On Coreference and the Semantic Web

Hugh Glaser, Timothy Lewy, Ian Millard, Ben Dowling

School of Electronics and Computer Science
University of Southampton, UK
{hg, icm}@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: Online Information Systems - Web-based services, Software Engineering - Interoperability, Reference management, Coreference, Web services

1 Introduction

1.1 Coreference

The emergence of the Semantic Web [1] 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 [2]. 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, 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 Universal Resource Identifiers. 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[1]; 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] Some unique concepts may have possible universal naming schemes, such as books and ISBN numbers or elements and atomic numbers. However, in the vast majority of cases this is not possible and even in these cases, there are many difficulties.

This activity is at the heart of the recent activity inspired by Tim Berners-Lee's note on Linked Data [3], where Principle 4 says: "Include links to other URIs. so that they can discover more things". See the Linked Data Initiative [4] for further details.

1.2 The Difficulties of Resolution

Mapping equivalent references is an important challenge. As part of the Advanced Knowledge Technologies project [5], 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 and is unlikely to be, 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).

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 [6]. These are generally 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. This is explored further in later sections. 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, some of which have been touched upon already:

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. I say this is lofty as it requires every item of knowledge to be somehow qualified against every other item. 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.

1.3 Implications

The issue of coreference within the Semantic Web is crucial. Take, for example, Tim Berners-Lee's Semantic Web agent [1],[7]. It is given the task to look up a patient's personal information, find their prescribed treatment and then present to the user an appointment at an appropriate clinic, at a time when they are available. There are many different knowledge sources involved here: The patient record, a register of clinics, the clinic's appointment system and the person's scheduler. From the outset the agent will have to do a lot of work to achieve its goal: The ontology for describing treatments in the patient's records might well be different to that used by the clinic registry, or the clinic appointment system. The three different source ontologies would have to be merged, or at least mapped before the agent could operate between them. This might be in the form of a service available to the agent, or it might be done on the fly [8].

Once mapped, our problem of referential inconsistencies and coreference is encountered. The patient record system and the clinic registry, whilst possibly using similar classes for treatments in their ontologies, may not have used the same URI for identifying the treatment in question. Likewise identifiers for the patient, locations and scheduling details will have to be mapped. The agent cannot work without resolving this problem.

Mechanisms for mapping coreferences are beginning to emerge, frequently for particular types of URIs. Ideally a solution is required that works in any situation, with any semantic application. This is not an easy goal and will require more than just clever matching systems; the solution must become integral to either the semantic applications running on the web or with the infrastructure of the web itself. This will require existing techniques to be set into a larger social and system infrastructure.

This paper investigates the origins of coreference and its evolution into the world of computing. With a full picture of coreference in mind, a flexible methodology for applying current techniques to more effectively tackle coreference is proposed.

2 Identity and Meaning

The details of how we create and use metadata is inextricably linked to ontology, the study of being and cognition. It is sometimes said that metadata in the Semantic Web aims to represent things and concepts. What it actually aims to represent is knowledge and what it actually does is capture linguistic prepositions purportedly pertaining to knowledge. When Descartes said "cogito ergo sum", "I think therefore I am", he had realised that all he could be certain of was that he was a thinking being. For what we assume to know about the world is the product of fallible senses and thus always subject to doubt. Cognition is the process of obtaining knowledge from sensation and we must not assume that the product of this process is discrete assertions that are perfect for annotating. Even the simplest facts are human constructs that may fail under scrutiny.

When looking at and designing systems for resolving coreference there are two crucial semantic pitfalls that affect how we tackle the problem. Firstly we must ask: how is it that we define, or represent something? And secondly what does it mean for two things to be the same, or different? The following sections illustrate the difficulty in answering these questions with respect to coreference identification.

2.1 Meaning and the Philosophy of Language

As mentioned, what we describe in RDF in the Semantic Web is not facts and truths about the world around us, but linguistic expressions that in turn attempt to describe some knowledge about the world. Asserting "Bill hasUncle Bob" does not necessarily entail anything about Bill's genealogy. All it asserts is that the creator understands that two expressions "Bill" and "Bob" are related by some predicate "hasUncle". Anyone else reading the statement that does not understand any of the three components will find it meaningless. Casimir Lewy explained this with an example [9]: reader X can only comprehend the statement "the concept of a Vixen is defined as the concept of a Female Fox" when four requirements are fulfilled:

- X understands the concept of "A Vixen"
- X understands the concept of "A Female Fox"
- X understands the expression "is defined as"
- X understand the syntax of the statement

