The Knowledge Engineering Review, Vol. 22:4, 361–377.
doi:10.1017/S0269888907001233 Printed in the United Kingdom

The usability of semantic search tools: a review

VICTORIA UREN[1], YUANGUI LEI[1], VANESSA LOPEZ[1], HAIMING LIU[1], ENRICO MOTTA[1] and MARINA GIORDANINO[2]

[1]*Knowledge Media Institute, Open University, Milton Keynes, MK7 6AA, UK, e-mail: v.s.uren@open.ac.uk, y.lei@open.ac.uk, v.lopez@open.ac.uk, h.liu@open.ac.uk, e.motta@open.ac.uk*
[2]*Centro Richerche FIAT, strada Torino 50, 10043 Orbassano (Torino), Italy, email: marina.giordanino@crf.it*

Abstract

The goal of semantic search is to improve on traditional search methods by exploiting the semantic metadata. In this paper, we argue that supporting iterative and exploratory search modes is important to the usability of all search systems. We also identify the types of semantic queries the users need to make, the issues concerning the search environment and the problems that are intrinsic to semantic search in particular. We then review the four modes of user interaction in existing semantic search systems, namely keyword-based, form-based, view-based and natural language-based systems. Future development should focus on multimodal search systems, which exploit the advantages of more than one mode of interaction, and on developing the search systems that can search heterogeneous semantic metadata on the open semantic Web.

1 Introduction

When cannot a seat be sat upon?

One answer to this riddle might be: when you search for it on the Web. In most of the popular Web search engines, such as Google, IceRocket, ASK and AltaVista, the top hits on a search for 'seat' are about SEAT, the Spanish car maker. It is problems like this with keyword-based search that are driving the quest for semantic search engines. The idea is that a semantic search engine would support the user in formulating a search to get, not the results about SEAT cars or seats of government but the results about furniture you can sit on, like chairs, sofas and benches. In this review, we explore the types of features semantic search systems require, and take a look at some of the semantic search systems that have been developed so far.

The phrase 'semantic search' is used to describe several different kinds of search. To clarify the kind of semantic search discussed here, we are not interested in searching for ontologies, as is done by systems like Swoogle (Ding *et al.*, 2004). Nor are we interested in semantic document search, which employs a combination of semantic metadata and other document features to enhance the performance of document retrieval, for example, Vallet *et al.*, (2005). Our interest lies solely in searching the semantic metadata.

We do not constrain ourselves to any particular ontology language, such as OWL Full, Resource Description Framework Schema (RDFS), rather the scope of our review is the search systems that operate on RDF triples and which may contain uniform resource identifiers (URIs) that link to Web-accessible resources (e.g. 'Enrico Motta' *has homepage* `http://kmi.open.ac.uk/people/motta/`). We refer to such RDF triples as 'semantic metadata'.

We define an annotation as an RDF triple, which is linked to a Web resource and represents the fact that a particular part of a resource denotes a particular concept. Annotations can be understood to include the relevant fragment of the original resource, but we concern ourselves only with the basic RDF triple.

