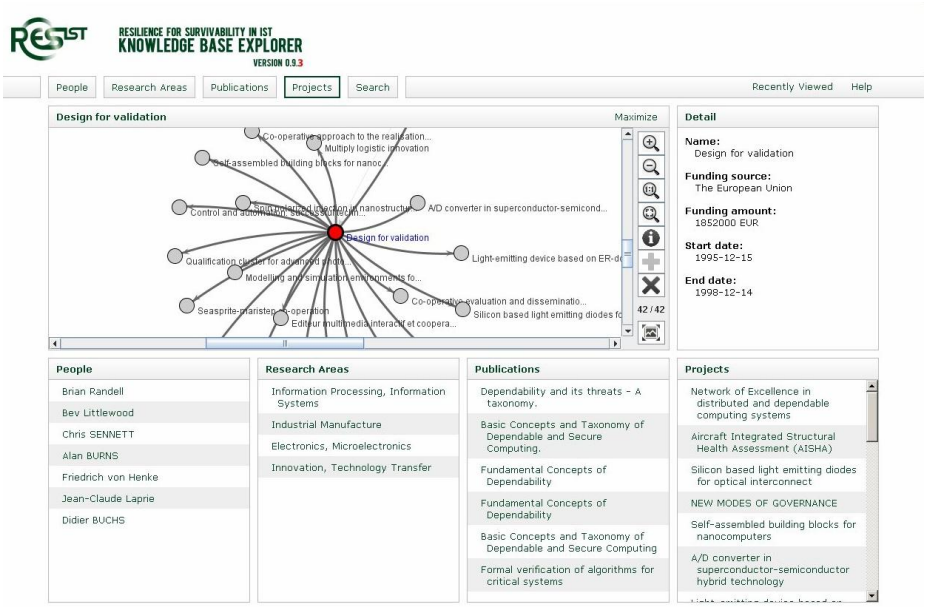

Fig.5.1. The single window interface of the faceted browser available at http://www.rkbexplorer.com/explore/

Figure 5.1 shows the user interface for the RKB Explorer. The main pane shows a chosen concept and related concepts of the same type that the system has identified as being related. In this figure, the ReSIST Project itself is under consideration, with its details on the right, and related projects are shown around it. These are chosen according to the relative weight given to ontological relationships, and the number of those relationships to each concept. The weight of the lines gives a visual ranking. They represent a project `Community of Practice' (CoP) for the project. Clicking on a resource will show the detail for it, while double-clicking will add the CoP for the new resource to the pane. This will then allow a user to see how different projects are related, and see the projects that provide linkage between them.

The panes in the lower half of the display show the related people, research areas, publications and projects, identified by similar ontologically informed algorithms, and are ranked by decreasing relevance. Thus the lower right-hand pane gives a list of the related projects found in the main pane, while the lower left-hand pane shows those people involved in the currently selected project.

The CRS system manages the URIs for each knowledge base. There are many URIs from each knowledge base that refer to the same resource, for example there are

hundreds of the same authors and papers in different knowledge bases, such as the ACM, IEEE and DBLP. Managing these millions of URIs has led to increased scalability and performance benefits as compared with taking an *owl:sameAs* approach. The RKB Explorer is being expanded and integrated with existing linked data and it is envisioned that the CRS system behind the explorer will also follow the same route.

Chapter 6 Future Work: Integrating CRS Functionality into the Semantic Web

6.1 Hypothesis

In order for Semantic Web search engines and other applications to work with the increasing amount of RDF data that is being made available on the Web, there needs to be a URI management system that will track URI usage and coreference between URIs. Such a system will need to:

- Detect and group together URIs referring to the same resource.
- Provide a service so that URIs can be added to a bundle.
- Provide a service so that URIs can be removed from a bundle.
- Integrate with existing linked data.
- Provide a suitable query mechanism so that bundles for a resource can be quickly discovered and used by other Semantic Web agents.
- Track the provenance of URIs.
- Provide a service to link URIs that the owner of the resource, or the resource itself have deemed to be the same through *owl:sameAs*.

The CRS has been implemented as a stand alone system that, at present, is being used in a closed world system. The most challenging of the above mentioned tasks is for the CRS to be integrated on the open Semantic Web. In order to demonstrate the

functionality and assess the improvements that can be made by using a CRS, a semantic search engine with an integrated CRS will be constructed. The flexible architecture of the system means that it can be adapted according to the developments that take place with the search engines mentioned in Section 3.3. If more effort needs to be made in making the CRS fully integrated and linked to open RDF data then the search aspect and integration with the document web can be suppressed in preference for extending the research of the CRS.