When any one of these does not hold, or if our understanding of any one differs, the statement is useless. This rings true for the example, as "Uncle" is sometimes used to encompass family friends (to avoid children calling adults by their familiar name alone), which is a quite different definition. It is virtually impossible to know for certain whether any two seemingly similar expressions from different sources are identical. This makes coreference resolution appear futile; however there is another property of the Semantic Web that can help us: all statements are both made and understood

with a specific use in mind[2]. Wittgenstein held that words are only defined by their use or effect in what he called language-games. To give an example: a child does not learn the word cookie by seeing a cookie and attributing the sound to it; he learns the meaning from the game that if he says "cookie" he will be given a round biscuity object [10]. He demonstrated this with a mental exercise: try to imagine a definition for the word "game". Any definition you come up with will be in some way flawed. If you say it is a form of entertainment, then how do you reconcile sportsmen who compete as a profession? If you define it has something related to competing, how do you explain solitaire? It is not that these definitions are incorrect, simply that they are only correct for a subset of instances or under a certain set of circumstances. This does not render the word meaningless, as it does not matter that we cannot define a word, so long as we are able to use it. This is exactly the case in the Semantic Web. If we try to define a class so rigorously that it will stand up in any situation, we will fail. All we need to be able to do is identify classes that we can use successfully for a specific application. Therefore our mappings between references should be handled at the application level, with notations to maintain the circumstances under which they were established and are known to hold.

2.2 Identity

"[The world] consists of "stuff" spread more or less unevenly and more or less densely around space-time… within our own "conceptual schema," the stuff occupying one spatiotemporal region will be taken as constituting a thing, while the stuff occupying another such region shall not" [11] In other words all concepts of the identity of things, people, places or entities are a product of our own cognition. The concept of identity is not founded in the physical world and therefore attempting to represent the physical through the identification of things can lead to difficulties.

With this in mind, identity is said to be the relation that an entity has with itself. It is the sum of an entity's properties that distinguish it from all others; that make it unique. This is the same notion as that of things being the same or different. Two things said to have the same identity, the same distinguishing features, are considered a single entity [12]. The nature of what features can be considered to distinguish something is the source of some conjecture. The classic example is that of the ship of Theseus. In the legend, reported by the Greek biographer and philosopher Plutarch, upon returning to Athens, Theseus' ship was preserved.

"The ship wherein Theseus and the youth of Athens returned had thirty oars, and was preserved by the Athenians down even to the time of Demetrius Phalereus, for they took away the old planks as they decayed, putting in new and stronger timber in their place." [13]

Philosophers of the time (and indeed today) debated whether the ship remained the same ship, even once every piece of material had been replaced. This has given rise to two notions of identity: qualitative identity and quantitative identity [14]. Qualitative identity takes some account for varying levels of sameness and some form of emotional sameness: Two things need only share some properties to be considered the same. So a Poodle and a Great Dane are identical in so far as they are both dogs. The latter is a more empirical definition that states that two things are the same if and only if they share every property and attribute. This is encompassed by the work of Gottfried Leibniz and is known as his principal of the identity of indiscernibles [15].

The obvious implication here is that the ship of Theseus was not the same; in fact the moment a single plank was replaced it ceased to be the same ship. However, this also implies that nothing ever remains the same: A person has a different identity the moment a single cell is shed from their skin. Clearly there is some deficiency in this approach. A theory to address this whilst maintaining Leibniz's rule is the concept of four-dimensionalism [16], in which properties can be asserted as temporally dependant. Existence is divided up into discrete time slices so that while the ship may have different properties from one slice to another, over the period of its existence it can be considered to be the same. Things that change, such as a new mast, are asserted with respect to time: Theseus' ship has mast X in AD80 and has mast Y in AD100. A serious problem with this approach is that there is no "correct" way to divide time.

If the old, replaced, timbers were used to build a second ship, which ship would then be the same as the original? Physically the second ship is composed of all the materials from the original and so is identical to it; but if we had previously taken the new ship to be identical to the original this would infer that the two entirely separate ships sat in dock are the same entity. We could go on, there are no easy answers. A pragmatic approach, which may be more relevant to our interests, is to turn again to Wittgenstein and say that whether two things are the same is dependant upon the use of the word "same" and therefore the purpose of the question. For instance, if it were a question of legal ownership, where ships are identified by a frame number, the original ship would be the one possessing the original mark [17]. Whereas the people of Athens might consider the ship that they have carefully maintained and looked after as being the true ship of Theseus. Neither is right or wrong.

This is a single example of the ambiguity of identity. Many others exist, such as: what is it to be a "person"? Is a person considered to have the same identity if they have changed both physically and mentally? [18] The message to take away is that we must not be blithe with what we declare to be coreferent; we should account for situations where there may be more than one answer.

---

[2] A single piece of metadata may be applied to multiple uses within its lifetime. This does not affect the point being made as the initial purpose is irrelevant so long as one can identify whether a given piece of information is applicable to a specific application.

3 Related Fields

Coreference is a common topic within many knowledge-based sciences. Each field has, largely in isolation, found its own ways to combat the problem and undoubtedly a lot can be learnt from these previous approaches. This section investigates occurrences of coreference and identifies how solutions have been engineered to the problem.

