On Coreference and the Semantic Web

Hugh Glaser, Ian C. Millard, Afraz Jaffri, Timothy Lewy, and Ben Dowling

Dependable Systems and Software Engineering Group
School of Electronics and Computer Science
University of Southampton
{hg,icm,a.o.jaffri}@ecs.soton.ac.uk, timlewy@gmail.com, bmd102@zepler.org

Abstract. Much of the Semantic Web relies upon open and unhindered interoperability between diverse systems; the successful convergence of multiple ontologies and referencing schemes is key. However, this is hampered by the difficult problem of coreference, which is the occurrence of multiple or inconsistent identifiers for a single resource. This paper investigates the origins of this phenomenon and how it is resolved in other fields. With this in mind, we have developed and tested an effective methodology for coreference resolution in the Semantic Web at large. This framework allows the user to a) record identified instances of coreference in a usable and retrievable manner b) integrate new and existing systems for reference management, and c) provide a thesaurus-like consistent reference service capable of providing on-tap resolutions to interested applications.

Keywords: URI, Linked Data, Coreference.

1 Introduction

The emergence of the Semantic Web is, in essence, a move from a web of pages designed and published for human consumption, with no intention other than to be viewed by the human eye and parsed by the human brain; to a web of data connected by machine interpretable semantics, that when applied or used in a suitable context produces content or services useful to other semantic systems, agents or end users.

Instead of documents described in HTML and connected by hyperlinks the web becomes entities (people, places, things or concepts) linked by associations and described in RDF. The knowledge represented by the web is gathered by many parties for a multitude of purposes, from many different sources. It is to be expected for inconsistencies to occur between data gathered by different processes, which might undermine its usefulness. Frequently it transpires that some entities have multiple representations or references that are in fact equivalent to one another. For example "N. Shadbolt", member of the School of Electronics and Computer Science (ECS)

could well be equivalent to "Nigel Shadbolt", president of BCS. This phenomenon is known as coreference: when multiple references point to a common referent.

The central problem of coreference in the Semantic Web is due to the inherently distributed and disparate nature of the information. Whilst it is entirely conceivable that a single data source may have occurrences of coreference within it, this is the responsibility of the owners, as with any other database, to keep it clean and consistent. The main problem arises in cross-referencing, integrating and reusing data from multiple sources. This is facilitated in the Semantic Web through the use of URIs. In theory a single URI should be used for each resource so the information regarding it can be identified in any setting. For example, it would be helpful if William Shakespeare were universally referred to using a single URI. However, it is absurd to assume that the whole world can agree on a single identifier for everything that exists, anymore than the world agrees on single words for even the most commonplace objects.

At best it is only possible to create a unique identifier (URI reference) for a resource in a given repository. This would be sufficient for an application only working within that repository but would have little significance to the outside world. Currently this is exactly what is being done; many semantic applications use URI schemes with only local significance. For instance, within ECS, people are assigned URIs based on the departmental context, such as http://id.ecs.soton.ac.uk/person/4860. No effort is made to investigate possible pre-existing identifiers. Anyone attempting to gather data on ECS staff from a foreign application or with reference to another knowledge source, would have to resolve ECS URIs against whatever other reference schemes they happen to be using. The problem then becomes one of mapping locally identified entities to foreign ones.

1.1 Resource Disambiguation

Mapping equivalent references is an important challenge. As part of the Advanced Knowledge Technologies project [1], data on UK computer science research was gathered from a variety of sources and combined into a single knowledge base. In merging data from different sources, similar references arose. Searching the knowledge base for the string "Nigel Shadbolt" revealed some 25 separate identifiers potentially representing the same person. Simply performing a naïve comparison of attribute values was unsatisfactory especially if the values are just string literals. Looking just at the name attributes: "Hall W." is author of one paper, "Wendy Hal" is author of another, "Wendy Hall" is a head of department. All this information has to be reconciled. Names can be overloaded i.e. there could be two entirely different people called Wendy Hall, both of whom might have written research papers. Names are frequently incomplete or inconsistent: "Nigel Shadbolt", "N. Shadbolt", "N. R. Shadbolt" or "Shadbolt. N". Sometimes they are inaccurate e.g. "Nigel Shadblot" (as opposed to "Nigel Shadbolt").