6.2 Proposed Architecture

The system architecture comprises of three main components as shown in Figure 6.1. A user enters their keywords (KW) in a normal Google style search box without the need for using special syntax or constructs. These keywords are then fed to the knowledge manager who passes them on to the knowledge mediator.

The knowledge mediator has access to an arbitrary number of knowledge bases and a CRS that are accessed over HTTP using SPARQL. The mediator then queries the CRS for concepts that match or are similar to the given keywords. The CRS returns a 'bundle' of resources that have been found to be the same. The knowledge Mediator then issues a DESCRIBE query on each URI to find the properties and literals for each resource. The RDF returned is then passed on to the Knowledge Manager who looks for links such as '*foaf:page*', '*rdfs:seeAlso*' and '*dbpedia:reference*' so that a list of web pages associated with the query can be returned. The knowledge manager then displays the knowledge and web links to the user according to the type of concepts returned. For example, the distributed knowledge bases contain definitions, articles, links, publications and other assorted information; these results are returned to the user as views relating to one of the concepts. The user can then select whichever view they choose to explore the results further. Formally:

Let *Concept* be the set of concepts contained in all knowledge bases.

Let *URI* be the set of URI's that represent each concept.

Let *Bundle* be the set of bundles whose elements are URI's representing a concept.

Let *V* be the set of property values of a URI.

We have a function BundleOf that returns one bundle for a given concept:

$$BundleOf : Concept \rightarrow Bundle$$

We have a function Describe that gives the property values for a given URI:

$$Describe : URI \rightarrow \mathrm{P}(V)$$

The result of a query for a concept from the Knowledge Mediator is:

$$\forall a \in Concept, \mathrm{Re}\,sult(a) = \bigcup_{i \in BundleOf(a)} Describe(i)$$

The results that are returned to the user are both document links and knowledge in the form of RDF statements that were returned from the knowledge mediator when the original query was sent. This enables the user to understand how the results were achieved. An investigation into whether the knowledge alone is enough to satisfy a user's query is planned as part of the research.

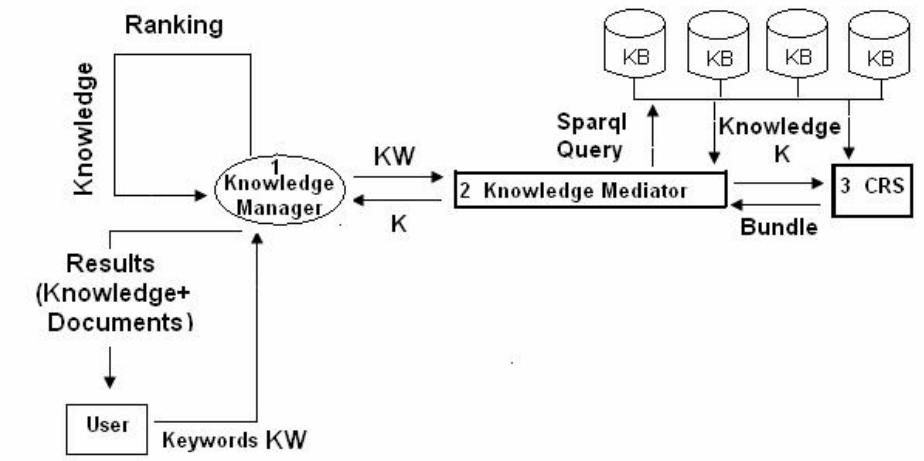

Fig.6.1. The system architecture diagram shows the three main components numbered 1, (Knowledge Manager), 2, (Knowledge Mediator) and 3, (CRS).

6.3 Prototype Application: Google Maps Country Info Mashup

The initial research that led to the architecture described in Section 6.2 being formed has been successfully completed. A proof of concept system is now being implemented that will take the form of a Google Maps mash-up of Wikipedia, DBpedia, CIA Factbook (https://www.cia.gov/cia/publications/factbook/index.html) and Geonames (www.geonames.org) data. These four sources contain an extensive set

of information about all the countries of the world. The four sources each have their own SPARQL endpoint that can be queried from the web.