3.1 Linguistics

Linguistics is the field that first coined the term coreference. It is the most obvious setting for it to occur. Within natural language we automatically perform mental coreference resolution. When this process fails, we fail to understand the sentence. Most commonly coreference is performed in the presence of pronouns and anaphora. Anaphora is when a word is used to refer back to another word that occurred previously, such as *it* and *do* in "I know *it* and he *does* too" [19] In the sentence "I saw Bill today, he was jogging" the words "Bill" and "he" are most likely coreferent, as they probably refer to the same person. Here the resolution might seem obvious, but this is a very simple example and the process is an innate Human ability. Instructing a natural language processing system to do the same task is a very tricky problem.

"Since he(1) hit him(2) on the head before he(3) had the chance to say anything, we'll(4) never know what the lecture was supposed to be about" [20]. This is a more complex example, but again, it may seem obvious to us that (2) and (3) are coreferent. The rules dictating this are incredibly complex, but they do exist. Within linguistics coreference is deterministic such that two noun phrases are either coreferent or non-coreferent. There is only ambiguity when the language is used poorly. This makes linguistic coreference quite different to that in the Semantic Web, where it is not a question of grammar and syntax within documents, but of cross referencing between highly diverse sources. So although the problem is shared between the fields, we cannot take inspiration from the linguistic approach.

3.2 Artificial Intelligence

The structure of the Semantic Web is essentially a web based semantic network. Semantic networks are a form of knowledge representation that has been used by artificial intelligence systems since the 1960s. Typically they are used to represent information comprising a knowledge base to some larger application, such as an expert system or a natural language processor. On appearance they look very similar to the structures in the Semantic Web, but due to the nature of their use they do not suffer from the coreference problem[3]. Semantic networks are generally not distributed; a single network will be employed for a specific use and will have a single schema that will be maintained by a single party. Therefore it is possible to make what is known as the unique name assumption. Quite simply, this means that all nodes in the graph are assumed to be uniquely identified. Within a discrete graph this is quite easy to do, in the same way that it is possible to find locally unique URIs for resources in the Semantic Web. For the same reasons that one cannot guarantee a globally unique and consistent URI, one cannot apply the unique name assumption in the Semantic Web. To do so would require that every URI be validated against every other and that every entity be verified as disjoint from all others before use.

3.3 Databases and Data Mining

Within the field of database management and data warehousing, coreference is manifested as two problems: schema mapping, which is similar to ontology mapping, and data cleaning.

Schema mapping occurs in scenarios such as data warehousing, when entries from multiple sources that use different database schemas are merged into a single database. It is a practise that has obvious similarities to merging and mapping ontologies in the Semantic Web. However, there are key differences that make it more straightforward. Within a data warehouse, the application of the different schemas is generally known beforehand and, in most cases, schemas to be merged are representing the same information. It is rare that schemas representing very diverse information would want to be combined, as the aim is to combine large quantities of homogeneous information in order to extract useful patterns rather than to correlate diverse information to infer new knowledge. Schema matching systems therefore only have to map classes within discrete sets. There is also a much higher probability of two similar classes being a match, as there are not the subtle semantic differences present in the Semantic Web. This allows the schemas to be matched using relatively straightforward algorithms, known as match operators, which include techniques such as structural graph matching, element-level text comparisons and entry pattern identification [8]. The choice between these algorithms depends upon the nature of the schemas and the amount of information available, such as whether instance data is present. It is also worth noting that schema matching is usually an offline process than can be performed by systems with significant processing power; there is no need to develop lightweight approaches that can be executed on the fly.

[3] Semantic networks used for natural language processing may well deal with coreference, but this coreference affects the linguistics, not the network.

Ensuring data consistency is the act of making sure that there are no individually coreferent entries within a database, such as two entries for the same person in a customer database. In many areas of database maintenance it is only necessary that there are few coreferent entries, rather than none, as would be ideal in the Semantic Web. When the data is only required to identify patterns, small instances of coreference are unimportant in comparison to the larger picture. When coreference resolution is important, the techniques used generally rely on application-specific heuristics or keys. As discussed, qualitative identity only requires a selection of properties to be the same; a key is a selection of properties that have been established as sufficient for discriminating entries within a specific table. For example, in a customer database, entries with the same customer number, or with the same combination of name, date of birth and address could be considered to be the same. Many databases have the well-known ability to automatically restrict keys so that no two entries are allowed to clash.

3.4 Library and Information Science

The field of information science is very close to the form of knowledge management present within Semantic Web circles. Indeed, the histories of the two fields are intertwined, as can be seen by examining of the first draft of the RDF model [21].

The field covers the study and management of metadata from libraries and other collections such as museums. Within this setting metadata is highly controllable. Whilst there may be a relatively large set of data, it is usually capable of being represented with a single ontology. What is more, all knowledge management efforts are focused inwards as there is no requirement to make metadata interoperable with that of non-associated institutions. As such, information science is able to apply a technique that Semantic Web researchers cannot: A Controlled vocabulary.