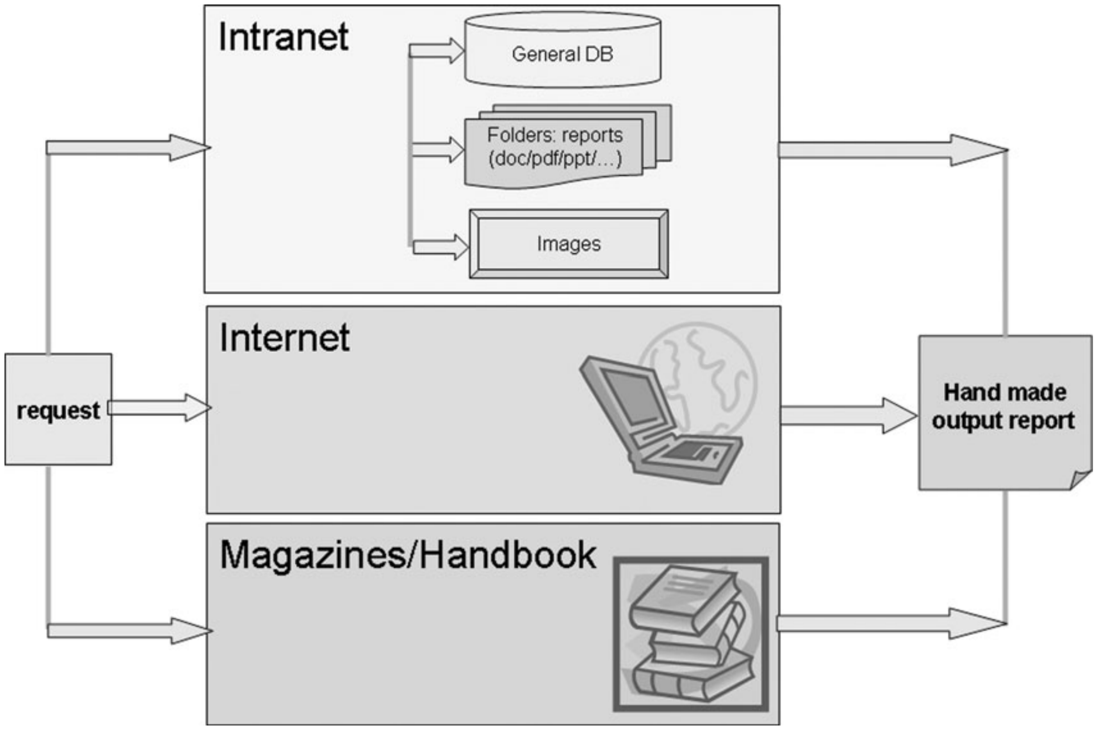

Figure 1 Current knowledge access process of FIAT analysts

Therefore, we define metadata-oriented semantic search as the search which:

- enables the querying of RDF triples, both semantic metadata and annotations;
- enables access to Web resources referenced as URIs in RDF triples;
- works on semantic metadata repositories, often called as knowledge bases or KBs, specifically RDF KBs, as used by the semantic Web;
- and makes use of semantics defined by domain ontologies to support query formulation and to improve the precision of search results.

Our focus in this review is on the user experience of search tools rather than on computational, formal or implementation issues associated with search or their retrieval performance. We start by presenting an example that motivates the need for semantic search. Then, in Sections 3 and 4, we identify usability requirements. Four different modes of user interaction are reviewed in Section 5: keyword-based, form-based, view-based and natural language–based. In Section 6, we suggest some challenges for future research in semantic search systems.

2 Motivating example

To further motivate the need for semantic search, we now examine an example from the X-Media project[1], based on the experience of the Competitor Analysis Department at FIAT AUTO. At FIAT AUTO analysts use a range of public and private resources to produce reports on engineering solutions developed by competitor companies. Figure 1 summarizes the process used, in which parallel searches are run on the Internet, on the Intranet using resources including the collection of photographs produced during dismantling of competitor's cars, and in collections of brochures, magazines, handbooks, etc., gathered by the Competitor Analysis Department at events such as trade fairs.

[1] http://nlp.shef.ac.uk/X-Media/
www.x-media-project.org

The example concerns a request for information about competitors' car models, which have a lever that adjusts the front seat cushion. Using current search technology, the search is difficult. A bilingual Internet search on the English terms 'seat, cushion, comfort, automotive' and the Italian terms 'sedile, cuscino, comfort, auto' resulted in about 15 000 documents, most with the keywords sparsely distributed and very few of them meaningful. Refining the search to look for documents, where the terms occurred close to each other, resulted in no hits. Searching the internal resources produced better results, but required laborious browsing through different sources for each different kind of data. With very considerable effort, the analysts managed to find three interesting solutions: a front seat angle adjustor in the Mazda 3, a clever seat folding mechanism developed by Toyota and a front seat swivel mechanism, designed to improve access for people with limited mobility, designed by Honda.