The extent of the difficulty can be seen within the UK research community by analysing the RAE 2001 returns. Within the list of researcher names in the institutional submissions (which are recorded as initials and surnames on the HERO

website, www.hero.ac.uk) 10% of names lead to clashes between two or more individuals. If the names are restricted to a single initial, the proportion of clashes rises to 17%. Within our own institutional open access repository, records show that depositors typically give up to six different ways of naming any individual author (due to combinations of full names, initials and names that are incorrectly spelt). It has also been shown that in the DBLP bibliographic database, which is also exposed as Linked Data, 90% of authors with common names have URIs that are incorrectly merged together [2].

One must also remember that the Semantic Web is not a simple data source; it may be used to represent any knowledge and any concept, no matter how abstract. Whether two or more concepts are actually the same raises many difficult questions. There are at least 8 well-known people, a University and a Hospital that are called "John Hopkins"; clearly we cannot rely on comparing names. A large part of identifying whether two entities are the same is identifying that they are things of the same type. Within Semantic Web metadata, the possible entity types and connecting relations are specified in ontologies. At present these are often created for specific applications and are only occasionally reused. Therefore whenever data is combined from overlapping ontologies, seemingly equivalent types must be reconciled or mapped. The more abstract or indefinite the types are, the harder it is to be certain they are the same, making determining coreference between instances increasingly haphazard.

Coreference is not new. Whenever knowledge is recorded, coreference occurs. As such it is well documented in several fields, including linguistics, the main focus of which is resolving pronouns within sentences. The problem for linguistics and other domains is relatively straightforward (though not necessarily easy); however within the Semantic Web it is significantly exacerbated. This is due to three main factors:

1. **Open Authoring and Provenance.** As with the traditional web, information can be gathered and published freely be anyone with an internet connection. Unlike say, a book, this form of knowledge capture is highly prone to inconsistencies. In a book, multiple occurrences of "Nigel Shadbolt" could be assumed to refer to the same person. Indeed if they did not one would expect the author to highlight the issue. This is because the onus of ensuring consistency and decipherability lies solely with the author (and/or editor). There are likely to be many Nigel Shadbolts in the world and information in the Semantic Web could be regarding any one of them.
2. **Multi-Purpose and Context-free.** Knowledge does not naturally stand up outside of its context, yet this is required for information to be useful across the Semantic Web. If a paper has been published in multiple forms it is likely to be represented in the Semantic Web by multiple identifiers. We could well say that the things denoted by these identifiers are the same: They are the same text, with the same author and the same words. Certainly many applications would wish to treat it this way. However, they are different entities, published by different organisations in different formats. They will have differing metadata, different page numbers and different editors. This information would be incorrectly asserted to refer to a single entity. Clearly we must be careful about the context in which the information is

being used. A means of coreference resolution is needed that can handle the above application whilst leaving the structure of the data intact.

3. **Universal Representation.** The Semantic Web has the lofty goal of being a fully integrated web of machine interpretable knowledge. With the exception of blank nodes, all resources represented in the Semantic Web are assigned universal identifiers. Previously, databases and information sources were free to use whatever local naming scheme they wished and did not have to worry about interactions outside of their own systems. Now designers must employ identifiers robust enough to be used across the globe, without clashing with others denoting something completely different. So even if points 1 and 2 are resolved there is still an issue of adequate representation and identification.

1.2 Coreference and Linked Data

The production of the first tutorial on how to link Open Data [3] 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 [4].

The first set of data that is being used as a base for all subsequent data linkage is the DBpedia [5] 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 ontology 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*.