A controlled vocabulary is somewhat akin to an ontology, only it goes further. As well as dictating the classes and possible attributes, the vocabulary dictates all the possible values as well. The available values and their meaning are especially designed and highly engineered to avoid inconsistencies. Every term has exact, discrete semantics; there are no synonyms or homographs [22], eradicating a large source of coreferences.

For instance, the Library of Congress [23] controlled vocabulary specifies a range of subject headings that different literatures may use to describe their contents. When all books are described to be members of one or more of these terms, there can be no ambiguity as to what topics they belong to or what those topics denote.

To give another example, a vocabulary describing a collection of films will have predefined possible directors such as “Stephen Spielberg” or “Francis Ford Coppola”. Where there are multiple directors of the same name, they may be disambiguated using a birth date or an index number.

Using such as system, the only time when there is the risk of coreference occurring is when data is imported from an external source that does not adhere to the same vocabulary. Then, the information scientists have discovered [24], even if both sources are of a high quality the product of the two is likely to suffer. This is due to the need to map between the two vocabularies (a familiar scenario), but this is an infrequent task and having performed it there is generally no need to worry about a library’s interoperability.

As and when libraries become semantically enabled, they will inherit many of the problems from the Semantic Web. The system described in the later sections of this paper would form an ideal thesaurus for mediating between different vocabularies.

4 Coreference in The Semantic Web

There are several schools of thought when it comes to dealing with coreferences 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.

Means of avoiding coreference include suggestions such as enforcing a unique name assumption, similar to the AI approach, or by reinforcing social structures to limit the problem’s impact. We consider the former suggestion to be largely impractical, as it would be quite impossible to implement. Enforcing a scheme such that every URI in the world pertains to a single entity and is the only entity that it refers to would, apart from the obvious technical impediments, completely impede any deployment of the Semantic Web. Such a system would require everything authored to undergo a strict validation process, making any form of open or rapid growth impossible. It has been shown that it is also impossible to universally differentiate some concepts. Many URIs would end up being arbitrary and counter intuitive.

Arguments for increased social engineering are in the right direction, but only look at part of the picture. 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 [25][26], we can see that even in a field with a de facto standard (Dublin Core [27]), 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. Section 4.1.2 is an example of a partially social solution.

4.1 Identity Mapping Techniques

Several techniques for performing coreference resolution in the Semantic Web have been proposed. These generally trace their roots back to data mining mechanisms, though most try to take advantage of the unique data structures and information available through RDF, OWL etc. Two very different examples of mapping systems are described below.

4.1.1 RDF Graph Matching

One interesting technology for resolving equivalent references from a set of candidates is a form of graph analysis known as communities of practice (CoP) [28]. A community of practice is a "group of people connected by a shared interest in a task, problem, job or practice" [29]. In the context of the Semantic Web, this can be viewed for a given person as other people who are connected to a large number of things that the given person is also connected to. By obtaining the CoP for the members of sets of potential coreferences, or individual entities, we can derive a measure of similarity from the degree of overlap between CoPs. When this measure is above a threshold level, the sets of coreferences or individuals in question are likely represent the same entity, when combined with textual matching. A tool, ONTOCOPI [30], has been developed for calculating CoPs and has been tested as a component part of a system for coreference resolution [31]. A system was proposed for resolving coreferences that integrates mapping and populating ontologies from multiple, possibly legacy, sources. A CoP system could well be integrated with the framework proposed in this paper and would provide a desirable degree of automation.

4.1.2 Social Engineering - ACIS

The Academic Contributor Information System (ACIS) [32] is a novel system that is being developed to aid academics in maintaining an online profile and CV. It achieves this by providing incentives for academics to match coreferences themselves. Whilst it is not strictly a Semantic Web application, it does demonstrate an interesting solution to the same problem. ACIS harvests information from EPrints repositories with the aid of a purpose built plug-in [33], which generates metadata whenever the repository is updated. It stores this data in its own database.

The onus for performing linking and identity mapping is placed entirely on the academic themselves. If they wish to participate and maintain a profile, they must register with the service and provide basic personal metadata [34]. From this metadata the system performs heuristic text searches for documents and institutions that link to similar authors; the user is presented with a list of possible matches and is asked to select those that relate to them. Selected items are then added to their metadata.

ACIS utilises an author-identification plug-in for EPrints to keep registered academics' profiles up to date. When depositing documents into a repository, on entering author details, the depositor is presented with a list of matching authors present in the ACIS knowledge base. If the depositor chooses one, the document is directly added to that author's profile. In this way documents do not become disassociated from their authors and hopefully additional coreferences will not occur. The depositor only has to perform a minor additional task to achieve this.

All linking of documents to authors is kept within the ACIS database; unfortunately the information cannot be reused for other purposes. Using this system each document has to be individually linked to its author, which for a large knowledge base, is a comparatively labour intensive process.

4.2 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 practise of "smushing" [35] 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 system in use is The Web Ontology Language, OWL [36]. 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 in 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 1.2 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 are likely to 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.