The semantic solution to this kind of knowledge access problem is to start by identifying, or creating ontologies (Gruber, 1995), which model the key aspects of the users' typical needs. In the FIAT case, the analysts need to know about vehicles, and their performance, components and manufacturers. Annotations must then be generated for the resources according to these ontologies, that is, if a resource contains a picture of a Mazda seat with a cushion adjustor lever, then links must be created to the parts of the ontology (or ontologies), which are about front seats, levers and Mazda cars, and critically the relations specifying that the front seat is a *component of* a Mazda and the lever is a *component of* the front seat. As we have argued elsewhere (Uren *et al.*, 2006), it is crucial that this annotation is automated: fully automated for processing legacy collections and with semi-automated support to help knowledge workers produce high-quality annotations as an integral part of document creation. Developing such automatic methods to support annotators is a critical part of the X-Media project, but is not the focus of this review. In this paper, we are interested in semantic search and, in particular, the usability of four modes of formulating semantic queries. For the example above, semantic search tackles the problem in two ways. The first is that it provides a single access approach for all the diverse internal resources, which are currently stored in multiple collections, opening the possibility of searching all the resources together. The second is that because the ontology and the metadata can represent the fact that a lever is a *component of* a seat, a search can look for that relation. If given appropriate search tools, the analysts would no longer have to hand filter thousands of hits, which just happen to contain both the word 'seat' and the word 'lever'. Instead, they could search specifically for the resources about seats with levers. Returning to our original, light-hearted example, by exploiting relational search, the semantic solution should even allow the user to search for SEAT seats.

3 Iterative and exploratory search

The usability of semantic search tools sets the agenda of this review. We are interested in what is required to support end-users, like the FIAT analysts, to formulate searches. Such users may lack knowledge about how the ontology models the information space of a specific domain, particularly if the information space is complex or their work requires them to seek information on a wide range of topics. Therefore, the search system needs to help them explore the ontology to find out what is there and how it is organized. In addition, ordinary users may not have a clear idea of what their information need is. Therefore, they need to use the results of one search to suggest ways to refine search parameters to get better results. Therefore, we see search as an iterative and exploratory process, in which the user actively engages with the system. The key difference between a purely iterative search and one which is also exploratory is that 'Exploratory searches are typified by uncertainty about the space being searched and the nature of the problem that motivates the search' (White, 2005). This lack of definition calls for new techniques that support browsing of the information space, since keyword search is of little help (Marchioni, 2006).

Seeing search as an iterative and exploratory process brings the realization that it is not just the formulation of one off search statements which is important but also longer-term activities, such as search refinement, and the reuse and recommendation of search strategies. The main types of activity an ideal search system would support are:

- Query formulation, which guides users to construct queries that meet their information needs.
- Query refinement, which allows users to modify previous queries to better capture their information needs.
- Query reuse, which enables users to reuse (parts of the) queries they have defined previously.
- Query recommendation, which suggests queries to users based on information the system has learnt from their past search behaviour.

Of these, the first is, of necessity, always addressed with differing degrees of success and we have identified four modes of user interaction, which have been used to structure this review. The last three are reported when they occur in the reviewed systems.

4 Searching semantically

Making searches semantic is about operating in an environment where the meanings of particular symbols have been defined. As indicated above, this is done using an ontology, or sometimes a simple taxonomy. Both ontologies and taxonomies divide the world into classes and subclasses of things, and define individuals as instances of one or more classes. Ontologies can also define relation types, for example, *components of* which allow them to specify that certain classes or instances have particular properties. Objects in ontologies may be linked to references to those objects in original documents, providing annotations.

There are some specific requirements for a system to exploit semantic metadata to its full potential. These relate to the types of query a user might need to make, the semantic search environment and some intrinsic problems that arise when dealing with search terms in a semantic, rather than a syntactic, way. We discuss these below.

4.1 Query type

Semantic search has the potential to resolve the ambiguities inherent in keyword search by specifying the kind of results that are required with much greater precision. In general, three types of search may be supported: searches for *entities*, searches for *relations* and *parameterized* searches. Entity search is the most commonly provided type of search. It serves the case where the user needs information about a particular kind of thing, such as front seats, which is represented in the ontology as a class, or about an individual, such as a specific front seat with an ID number, which is represented in the KB as an instance. Relation-based search looks for relations between different entities. It serves the case where the user wants to find out about how entities are connected without specifying the type of link. For example, it might look for relations between seats and cushions. The third type, parameterized search, serves the case where the user has very precise needs for which he/she must specify several parameters. Typically, both the relation type and the values that could satisfy it would be specified. For example, he/she wants to find information about a particular seat used in the Mazda 3 that has at least one lever as a component. With parameterized search, the results would be subgraphs of the ontology.