The interface for the system takes the form of a Google Map where the user can select any country. The system then performs queries over the RDF and returns information about the country from the different sources. The results are presented so that different types of information are separated on screen. The separation of the different types of information is achieved by looking at the predicates of the triples that are returned. The types of information vary from country to country and include geographical, social and political information as well as the people associated with a country and the events that have taken place in a country. Each type can be explored in more detail so that links to web sites can also be seen.

The CRS for the system identifies URI's from each knowledge base that refer to the same country. The DESCRIBE query issued to each URI then performs similarity matching on the properties to filter out duplicate entries. The Knowledge Manager then looks at each property to determine the type of information that is being referred. The results are then presented to the user under the Google Map.

6.4 System Components

6.4.1 Ontology Mapping

Research into ontology mapping has been continuing for a number of years as it addresses one of the major challenges for the Semantic Web. There are a number of ontologies that are currently being used for a number of different domains. There are also ontologies, termed 'upper ontologies' that try to form a base of concepts that can be used in other ontologies. Examples of such ontologies are SUMO and CYC. There are also a number of domain specific RDF vocabularies that can be extended into custom built ontologies such as SIOC and SKOS.

If two ontologies have a representation of the same concept, then ontology mapping refers to the technique used to 'match' the concepts from both ontologies. Such a system is vital for the Semantic Web as there is no single universal ontology

that is used by everyone. For example a person could represent his email address using either *foaf:mbox* or *vcard:email*. The two properties represent the same concept, yet in order for data to be dealt with and understood by Semantic Web agents, the properties will have to be understood to be the same.

The architecture given in Section 6.2 will require a component that can perform ontology matching to a reasonable level. It is not proposed that any new research in ontology matching will be performed. There are a number of existing systems that have been produced or are currently in development that can be used in the system. The prototype application has demonstrated the need for ontology matching tools to be deployed in the full application. The different ontologies used by the data providers in order to describe countries have a considerable amount of overlap. Each provider will frequently provide their own predicate for properties such as population, area, capital etc. A good ontology matching tool would be able to detect these similarities given the overall structure of the data. Two existing tools that are being investigated for this task are CROSI (Kalflogou et al., 2005) and Falcon (Jian et al., 2005).

6.4.2 Equivalence Mining

The variety of different datasets that are being published as RDF has led to the problem of equivalence mining. Equivalence mining simply refers to the task of detecting equivalences between URIs of non-information resources. This has particular reference in the field of coreference, since in order to be able to group URIs into a bundle, the URIs must first have been flagged as being potentially equivalent. At present there is very little in the way of tools or applications available to support equivalence mining. Those tools that have been made are based on a specific domain or task and cannot be universally applied to the whole Semantic Web. Raimond (2007) provides an algorithm for detecting equivalences between the Yamendo and MusicBrainz datasets. This type of application typifies the approach taken to equivalence mining, which is to develop algorithms on the basis of the datasets that need to be linked. An algorithm for detecting equivalences between authors in a knowledge base is being developed in the ReSIST project. The system described in Section 6.2 will need to use an equivalence mining application in order to find potential URIs that can be included in a bundle. The increasing number of datasets being published means that there will always need to be equivalences made between URIs

since this is the basis on which open data is being linked together. These equivalences can be easily included in the CRS that will be used by the system described in Section 6.2.

6.4.3 Semantic Web and Document Web Integration

There has already been a proposal to unify the way in which data is retrieved from the document web and the Semantic Web, (Immaneni & Thirunayanan, 2007). The Unified Web consists of ordinary Web and Semantic Web documents that are linked together by nodes (documents) and edges (links). The proposal however does not take into account the recent http-range-14 resolution concerning information and non-information resources. This means that a URI for a document is considered to be the URI of the thing that the document is describing. The Unified Web model that is proposed has its own proprietary query language that is based on simple set theory.

A more pragmatic approach to document and Semantic Web integration lies within the Semantic Web itself. There are a large number of *rdfs:seeAlso* predicates in the Semantic Web whose objects are ordinary HTML documents. Web pages that have been associated with a URI are more likely to give related information about a resource than Web pages that are found by means of a search engine. It is also much easier to merge information from the document web and the Semantic Web through the use of *rdfs:seeAlso* links than forming a specialised data structure or query language. The Search Engines mentioned in Section 3.3 may be used to provide this feature in the system described in Section 6.2.