5 Coreference Architecture

Now there is a range of available mechanisms for identifying and matching coreferences in existence, 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 CRS server and its experimental application at the University of Southampton are described in section 5.2. The framework that achieves this is described in the next section.

5.1 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 1.2, 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.

Bundles are metadata structures in RDF and OWL, and have the following features:

- Each bundle contains a set of references that are believed to refer to the same resource under a given set of circumstances. The predicate "hasEquivalentReference" is used to denote this.
- A bundle may contain a second set of references that explicitly do not to refer to the same resource as the first set. This is achieved using "hasNonEquivalentReference" and does not imply anything about what the references do refer to; just that the do not represent the same resource as the bundle. We found having non-equivalent references a useful, as it often takes as much work to ascertain that two references are not the same as it does to ascertain that

they are. It is therefore important to record this knowledge. For example, when there are multiple references that refer to two people with the same name.

- There is an optional allowance for explicit context. A bundle is only said to be applicable within a specific context. If bundles from multiple contexts are stored together they may be differentiate by a BundleContext element. This is connected using the predicate "hasContext".
- A single reference in each bundle may be marked as canonical with the predicate "hasCanonicalReference". This is used to indicate a preferred reference to be used in new assertions regarding this entity, in this context. It is an optional addition but may be useful in consolidating the number of different URIs being used.

The example in Fig. 1 shows an RDF graph for a sample bundle. In the diagram a bundle has been constructed to collate references to "Hugh Glaser". As can be seen, two references have been found that refer to him, one of which has been chosen as canonical. Additionally, the date on which the bundle was last updated has been recorded and a reference to a "Henry Glaser" has been identified as not being the same person.

The diagram suggests a method of assigning URIs to bundles: the checksum of the combined URIs for the canonical reference and the context is appended to the base URI of the knowledge base it came from. This would always be unique; two bundles with the same URI, using this scheme, would become correctly merged.

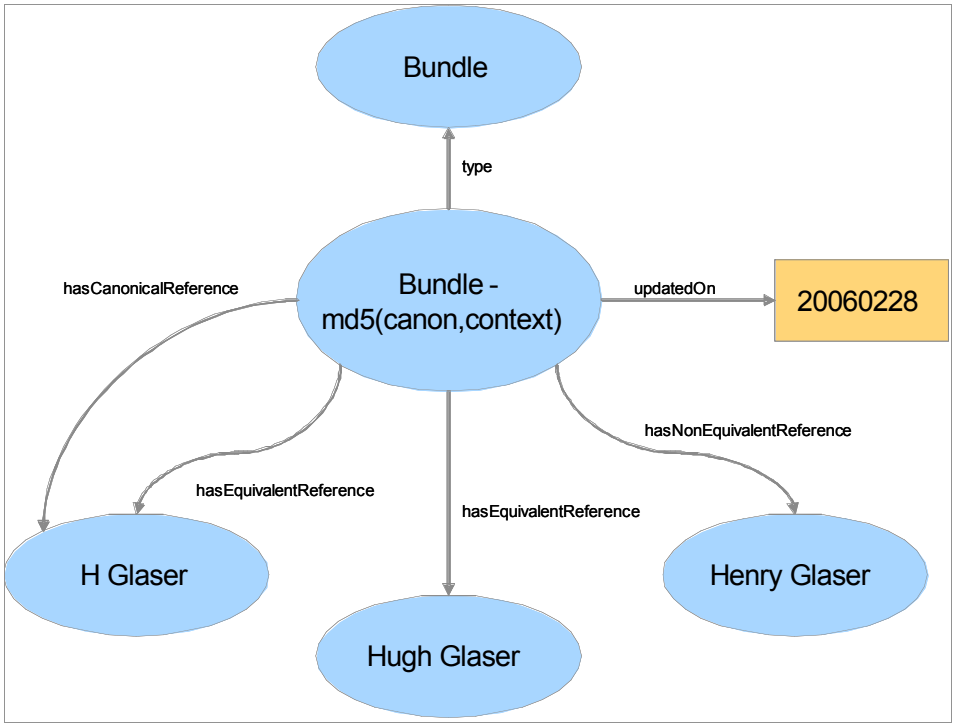

Fig. 1. Example Bundle RDF Graph

5.2 Consistent Reference Service

The Consistent Reference Service, or CRS, is designed to be a thesaurus-like reference 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.

5.3 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 [37], 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 [38].

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.

5.4 Services

We have built a CRS implementation that operates by providing a range of web services for performing various operations. These include services for retrieving, uploading and establishing equivalences between existing references. The services can currently either be used via provided web interfaces, or can be invoked via the REST protocol. This allows them to be more easily integrated into scripts, or as a basis for other scripts. A breakdown of the system's functionality is presented in the following sections.

5.4.1 Import

The CRS employs its own knowledge base that can be optionally populated with metadata relating to the URIs contained within. This allows bundles to be retrieved based on metadata searches and enables the CRS to provide this functionality independently from the original sources.