4.2 Search environment

The semantic search environment is potentially complex and large. The complexity arises because multiple ontologies will be required to describe the different topics and domains. This entails two requirements. The first is *portability*; the system has to move between ontologies without any need for domain-specific reconfiguration. The second requirement is the ability to support *heterogeneity*; the system must be able to search several different ontologies at the same time, for example, the term ‘seat’ may occur in a furniture ontology (chairs), a vehicle parts ontology (car seats),

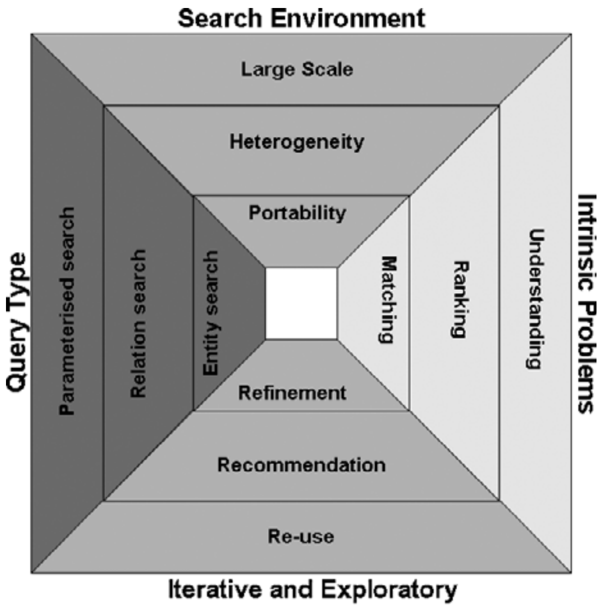

Figure 2 A pyramid representing the requirements space for semantic search systems

an international affairs ontology (seats of government) and a travel ontology (booking a seat). The search system would need to find all of these and eliminate irrelevant cases. If necessary, it might also have to fuse the knowledge structured according to different ontologies together. The potential size of the semantic search environment is predicted by a recent study (Lee & Goodwin, 2005), which suggests that the semantic Web is currently growing at the same rate as the Web in the early 1990s. This requires that semantic search systems should be able to support both large-scale ontologies (with many classes and instances) and large-scale repositories (with many annotations). An extended discussion of these and related issues of open semantic environments is presented by Motta & Sabou (2006). Of these two factors, this review concerns itself with whether the user interface would be able to support *large-scale ontologies* in a usable manner. The question of whether large-scale repositories can be supported is determined more by the capacities of underlying repository technologies, such as Sesame[2], and the efficiency of search algorithms.

4.3 Intrinsic problems

Dealing with searches and queries at the semantic level, instead of the syntactic level, presents some intrinsic technical problems (see Figure 2), which we will discuss because they have usability implications, and an intrinsic usability problem. The first of the technical problems is how to do semantic *matching* of search terms to entities. This includes finding mappings (e.g. mapping the class *Car* in one ontology to the *Automobile* class in another) and disambiguation (e.g. distinguishing *seats* from *SEAT* cars). The second technical problem is to find a way of returning results that are presented according to how well they satisfy user queries. The problem is that semantic search systems are logic-based and classical logic returns true/false answers. Foundational work is underway on producing approximate answers to logical queries and on *ranking* (see e.g. Bamba & Mukherjea, 2005; Stuckenschmidt *et al.*, 2005). Both of these technical issues have implications for the user experience; solutions may reduce the need for user interaction and query refinement. The final intrinsic usability problem is how to help ordinary end-users *understand* multiple

[2] `http://www.openrdf.org/`

1. Tool description:
   - Name
   - URL
   - A brief overview
   - Type: document oriented/semantic data oriented
2. Investigation:
   - User support:
   - Query formulation
   - Query re-formulation
   - Result presentation (e.g., search results clustering, de-duplication, relevance ranking)
   - Matching of search terms
   - Syntactic match (i.e., string match)
   - Semantic match (i.e., word sense match, e.g., desk to table)
   - Performance (where applicable)
   - Response time (real time)
   - Accuracy of answers
   - Semantic oriented performance: heterogeneity and search for relations
   - Effective ranking (?)
   - Data formats (text, image, audio, video, etc.)
   - Portability (how much effort will be needed to apply the tools in different domain)
   - Heterogeneity
   - The capability of handling different data representation formats in terms of semantic web standard languages (e.g., RDF, RDFS, OWL)
   - The capability of searching and aggregating data from different sources
   - Entity centred search and relation centred search
   - Entity search: searching for semantic entities
   - Relation search: searching for semantic relations
3. Observation: a brief description about the overall impression of the specified tool, in particular good features and not so good aspects.

Figure 3 The template used for reviewing individual tools

domains in order to formulate good queries, since it cannot be assumed that every user will be familiar with all the ontologies he or she needs to use.