Since DBpedia has harvested knowledge from Wikipedia, there is the potential to create links to any subject that is described in Wikipedia.

The datasets that have been interlinked so far have knowledge relating to people, places, books, songs and CYC [6] concepts as well as many others. Entities such as these are often prone to the problems of duplication and co-reference.

Whilst extensive linking between datasets has been widely encouraged, there has been little analysis of the accuracy of the links or the datasets themselves. Datasets are often converted from existing sources which can themselves be either incomplete or inaccurate. The linking process accentuates these inconsistencies and produces a snowball effect as more datasets are added. If the Semantic Web is to provide a meaningfully interconnected web of assertions and relations, there must also be some guarantee or measure of the correctness of the information. This paper presents a solution for managing the consistency of data across different providers. Section 2

describes related work in the field, including projects that are trying to address the coreference problem. Section 3 looks at the problem of coreference in the Semantic Web in more detail. Section 4 presents our architecture for managing coreference and Section 5 describes an application built on top of this infrastructure. Section 6 concludes with some open issues and future work.

2 Related Work

During the early stages of the Web there were competing systems that were trying to provide alternative approaches for open hypermedia systems [7]. One such project was Microcosm which featured a selection and action link following paradigm and a message passing framework that was compatible with Web architecture [8]. The feature that we wish to highlight here is the separation of content and link information into a linkbase. The linkbase was a link database that contained all information about link availability within a document. The linkbase stored specific links, contained within a source document, and generic links which could be made from any document. The purpose behind the linkbase was to counter the early navigational problems on the Web, such as only being able to access pages by following a set of specific links or knowing an address beforehand and typing it into a browser. Even though the CRS architecture is substantially different from the linkbase model, the underlying idea of separating links from data to facilitate ease of use, remains similar.

The most recent project to offer a system of URI identity management is the Okkam project [9]. The architecture used in this project aims to mimic the DNS architecture of the Web. Instead of a DNS server, an ENS (Entity Name System) server or servers are provided that aim to create an environment of unique URI provisioning and usage. The ENS acts as a global repository of URI identification which searches for entities, adds new entities and issues new identifiers. The goal of the project is to have data providers use Okkam issued URIs for entities that exist in the system.

There are several reservations that we have with such an infrastructure. Firstly the analogy with the DNS system appears incorrect. The DNS is a hierarchical system that is used for finding the *location* of a particular resource. The Semantic Web needs a system for finding the *identity* of a resource, and the two are quite difference tasks. A postal address will tell you that person A lives at the given house, but how do I find out who person A is?

Secondly the issuing of identifiers by Okkam or what is referred to as the Okkamisation of entities will only add to the proliferation of URIs on the Semantic Web. When someone mints a new URI for a resource it is because they have knowledge about the URI that they wish to disseminate. There can never be a way of accurately determining that the Okkam URI is the same entity to which a knowledge provider wishes to refer. Furthermore, if someone wishes to use a DBpedia URI because they believe it fits their purpose, then the requirement for using an Okkam URI becomes a hindrance. This also leads on to the question of how the system will determine that a URI is the same as one in their system. Equivalence determination is

always prone to error and as already explained, URI similarity is subject to the context in which the URI is used.

The final and strongest criticism is that the ENS architecture is a centralised system which goes against the principles of Web architecture [10]. Furthermore, the creation and interaction between multiple ENS serves is not clear or explained in detail. Even though the ENS approach has many drawbacks, the project has given a lot of thought and consideration into the problem of URI coreference and should be applauded for giving the topic due importance in Semantic Web research.

An approach to identifying equivalent instances occurring across data sources has been used to perform object consolidation on the Semantic Web [11]. The algorithm looks for and uses inverse functional properties to detect instance equivalence and additional algorithms are used to describe how these equivalences are stored and ranked in memory. This work can be used to assist in the automated population of a CRS from crawling linked data URIs and pages. Since the major concern of any identity management application is the establishment of similarity metrics, this research provides one possible method to accomplish this task.