URIs that are to be mapped within the CRS are supplied via the import interface. This service processes new URIs by adding them to their own initial singleton bundles. Having done this they are ready to be mapped to other references. The URI import service can be operated by supplying the URI for a new reference or can be instructed to acquire new references by connecting directly to a remote knowledge base. It takes as input arguments the URI for the reference, an optional URI to a SPARQL [39] endpoint for the remote source and the name of the knowledge base that the URI is contained in. The service adds the URI to the CRS knowledge base and attempts to retrieve metadata from the SPARQL endpoint.

5.4.2 Update

The update service is provided to establish individual instances of coreference between references. Conceptually the interface is provided on a reference to reference level but underneath it operates on the bundle level. It takes two URI references as inputs, finds the bundles that they belong to and merges them. When a user asserts that two references refer to the same entity it can be inferred that all the other references bundled with them also refer to that single entity. Merging the two bundles achieves this result. The service can also be used to remove coreference information about a reference. This operation removes the reference from its bundles and resets it to its initial singleton-bundle state.

5.5.3 Export

Once equivalent references have been discovered and the corresponding bundles constructed, it is possible for the CRS to export bundles for use by other systems.

The export service is central to the CRS, supporting the retrieval of data. It is the thesaurus interface, allowing users to look up URIs to see if multiple equivalent references exist. It can take a variety of arguments: users can provide the URI of a reference, in return for any matching bundles; a string literal can be provided, for situations where the user may be unaware of a URI, or wishes to do a more general search; or a bundle may be retrieved directly by providing its URI.

The export service outputs data in one of two different formats: raw bundles in RDF, or as OWL equivalence statements. OWL output is provided for agents that wish to make strong equivalence assertions, and backwards compatibility with legacy systems. It is formed by translating the bundle data into owl:sameAs and owl:differentFrom statements, with the canonical URI being used as the subject of the relations.

5.5 Prototype Interface

As an illustration of how to utilise the CRS' services and to provide a manual fallback, we created a simple interface (shown in Fig. 2) through which one may search through bundles and establish coreference.

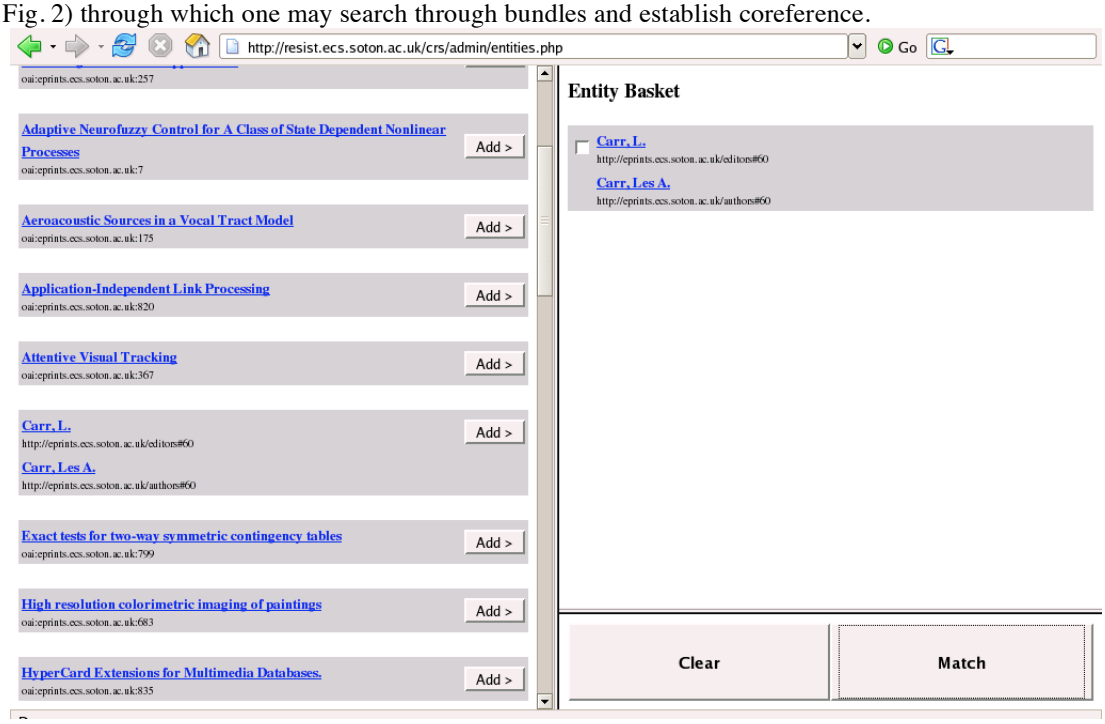

Fig. 2. Screenshot of the Manual Interface

The interface has the facility to display all the metadata associated with any references contained within a bundle, thereby providing all the available information required to perform the matching. This allows the user to make an informed decision as to whether to assert that given bundles are equivalent.