5 Review of semantic search systems

To perform the review, we divided up the semantic search tools on the basis of their user interaction mode into the categories: keyword-based, form-based, view-based and natural language-based systems. Keyword-based systems accept one or more keyword/s as input and return semantic entities that match the keyword/s. These systems look like ordinary information retrieval systems on the surface, but allow the user to specify their information needs more accurately by ‘translating’ query terms into semantic entities. Form-based systems use forms, menus and drop-down lists, based on ontology structures, to guide the user in formulating semantic queries. View-based systems use ontology presentation and ontology navigation to support query construction and domain exploration. Typically, the user can build a query by navigating through a visual representation of the ontology structure and/or query structure. Natural language systems using various linguistic techniques to translate user queries submitted as natural language sentences into ontological queries. Some systems support several interaction modes. These have been categorized according to which mode of interaction was most novel or best developed.

The work of reviewing individual systems was divided among the first four authors. To ensure consistency, we used a template (see Figure 3) as a reminder to the reviewers to look for the features that interested us most. Tools that had a demonstration available were considered, as well as published papers. Based on the filled templates, we selected the tools to feature in the paper. For each mode of interaction, we describe at least one typical example in some detail to illustrate its key features. Other tools are discussed if they have features that are pertinent to the discussion. However, it should be noted that the paper does not represent an exhaustive catalogue of semantic search tools. In what follows, we discuss the key features of examples of the four modes of interaction. From the individual systems, we draw out the strengths and weaknesses of the different interaction modes in terms of their usability. The discussion is necessarily qualitative since there are, as yet, no benchmark datasets for the semantic search systems and, therefore, they all focus on different domains.

5.1 Keyword-based systems

Keyword-based semantic search systems make efforts to enhance the performance of traditional keyword search by exploiting the availability of explicit semantics. They often accept one keyword as input and return documents, which are annotated with those semantic entities that are closely related to the keyword. The search process typically starts by locating the keyword in the specified domain ontology and the associated semantic metadata repository by using text search techniques. The search systems then find related semantic entities and their associated documents for the keyword match.

One major advantage of keyword-based tools is that they provide end-users with a straightforward way to specify queries; end-users are often very familiar with this mode of interaction. They are able to use them without having to know about the exact vocabulary or structure of the ontology, or having to master a special query language. The success, or otherwise, of the search depends only on the choice of keyword and how the search algorithm processes it.

The TAP search engine (Guha *et al.*, 2003) was one of the first keyword-based semantic search systems. It provides a good example of a typical keyword search algorithm. The TAP search engine first compares the keyword term against the labels (rdfs:label) of semantic entities indexed by the search engine. It then selects one entity from the candidate matches by looking at (i) the popularity of the entity in the gathered text corpus or in the metadata repository, (ii) user profiles, and (iii) search context. After finding the semantic entity match of the keyword, the TAP search engine uses graph-walking techniques to collect and cluster search results in order to decide what to show as results and in which order. The results are clustered semantic entities that are closely related to the keyword entity match. The clustering technique requires 'close' relations to be predefined for each class.

To choose the right keyword for a successful search, end-users have to have general knowledge about the problem domain, in the same way that they would to formulate a good keyword query for text search. Keyword search tools do not support the case where the user needs to first learn more about the domain in order to formulate a search. For example, when experimenting with the ski demo of the Squiggle[3], search engine in the review process, the second author did not know what to ask, as the images are annotated with skiers' names and event names such as '*Riesenslalom'*. Without some knowledge of the skiing domain, it was impossible to formulate a search.

More commonly, the user knows some of the vocabulary of the domain, but is unable to formulate the search perfectly first time. In this situation, the query needs to be refined. Users of text search engines use various strategies for refinement including substituting broader and narrower terms to increase or decrease the number of hits, respectively. We have taken some first steps towards implementing search refinement for narrowing scope in our SemSearch[4] (Lei *et al.*, 2006) tool, although currently the approach is predefined and not flexible. The mechanism involves narrowing down the meaning of user keywords by picking the desired terms from a list of candidates found by string matching (see Figure 4).