3 Coreference in The Semantic Web

There are several schools of thought when it comes to dealing with coreference in the Semantic Web. These largely fall into two categories: up-front approaches to defeating the problem and philosophies and principals to undermine or circumvent it.

Coreference is not purely a social problem; we cannot expect that metadata will simply converge on a set of agreed URIs over time. Looking at the usage of ontologies with the OAI-PMH protocol [12][13], we can see that even in a field with a de facto standard (Dublin Core), there are still over two hundred different ontologies in use. Clearly there are technical as well as social reasons for the existence of coreference, such as repositories trying to leverage information from legacy systems. Having said this, a solution that integrates both technical and social aspects is more likely to succeed. By involving the users of the Semantic Web, we massively decrease any one organisation or individual's personal responsibility.

3.1 Representation and Use

It is a first step to have mechanisms for matching equivalent identifiers to one another, but this is of little use without some way of applying these results to a semantic application. In many cases this is done through either an application-specific or manual process. For instance, the practice of "smushing" [14] has become relatively common. This generally involves merging the metadata associated with coreferent identifiers by reasserting the information so that every property relates to a single URI. Other similar methods involve bespoke solutions that identify references as being related without utilising any formal or established mechanisms.

By far the most common ontology in use is OWL. This allows the expression and exploitation of established coreference through the use of the owl:sameAs predicate,

which, according to the OWL ontology means that "two URI references refer to the same individual". This is a part of OWL's description logic. When used with a knowledge base capable of performing at least OWL-Lite inference, the predicate infers that the two URIs should be treated as though they were one. This has the same affect as smushing the two URIs, though without the need to reassert data: they become indistinguishable. Through our experiences and research we have come to the conclusion that this is not necessarily the best approach to use in most circumstances.

As argued above, the notion of identity is not as concrete as one might first think, somewhat undermining the semantics behind owl:sameAs. Such a strong assertion has serious connotations. It relates back to the notion of equivalence within context: with the exception of very elementary examples, one can only be sure that two URIs are equivalent within the confines of a specific application, whereas owl:sameAs asserts that two references are always the same. As Wittgenstein said, words only have meaning through use. The example of contextual equivalence in section 1 is an excellent example of when using the OWL solution is inappropriate. owl:sameAs should only be used when the two concepts being represented are utterly indistinguishable. This could occur as the result of an erroneous data mining process, when two URIs have been produced in identical circumstances and have an identical provenance and meaning. This was probably the true intention of the notation: to account for situations where the very existence of multiple URIs is the result of an error or poor initial knowledge.

To give another example of how not to use the predicate: It is possible that two different references both refer to the same person, but in different roles. For example, there may be one reference referring to "Wendy Hall" as head of school, and another referring to "Wendy Hall" as an author of a paper. The graphs associated with each reference may well contain different information, such as different email addresses or phone numbers. By asserting both references to be the same using OWL you can no longer differentiate one from the other and so in all further uses they would have to be treated as the same. This would make obtaining separate contact details or other specific metadata very difficult. In such a situation you would not want both references to be treated identically, even though in some sense they both refer to the same person. Theoretically one could carefully restructure the metadata into a form where all the information is preserved together with its context, but in many situations this is impractical as it would have to be performed many times. Frequently the application performing the resolution does not have the privileges or capability to rewrite data; it can only make its own assertions, as is the case with most agents. In this situation, restructuring the data would be impossible.

3.2 URI Multiplicity

The Linking Open Data project and our own ReSIST project [15] 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://www4.wiwiss.fu-berlin.de/eurostat/resource/countries/Espa%C3%B1a
```

These URIs come from 4 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
```

We have grouped these URIs 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 problems of using owl:sameAs have already been discussed in section 3.1. The semantics of *owl:sameAs* mean that all the URIs linked with this predicate have the same identity, this means that the subject and object must be the same resource.