The interface is used by performing keyword searches on literal values. The results are displayed as a readout of the matching bundles and their contents, upon which a variety of operations can be performed. It is possible to merge bundles, delete bundles, and remove references from them. When a reference is removed, a new singleton bundle is created containing just the removed reference. If a bundle is deleted all the references are reset into singleton bundles.

5.6 Test Deployment

In order to demonstrate that bundles and the CRS are a capable method of handling coreference, we performed a test deployment of our implementation. The CRS we constructed was built upon version 3 of the 3Store knowledge base software. 3Store is a triple store implementation that uses MySQL, whereby queries made to it are translated into SQL queries upon the underlying relational database. The advantage of this approach is that it inherits the mature query optimisation present in MySQL, helping it to maintain a high level of responsiveness and scalability.

The ReSIST project [40] knowledge base was used as a source of data. This was populated with metadata from the ECS EPrints server, which had previously been shown to contain many instances of coreference and held metadata pertaining to some 10000 different articles.

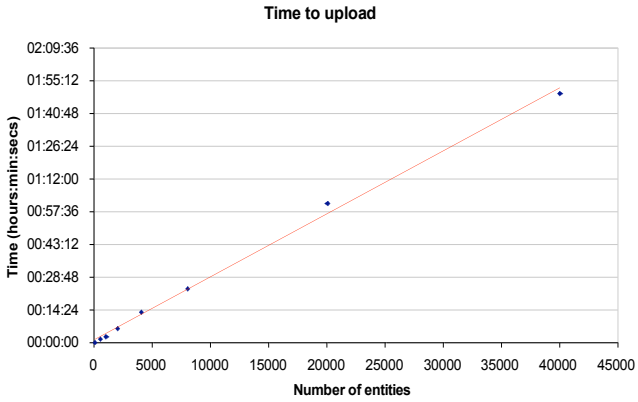

Fig. 3. Import service scalability test results.

With the server successfully deployed, scalability tests were performed to show that the CRS is capable of handling the amount of data present in a sizeable live deployment. Two subsystems were tested: a worse case scenario for the import service, whereby a new service connection was established to upload each URI individually; and the export service. The former was tested by timing the length of time taken to upload an increasing number of entities. The export service was tested by comparing the response time with different quantities of data in the server. The search by string mode was chosen as it performs the most complex queries; it would therefore be the first subsystem to show a drop in performance.

The import service was shown to scale in roughly linear fashion, as can be seen in Fig. 3. It took under two hours to upload metadata from the full, 10,000 EPrint repository. The performance of executing a search on the CRS was consistently well under one second, it did not noticeably degrade with the increasing number of entities: a very favourable result, as the performance of the query engine is crucial to the performance of any plug-in, or software utilising the CRS. The level of scalability that the server is capable of, demonstrated by these results, shows that this method of representing and handling coreferences is highly efficient.

6 Conclusions

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.

As larger knowledge bases and initiatives appear more frequently, coreference will become a significant barrier to progress and the need for an efficient system for managing references will increase, rather than subside. 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.

Acknowledgements

Many people have contributed directly and indirectly to this work over a number of years, including many members of the AKT and ReSIST projects. We thank them all, and in particular Harith Alani, Les Carr, Nick Gibbins, Steve Harris, Afraz Jaffri, Duncan McCrae-Spencer, Brian Randell, Benedicto Rodriguez, Nigel Shadbolt and Mikael Suominen.

This work was partially supported under the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC) and the ReSIST Network of Excellence.

The AKT IRC is sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/01. It comprises the Universities of Aberdeen, Edinburgh, Sheffield, Southampton and the Open University.