The key innovation of SemSearch is that it supports multiple matches. This is challenging for semantic search, because the simple presence of two concepts in the same document is not sufficient. Instead, it is necessary to show that there is a semantic link between them, expressed in the RDF triples. This is the activity that we call relation search. SemSearch constructs several formal queries for one semantic relation query, using different combinations of keywords' matches, and combines their query results together. One advantage of this mechanism is that the search engine considers all the possible meanings of keywords and does not miss out

[3] http://swa.cefriel.it/Squiggle/
[4] http://semanticweb.kmi.open.ac.uk:8080/ksw/pages/semantic_searching.jsp/

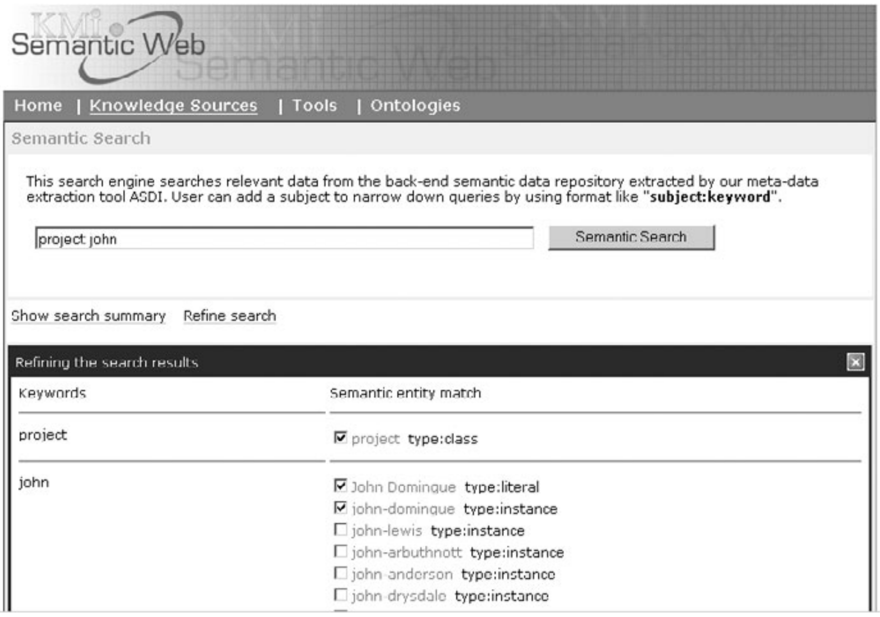

Figure 4 Screenshot of SemSearch showing the user refining a search that has multiple possible meanings for the search term 'john'

potentially important relations. The limitation is that it slows down the response performance of the search engine when the number of combinations gets big, hence the need for a refinement mechanism.

The keyword-matching mechanisms of current tools are typically dealt with, as by SemSearch, at the syntactic level, using string-matching techniques. This means they can easily be applied to different domains, as they tend not to be coupled with domain ontologies but as a consequence the systems cannot really understand the information needs of end-users. Hence, they cannot always produce good results for end-users. For example, when querying for 'seat', any information about 'bench' would be missed out because of the syntactic matching. Thus, semantic matching mechanisms should be incorporated in the keyword semantic search routine. Work is being undertaken to extend keyword matching from a syntactic level to a semantic level, by making use of domain-specific ontology and lexical resources (e.g. WordNet) to find semantic matches for the user keyword. Typical examples include ZOOM[5] and the tools presented in Mihalcea & Moldovan (2000) and Buscaldi *et al*., 2005. Although they do not target semantic metadata search, their distinctive feature of semantic matching is appealing.

5.2 Form-based systems

Forms are a familiar interface metaphor for all computer users, making them a popular approach for developing semantic search interfaces. Form-based interfaces can side-step mapping issues by getting users to select terms from the lists of valid terms. They support user understanding of the domain by, quite literally, showing the user what is there, and what they can expect valid searches to look like.

The production of forms is facilitated by the fact that selected parts of ontology structures can be directly translated into lists and forms and, as a result, form-based systems are relatively portable. For example, if a field in a form must be filled with the name of a company that makes cars, then a pick list might be generated from the ontology of all the instances of the class 'CarManufacturer'. The slots used to define the properties of a class can similarly be turned