We subscribe to the belief that the meaning of a URI may change according to the context in which it is used [16]. 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 some form of 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 our attempt to handle URI management, called the Consistent Reference Service (CRS).

4 Coreference Architecture

Now there are a range of available mechanisms for identifying and matching coreferences developing, it is an appropriate time to develop these systems into a more complete solution. Our solution architecture is composed of two parts: a method for effectively representing coreference and a communication mechanism, called a Consistent Reference Service (CRS) that provides a thesaurus-like medium for publishing mappings. This involves no new technology and as such is as extensible as the hardware it runs on. It can be deployed on a range of scales from personal to international. The framework that achieves this is described in the next section.

4.1 Bundle Framework

Our framework is designed to both annotate and communicate instances of coreference in a more efficient and flexible manner than using OWL. This is achieved by providing lightweight inference-free mechanisms with clear semantics. Collections of coreferent references are collated into sets, called bundles, so that each bundle contains references to a single resource. Without the complications of inference, the bundles can be searched for and handled explicitly. Multiple bundles may be used to represent a resource for different uses. For example, "Nigel Shadbolt" might have one bundle for references to him at ECS and another for references to him at the University of Nottingham. An application could then opt to use one, both, or neither bundles. Looking back again to the example in Section 1, the problem would be solved by having one set of bundles for when papers need to be identified in different publications and another set for when they need to be identified as single bodies of academic work.

Bundles may be used as a convenient method of communicating references between systems. By passing whole bundles between applications, systems can share information on coreference in a way that OWL could only achieve with the help of expensive inference.

Bundles are a method of coreference representation and not a solution to the problem on their own. However, they are an effective means of collating mappings. They are essentially sets to which equivalent and non-equivalent references may be added and removed at will. An added bonus of this is that a form of set calculus can be performed upon them. If two bundles are found to represent the same entity and usage, the union of their members can be used to perform a simple merge. If two bundles represent different usages, the union can be used to obtain references regardless of certain contexts, such as references to Nigel Shadbolt at any Institution. Likewise, the intersection of two bundles may be used to obtain only the resolutions applicable in both contexts.