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

References

1. T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web". *Scientific American*, May 2001.
2. O. Lassila and R. Swick, "Resource Description Framework (RDF) Model and Syntax Specification". *W3C recommendation*, W3C, Feb 1999.
3. T. Berners-Lee, "Linked Data – Design Issues", http://www.w3.org/DesignIssues/LinkedData.html
4. "Linked Data Initiative", http://linkeddata.org/
5. AKT, "The AKT Manifesto". Technical report, 2001. http://www.aktors.org/publications/Manifesto.doc
6. M. Uschold and M. Gruninger, "Ontologies: Principles, Methods and Applications," *The Knowledge Engineering Review*, vol. 11, no. 2, pp.93–136, 1996.
7. M. Wooldridge, and N. R. Jennings. "Intelligent agents: Theory and practice" *The Knowledge Engineering Review*, vol 10, no. 2, pp. 115-152, 1995.
8. A. Rahm and A. Bernstein, "A survey of approaches to automatic schema matching". *The Very Large Databases Journal*, 10(4):334-350, 2001.
9. C. Lewy, *Meaning and Modality*, Cambridge: Cambridge University Press, 1976.
10. L. Wittgenstein, *Philosophical Investigations*, 2nd. ed., Oxford: Blackwell, 1958.
11. M. Jubien, *Ontology, Modality and the Fallacy of Reference*, New York: Cambridge University Press, 1993.
12. Wikipedia contributors, "Identity (philosophy)," *Wikipedia, The Free Encyclopedia,* http://en.wikipedia.org/w/index.php?title=Identity_%28philosophy%29&oldid=127375038, accessed May 14, 2007.
13. J. Dryden (trans.), Plutarch, *Theseus – A study of the life of Theseus*, AD75.
14. B. A. Brody, "On the Ontological Priority of Physical Objects", *Noûs*, Vol. 5, No. 2, pp. 139-155, 1971.
15. L. Loemker (ed. and trans.), G. W. Leibniz, *Philosophical Papers and Letters*, 2nd. ed., Dordrecht: D. Reidel, 1969.
16. T. Sider, *Four-dimensionalism: An Ontology of Persistence and Time*, Oxford University Press, 2001.
17. G. S. Cumming and J. Collier, "Change and identity in complex systems", *Ecology and Society* 10(1):29, 2005.
18. C. McCall, *Concepts of Person: An Analysis of Concepts of Person, Self and Human Being*, Aldershot: Avebury, 1990.
19. "anaphora." Dictionary.com Unabridged (v 1.1), Random House, Inc. 14 May, 2007, Dictionary.com http://dictionary.reference.com/browse/anaphora
20. T. Reinhart, *Anaphora and Semantic Interpretation*, London: Croom Helm, 1983.
21. O. Lassila, "Introduction to RDF Metadata" *W3C Note*, W3C, November 1997, http://www.w3.org/TR/NOTE-rdf-simple-intro
22. F. Leise, K. Fast and M. Steckel, "What Is A Controlled Vocabulary?", December 2002, http://www.boxesandarrows.com/view/what_is_a_controlled_vocabulary_, Accessed 03 April 2007.
23. Library of Congress Classification Outline, Library of Congress, http://www.loc.gov/catdir/cpso/lcco/, Accessed 03 April 2007.
24. S. Mazzocchi, "On the Quality of Metadata..." *Stephano's Linotype*, Blog Article, January 2006, http://www.betaversion.org/~stefano/linotype/news/95/, Accessed 03 April 2007,
25. C. Lagoze, H. Van de Sompel, N. Nelson and S. Warner (editors), The Open Archives Initiative Protocol for Metadata Harvesting v2.0, 2002.
26. The University of Illinois OAI-PMH Data Provider Registry, University of Illinois at Urbana-Champaign Engineering Library, http://gita.grainger.uiuc.edu/registry/ListSchemas.asp. Accessed 03 April 2007.
27. Dublin Core, Dublin Core Metadata, 1997, http://purl.org/metadata/dublin_core.
28. E. Wenger, *Communities of Practice: Learning, meaning and identity,* Cambridge University Press, 1998.
29. K. O'Hara, H. Alani, and N. Shadbolt, "Identifying Communities of Practice: Analysing Ontologies as Networks to Support Community Recognition" in *Proceedings of the World Computer Congress*, 2002.
30. H. Alani, K. O'Hara, and N. Shadbolt, "ONTOCOPI: Methods and tools for identifying communities of practice" in *Proceedings of the 2002 IFIP World Computer Congress*, Montreal, Canada, August 2002.
31. H. Alani, S. Dasmahapatra, N. Gibbins, H. Glaser, S. Harris, Y. Kalfoglou, K. O'Hara, and N. Shadbolt, "Managing Reference: Ensuring Referential Integrity of Ontologies for the Semantic Web" in *13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, pages 317-334, Siguenza, Spain, 2002.
32. Academic Contributor Information System Project, http://acis.openlib.org/, 2006.
33. T. Krichel and I. Kurmanov, ACIS Stage Three Plan, http://acis.openlib.org/stage3/, 2005.
34. T. Krichel and I. Kurmanov, ACIS project: phase 1 requirements, http://acis.openlib.org/documents/kathmandu.html, 2003.
35. RDFWeb: FOAF Developer site Wiki, "smushing", http://rdfweb.org/topic/Smushing, Accessed 15 May 2007.
36. D. McGuinnes, F. van Hermelen, "OWL Web Ontology Language Overview". *W3C recommendation*, W3C, Feb 2004.
37. C. Gutteridge, "GNU EPrints 2 Overview" in *Proceedings of 11th Panhellenic Academic Libraries Conference*, Greece, 2002.
38. T. Lewy, "A Consistent Reference Service for the Interoperation of EPrint Repositories" Technical Report, School of Electronics and Computer Science, University of Southampton, 2006.
39. "SPARQL Query Language for RDF". *W3C Working Draft*, 2005, http://www.w3.org/TR/rdf-sparql-query/.
40. ReSIST Project. "ReSIST: Resilience for Survivability in IST", Project Presentation, http://www.resist-noe.org/Project_presentation.pdf, 2006.