[5] `http://www.semantic-knowledge.com/doc/V70/semantic-search/quick-guide.htm`

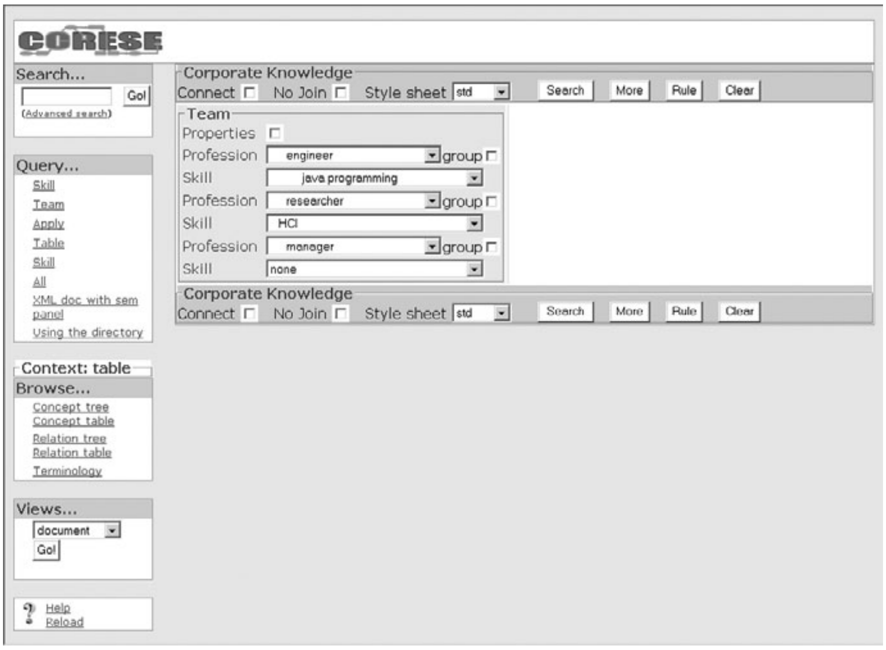

Figure 5 A typical form in Corese. This allows the user to search for employees with selected skills to form a team. Skill and Profession slots are filled by picking from drop-down lists of valid classes

into search forms. Continuing our example, one could generate a form for searching for cars, which allows the user to specify CarManufacturer, VehicleSector and so on.

The Corese library can support form-based interfaces. The Online demo[6] provides keyword search and hierarchy browsing, as well as one form for each of the most searched classes in the ontology (see Figure 5). The forms have check boxes and drop-down lists, which allow the user to specify the properties required for a parameterized search. It is a well-developed tool that has demonstrated the portability of its approach by having been applied for more than ten different domains, including competence management and bioinformatics (Corby *et al*., 2006). Magnet (Sinha & Karger, 2005), which is a component of Haystack, takes portability a step further by being able to generate the content of listings and search options on the fly, out of the information structure. This demonstrates the natural relationship between forms and structured information, which can be exploited in semantic search systems. Whether this approach is scalable for large ontologies is a more difficult issue, since there is a limit on the number of items that can sensibly be included in a scrolling list, or the number of slots that can be put on a form. We believe this limits the usability of form-based interfaces.

While tools like Corese are excellent for constructing and refining parameterized searches, they do not support exploratory browsing particularly well. An alternative approach, which also generates form-like interfaces direct from the structure, is demonstrated by mSpace[7] (schraefel *et al*., 2006) The 'form' in mSpace comprises columns, each representing a major concept in the ontology. By selecting or deselecting items in a set of columns, the user can iteratively refine a faceted search until they get the desired result. Since columns can be added and removed and their order changed, the number of possible searches is quite high. mSpace is an excellent approach for introducing a user to an unknown information space, where they do not know how the information is structured, or what they could get out of it. It is, however, limited to setting parameters in a search for entities, and it is not clear at this point whether it could be further developed to search in a

[6] http://ubaye.inria.fr:8080/demo/servlet/corese?screen=tree_concepts.xsl

[7] http://beta.mspace.fm/

heterogeneous environment, since the interface design is locked into the main concepts of a domain.

Indeed, heterogeneity, that is, searching in multiple ontologies, was not tackled by any of the form-based systems we reviewed, and it is not obvious how something like a form with attribute slots could be generated in a heterogeneous scenario. Since searching in multiple ontologies is particularly important for the open semantic Web, as opposed to the more controlled environment of corporate semantic Webs, this may limit the applicability of form-based search.

5.3 View-based systems

View-based search tools can work well for end-users in domain understanding, query construction and query refinement. The underlying semantic metadata collection is presented as graphical, tree-based, or textual views, using the associated domain ontology to structure the view. The advantage is that the query vocabulary and content classification scheme can be presented in intuitive formats, thus facilitating domain understanding. Queries are often built by means of navigation.