6.4.4 Knowledge Ranking

At present there is no accepted way of ranking knowledge on the Web. A method for ranking ontologies has been produced by Alani et al. (2006). The Swoogle search engine described in Section 3.2 uses an adaptation of the PageRank algorithm used by Google to rank ordinary Web pages (Brin et al., 1995). However, ranking knowledge on the Semantic Web should not just follow the pattern of ranking used for text and HTML pages. The very nature of RDF means that more sophisticated ways of ranking knowledge are available. The graph structure of RDF can be utilised to find relevant matches to a query and the graph can then be traversed to find a finite set of related

resources. The ReConRank (Hogan et al. 2006) system retrieves a set of resources from which a subgraph is derived. This is combined with the set of named graphs to effectively build a graph of quads. The resources and links from the quad graph are then extracted and analysed and a rank of relevant resources is created based on the number of links that a resource has to a named graph. Systems such as this have been used to efficiently query large RDF stores (Hogan et al. 2007). Such a technique could also be used to rank knowledge on the open Semantic Web as RDF triplestores have become suitably developed so that it is possible to pull in large amounts of RDF and perform queries in seconds (Harth & Decker, 2005).

6.5 Research Methodology

Once the prototype system has been fully implemented the research will then focus on broadening out the searches for queries on any kind of entity, not just countries. The main focus of the research will be into integrating the CRS into the open Semantic Web. The search engine using the integrated architecture will then be used to compare the results of searches made with the CRS and without the CRS. Services such as DBpedia and Zitgist will be used as comparisons. The performance will not simply be measured by the relevancy of the results returned but on the recall of the results. The CRS groups together URIs from multiple sources, which is a process that is not replicated by other Semantic Web applications. Therefore, in order to prove the hypothesis it will be necessary to check whether the URIs that result from a search include all possible uses of the URI.

Together with the practical investigation there is also a theoretical aspect to the CRS. Research will be carried out into the consequences of using *owl:sameAs* compared with using a CRS to link URIs. This will involve looking into the semantics of *owl:sameAs* and what inferences stem from its usage on the open Semantic Web. It should also be possible to formally characterise the bundle mechanism and analyse how it changes or affects the structure of an RDF graph.

6.6 Summary

This thesis has given an overview of current Semantic Search technology and provided details of the next generation of Semantic Search based on Open Data. The problem with coreference on the Semantic Web has been presented and compared with similar problems in other disciplines. A solution to the coreference problem has also been detailed. The future work section has explained the architecture that will be built following on from the initial prototype used to gather information for countries from different knowledge repositories.

References

Alani, H., Dasmahapatra, S., Gibbins, N., Glaser, H., Harris, S., Kalfoglou, Y., O'Hara, K. and Shadbolt, N. (2002) Managing Reference: Ensuring Referential Integrity of Ontologies for the Semantic Web. In Proceedings of 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW'02), pp. 317-334, Sigenza, Spain.

Arnold, S.E., 2005. *The Google Legacy.* Tetbury, England: Infonortics.

Bechofer, S., Van Harmelen, F., Hendler, J., Horrocks, I., Mcguiness, D.L., Schneider, P.F. & Stein, L.A. 2004. OWL Web Ontology Language Reference, Technical Report, W3C,[online] http://www.w3.org/TR/owl-ref/

Berners-Lee, T., Hendler, J. & Lassila, O., 2001. The Semantic Web, Scientific American [online], May 2001, Available from: http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2 [Accessed 4 April 2006].

Berners-Lee, T., 2007. Semantic Web "Layer Cake", [Online], Available from: http://www.w3.org/2007/03/layerCake.png [Accessed 10 July 2007].

Berners-Lee, T. Cool URIs Don't Change,2004, [online] http://www.w3.org/Provider/Style/URI [01 August 2007]

Berners-Lee, T., Chen, Y., Chilton, L., Connoly, D., Dhanara, R., Hollenbach, J., Lerer, A. & Sheets, D., 2006. Tabulator:Exploring and Analyzing Linked Data on the

Web. Proceedings 3rd International Semantic Web User Interaction Workshop. Athens,Georgia,USA.

.Booth, D. URIs and the Myth of Resource Identity, Proceedings of the Workshop on Identity, Meaning and the Web (IMW06) at WWW2006, Edinburgh, Scotland.

Bizer, C., Cyganiak, R. & Heath, T., How to Publish Linked Data on the Web, [online], http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/ [20 July 2007]

Brin, S., Page, L., Motwani, R. & Winograd, T., 1998, *The PageRank citation ranking: Bringing order to the web*. Stanford Digital Libraries Working Paper.