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 look like:

```
<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.

4.2 Usage and Social Engineering

A system that allows coreference information to be easily queried-for could be employed in a number of scenarios. In our early experimentation, we employed CRS servers at an institutional level; our server provided a source of mediation between all the different identifiers used within the University of Southampton. At Southampton we publish our academic output openly through a software package called EPrints [**Error! Reference source not found.**], this creates a lot of metadata and a lot of instances of coreference. By providing a central point of mediation, combined with existing mechanisms for mapping identifiers, it was significantly easier to develop semantic applications. These provided new and interesting services upon the data. A lightweight plug-in was created for the EPrints software that significantly enhanced its use by leveraging the CRS' services [18].

How the CRS is socially integrated is important to its success. Our preliminary use of a CRS server is effective for situations where there is a clear central point of administration and responsibility, such as within a University. On the larger Semantic Web, the responsibility for content is divided amongst all the users. Here CRS servers could be run by institutions that would benefit from them, such as a car manufacturer publishing all the references to their cars, or a consumer watchdog site publishing references to reviewed products. Alternatively third parties will choose to offer CRS services of varying quality, possibly charging for good services.

An additional mechanism would be a CRS coreference cache held by agents. A personal agent would hold a record of the different URIs for entities it commonly

handles, such as ones for its owner and their interests. For instance, the agent in the example given by Tim Berners-Lee would hold a bundle for its owner, for the treatments and treatment centres that it has come across and for other agents and persons that it frequently interacts with. This would be built up over time; agents communicating with each other could share bundles relevant to their interactions, allowing them to operate without the need to constantly refer to larger coreference sources.

4.3 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. We have developed an underlying information infrastructure that utilises the CRS architecture given in Section 4. Our 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.

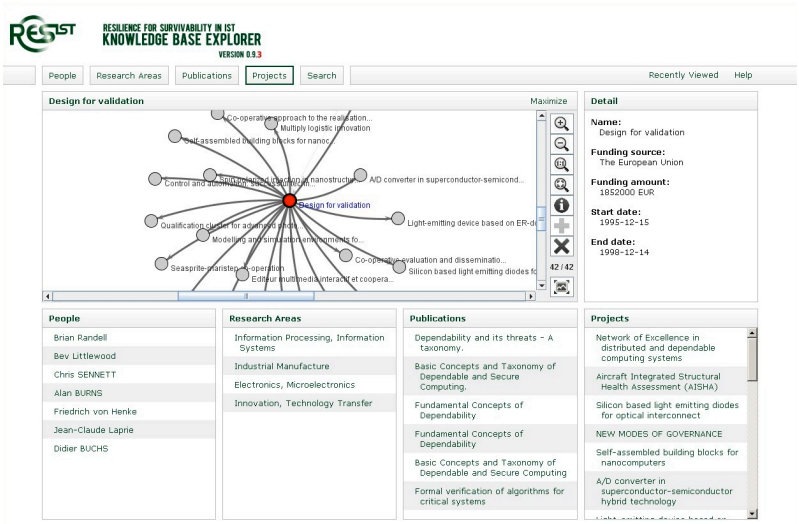

Fig. 2. The figure above shows the single window interface of the faceted browser available at http://www.rkbexplorer.com/explore/

Figure 2 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 RKB Explorer is based on the implementation described in Section 3, and provides a unified view of more than 20 triplestores, where the coreference information is supplied from the multiple associated CRSes that manage 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.

In terms of performance, we have found that the response of the CRS system is satisfactory, as long as the underlying triplestore or SQL DB is reasonable; most systems are able to look up a URI and return the bundle in a time that is almost independent of the number of bundles. For example one of the CRSes we have (concerned with the DBLP data) has approx. 1.4M bundles with almost 4M URIs, and the RKBExplorer can use it as one of its many CRSes. Because the application only needs to query a subset of the CRSes, performance is not sensitive to the existence of other coreference data.

5 Conclusion

There are several issues that arise when implementing the above methodology. Firstly, the difference between this approach and using *owl:sameAs* must be highlighted. As noted in the introduction the semantics of *owl:sameAs* are very strict and it is debatable whether the two Eurostat URIs should be *owl:sameAs*. The other consideration is of Semantic Web applications that must always load the data of each URI that is *owl:sameAs* the current URI. This limits performance and imposes unnecessary loading of data. The CRS architecture allows for following as many, or as few duplicate URIs as required with no significant barrier on performance. It is not our intention to remove *owl:sameAs* from linked data, rather we would definitely encourage its use in situations where the semantics of the relation are correct.

The second issue that arises is how the URI synonyms are acquired. In our prototype application the CRSes created for each dataset were made with datasets of links that were already made available on the Web. It is simple a case of putting the

same URIs that would be linked using *owl:sameAs* into a separate knowledge base. There is plenty of work needed in developing linking algorithms for detecting URI equivalence. The CRS system is envisaged to utilise these algorithms and provide links in such a way as to preserve URI equality without establishing the formal semantics of an *owl:sameAs* relation.

Another issue arises over which CRS contains which duplicate URIs. The example above uses URIs that are randomly distributed amongst the CRSes. It is entirely possible for one CRS to contain all equivalences of a URI, thus reducing the work needed to find the full equivalence set. However, the more common scenario is that data providers will not be aware of every single synonym for their URIs and hence there is a need for multiple CRSes. As an example, we can look at the current DBpedia data for Portugal which does not contain all URI synonyms in the form of *owl:sameAs* links.

The CRS is designed to be a service that can be used by semantic applications as a source of coreference resolution. An application may look up a reference it knows about and discover other URIs that correspond to the same entity. The CRS achieves this by storing and making available established mappings, freeing individual applications from the need to develop their own costly resolution systems. The mappings stored by the CRS can be contributed by anyone and it is expected that existing resolution systems will be connected to it.

Coreference within the Semantic Web is a growing, yet unappreciated problem, at least until recently. It has been suggested that it is a matter that will resolve as the Semantic Web evolves, with careful social engineering and planning. However, having performed a detailed study into the nature of this problem, investigating its occurrence not just within the Semantic Web but in other fields as well, we consider that the problem cannot be avoided. When looking at its appearance in related fields such as data warehousing and Artificial Intelligence, it becomes immediately obvious that the nature of the Semantic Web causes coreference to be systemic and prevents any existing solutions from being transferred.

It is our conclusion that the most effective means for combating the issue is to make coreference-awareness an architectural feature of future semantic applications.

In support of this finding and in anticipation its requirement, we have designed and proposed the methodology and framework outlined in the latter half of this paper. Use of the bundle framework provides a flexible, expandable and readily compatible notation for recording and managing coreferent identifiers. This, combined with the CRS system, provides a broad strategy for coreference resolution that integrates the process of reference management into the architecture of the Semantic Web by utilising both social and technical engineering.

6 Acknowledgements

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

References

1. AKT, "The AKT Manifesto". Technical report, 2001. http://www.aktors.org/publications/Manifesto.doc
2. Jaffri, A, Glaser, H. & Millard, I. URI Disambiguation in the Context of Linked Data. In Proceedings of the 1st Workshop on Linked Data on the Web at WWW2008, Beijing, China.
3. 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]
4. 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.C. Lewy, *Meaning and Modality*, Cambridge: Cambridge University Press, 1976.
5. DBpedia [online] http://dbpedia.org/docs [1 July 2007]
6. Cycorp Inc. http://www.cyc.com
7. Davis, H.C., Hall, W., Heath, I., Hill, G.J. & Wilkins, R.J. Towards an Integrated, Information Environment with Open Hypermedia Systems. In Proceedings of ECHT'92, ACM Press, pp 181 - 190 (1992).
8. Carr, L., Hall, W., Davis, H. & Hollom, R. The Microcosm Link Service and its Application to the World Wide Web. In Proceedings of the 1st World Wide Web Conference, Geneva Switzerland, May 25-27, 1994, ACM Press.
9. Bouquet, P., Stoermer, H & Giacomuzzi, D. OKKAM: Enabling a Web of Entities. In Proceedings of the 16th International World Wide Web Conference (Banff, Canada) ACM.
10. Jacobs, I. & Walsh, Norman. Eds. Architecture of the World Wide Web, Volume One, W3C, [online] http://www.w3.org/TR/webarch/ [10 March 2008]
11. Hogan, A., Harth, A & Decker, S. Peforming Object Consolidation on the Semantic Web Data Graph. In Proceedings of the Workshop on Identity, Identifiers and Identification at WWW2007, Banff, Canada, 2007. ACM Press.
12. Academic Contributor Information System Project, http://acis.openlib.org/, 2006.
13. T. Krichel and I. Kurmanov, ACIS Stage Three Plan, http://acis.openlib.org/stage3/, 2005.
14. RDFWeb: FOAF Developer site Wiki, "smushing", http://rdfweb.org/topic/Smushing, Accessed 15 May 2007.
15. Resilience for Survivability in IST (ReSIST) Network of Excellence. http://resist-noe.eu
16. Booth, D. URIs and the Myth of Resource Identity, Proceedings of the Workshop on Identity, Meaning and the Web (IMW06) at International World Wide Web Conference 2006, Edinburgh, Scotland.
17. C. Gutteridge, "GNU EPrints 2 Overview" in *Proceedings of 11th Panhellenic Academic Libraries Conference*, Greece, 2002.
18. T. Lewy, "A Consistent Reference Service for the Interoperation of EPrint Repositories" Technical Report, School of Electronics and Computer Science, University of Southampton, 2006.