Clarke, S., & Willett, P., 1997. Estimating the recall performance of search engines. *ASLIB Proceedings*, 49 (7), 184-189.

D'Aquin, M., Sabou, M., Dzbor, M., Baldassarre, C., Gridinoc, L., Angeletou, S. & Motta, E. WATSON: A Gateway for the Semantic Web. *European Semantic Web Conference,*2007, Innsbruck, Austria.

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, S., Peng, Y., Pavan, R., Doshi, V.C. & Sachs, J. 2004. Swoogle:A Search and Metadata Engine for the Semantic Web. *In Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management.* November, 2004, ACM Press.

Ding, L., 2006. Enhancing Semantic Web Data Access, PhD Thesis, University of Maryland [online], http://ebiquity.umbc.edu/paper/html/id/317/Enhancing-Semantic-Web-Data-Access

Fallows, D., 2005 Search Engine Users, PEW Internet and American Life Project [online], http://www.pewinternet.org/pdfs/PIP_Searchengine_users.pdf [16 Feb 2007]

Fellegi, I.P. & Sunter, A.B. A Theory for Record Linkage, Journal of the American Statistical Association, 64(328),pp.1183-1210,December 1969

Fensel, D. 2001, Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce, Springer-Verlag Berlin and Heidelberg GmbH & Co. KG.

Fielding, R., W3C Technical Architecture Group mailing list, June 18, 2005.[online] http://lists.w3.org/Archives/Public/www-tag/2005Jun/0039

Genesereth, M. R., & Fikes, R. E, 1992. Knowledge Interchange Format Version 3.0 Reference Manual, Report of the Knowlegde Systems Laboratory (KSL), 91 (1), Stanford University.

Gruber, T.R., 1993, A Translation Approach to Portable Ontology Specifications, Knowledge Acquisition, no.5, pp. 199-220.

Guha, R. & McCool R., 2003. TAP: A Semantic Web Platform. *Computer Networks*, 42 (5), 557-577.

Halpin, H. Identity, Reference and Meaning on the Web, Proceedings of the Workshop on Identity, Meaning and the Web (IMW06) at WWW2006, Edinburgh, Scotland.

Harth, A. & Decker, S. 2005. Optimized index structures for querying rdf from the web. In *Proceedings of the 3rd Latin American Web Congress*. IEEE Press.

Hernandez, M. & Stolfo, S. Real-world Data is Dirty: Data Cleansing and the Merge/Purge Problem, Data Mining and Knowledge Discovery, 2(1), pp.9-37,Kluwer,Hingham,MA,USA

Hogan, A. Harth, A. & Decker, S. 2006. ReConRank: A Scalable Ranking Method for Semantic Web Data with Context. In *2nd Workshop on Scalable Semantic Web Knowledge Base Systems*.

Hogan, A., Harth, A., Umrich, J. & Decker, S.2007. Towards a Scalable Search and Query Engine for the Web. *In Proceedings of the sixteenth International Conference on WWW,*Banff, Alberta, Canada, pp.1301-1302, ACM Press.

Immaneni, T. & Thirunarayan, K. 2007. A Unified Approach to Retrieving Web Documents and Semantic Web Data. *In:* E. Franconi, M. Kifer, W. May, eds. *Proceedings of the 4th European Semantic Web Conference (ESWC 2007),* June 2007 Innsbruck, Austria. Germany: Springer, 579-594.

Jian, N., Hu, W., Cheng, G., Qu, Y.: Falcon-ao: Aligning ontologies with falcon. In: K-Cap 2005 Workshop on Integrating Ontologies. (2005)

Karger, D. R., Bakshi, K., Huynh, D., Quan, D., & Sinha, V., 2005. Haystack: A General Purpose Information Management Tool for End Users of Semistructured Data. 2nd Biennial Conference on Innovative Data Systems Research (CIDR 2005), 4-7 January 2005, Asilomar, California.

Kalfoglou, Y. & Hu, B.. Crosi mapping system (cms). In Proceedings of K-CAP 2005 Workshop on Integrating Ontologies, 2005.

Kifer, M., Lausen, G., & Wu, J., 1995. Logical Foundations of Object-Oriented and Frame-Based Languages. Journal of the Association for Computing Machinery (ACM), May 1995.

Lassila, O & Swick, R., 2002. RDF Model and Syntax Specification, (W3C Recommendation), [Online], Available from: http://www.w3.org/TR/REC-rdf-syntax [Accessed 5 July 2005].

Langville, A.N. & Meyer, C.D. 2004. The Use of Linear Algebra by Web Search Engines, Department of Mathematics, North Carolina State University.

Lei, Y., Uren, V.S. & Motta, E.. (2006) SemSearch: a search engine for the Semantic Web. Proceedings EKAW 2006, pp.238-245,Podebrady, Czech Republic.

Madhavan, J., Shawn, R.J., Cohen, S., Dong, X., Ko, D., Yu, C. & Halevy, A., 2007. Web-Scale Data Integration: You Can Only Afford to Pay As You Go.

ThirdBiennial Conference on Innovative Data Systems Research, January 7 – 10, Asilomar, California.

Mcguiness, D. & Harmelan, F. V. Eds., 2004. OWL Web Ontology Language Overview, W3C Recommendation [Online] Available from: http://www.w3.org/TR/owl-features/ [Accessed 4 April 2006].

Millard, I., Jaffri, A., Glaser, H. & Rodriguez, B. (2006) Using a Semantic MediaWiki to Interact with a Knowledge Based Infrustructure (Poster). *Proceedings EKAW* 2006, Podebrady, Czech Republic.

Notess, G.R., 2002. Search Engine Statistics: Unique Hits Report [online], Available from: http://www.searchengineshowdown.com/statistics/unique.shtml [Accessed 4 April 2006].

Noy, N. F., Fergerson, R. W., & Musen, M. A., 2000. The knowledge model of Protege-2000: Combining interoperability and flexibility. 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2000), 2000, Juan-les-Pins, France.

Passant, A., Bojars, U., Giasson, F. & Breslin, J. 2007. Smart Tools to Easily Dicover and Query Decentralised Semantic Web Data. *European Semantic Web Conference 2007*. June 4 2007. Innsbruck, Austria

Passin, T.B. 2004, Explorer's Guide to the Semantic Web. USA: Manning.

Raimond, Y. Linking Open Data: Interlinking the Jamendo and the Musicbrainz Datasets. [online] http://blog.dbtune.org/post/2007/06/11/Linking-open-data:-interlinking-the-Jamendo-and-the-Musicbrainz-datasets [13 August 2007]

Rodrigo, L., Benjamins, V.R., Contreras, J., Paton, D., Navarro, D., Salla, R., Blazquez, M., Tena, P., & Martos, I. 2005. A Semantic Search Engine for the International Relation Sector. *In:* Y. Gill, E. Motta, V.R. Benjamins, M.A.

Musen, eds. *Proceedings of the 4th International Semantic Web Conference (ISWC 2005),* November 2005 Galway, Ireland. Germany: Springer, 1002-1015.

Shadbolt, N. R., Gibbins, N., Glaser, H., Harris, S. and schraefel, m. c. (2004) CS AKTive Space or how we stopped worrying and learned to love the Semantic Web. IEEE Intelligent Systems.

Shafi, S.M., & Rather, R.A. 2005. Precision and Recall of Five Search Engines for Retrieval of Scholarly Information in the Field of Biotechnology. Webology, 2 (2), [online], Available from: http://www.webology.ir/2005/v2n2/a12.html [Accessed 20 April 2006].

Suchanek, F.M., Kasneci, G. & Weikum, G. 2007. YAGO:A Core of Semantic knowledge. Proceedings International WWW Conference 2007, ACM Press, pp.697-706,Banff,Alberta,Canada.

Sullivan, D., 2006. Nielsen Netratings Search Engine Ratings [online], Available from: http://searchenginewatch.com/reports/article.php/2156451 [Accessed 4 April 2006].

Tan, Y.F., Kan, M.-Y. & Lee, D. Search Engine Driven Author Disambiguation, Proceedings 6th ACM/IEEE-CS Joint Conference on Digital Libraries, pp.314-315,ACM Press, New York.

Tumarello, G., Oren, E. & Delbru, R. 2007. Sindice.com: Weaving the Open Linked Data. *In Proceedings of the sixth International Semantic Web Conference*, Busan, Korea.

W3C (2001) URIs, URLs and URN's: Clarifications and Recommendations 1.0,W3C Note, 21 September 2001. [online] http://www.w3.org/TR/uri-clarification [01 August 2007].

Yang, K., Jiang, J., Lee, H. & Ho, J. Extracting Citation Relationships from Web Documents for Author Disambiguation, Technical Report No.TR-IIS-06-

017,Institute of Information Science, Academia Sinica, Taipei, Taiwan, December 2006.