A typical example of a tool that has this mode of interaction is GRQL (Athanasis *et al.*, 2004), which assists end-users to construct RDF queries dynamically on the fly by navigating through visualizations of the specified ontology. It allows end-users to construct complex queries without having to have domain expertise or having to master RDF query languages. First, the user selects a class in the ontology, as point to start navigating from. All properties defined as applicable to the class in the ontology are then given for expansion. Clicking on a property expands the graph pattern to contain that relation, and moves selection to the range class defined for that property, for example, clicking the 'creates' property in an Artist class creates the pattern 'Artist->creates-> Artifact', and moves the focus to the Artifact class, showing the properties for that class for further path expansion. The underlying assumption of query construction in GRQL is that each concept or property that the end-users clicked should be part of the query. Such an assumption may not apply when users focus on exploring the problem domain, rather than constructing meaningful queries.

Query construction by means of navigation, however, can be time consuming and may feel rather inflexible. For example, SEWASIE (Catarci *et al.*, 2004) is another graphical query interface, which supports end-users in query construction. The user is first presented with a choice of starting points that are key concepts like 'goods', 'enterprise' or 'supplier', which can then be extended and customized in a series of refinement steps. The refinements to the query can be either adding additional property constraints to the classes or replacing a class in the pattern with another compatible one, such as a sub- or superclass. In systems like this, the number of steps to complete a query can get big when the associated ontology is complex. The support offered by the stepwise process can be very helpful, particularly when the user does not understand the domain, but it requires patience and determination to complete the process. One solution to speeding up the view-based query formulation has been developed in Ontogator (Hyvonen *et al.*, 2003), a multi-facet search tool, by integrating the view-based navigation routine with keyword search.

The time-consuming interaction with some view-based tools poses one concern for how they would scale up to big ontologies. Another concern is that, when the associated ontology gets complex, it is more challenging to present views of the problem domain that are both effective and intuitive, so that end-users do not get lost in the information space. The problem is illustrated in Figure 6, which shows the interface of Falcon-S (Wu *et al.*, 2006) a semantic search prototype focusing on the soccer domain. The top right panel shows the graph-based user interface for customizing existing queries or constructing new queries. For the simple ontology used in this example, the query graph is clear and easy enough to interpret. However, the panel uses a significant amount of screen 'real estate', and squeezing many more nodes onto it would start to make it cluttered. As in any graphical interface, there are technical limits on what can be presented on a screen, and perceptual limits on how much a user can take in from the visual representation. Tree-like listings can convey quite a lot of information [20–60 items in a list according to

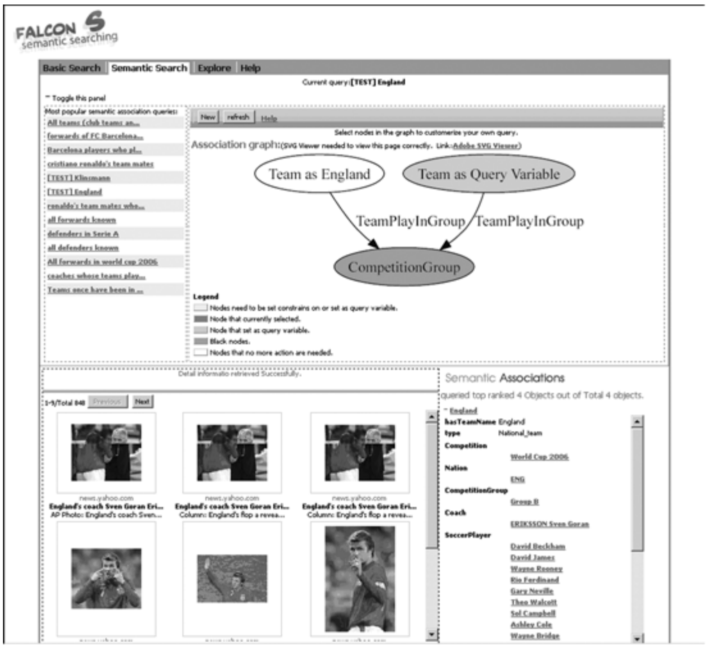

Figure 6 Screenshot of the Falcon-S interface showing reuse of a search selected from the menu (top left). The view (top right) showing the relations between the semantic associations, which are expanded on the bottom right. Search results are the images shown in the bottom left of the screen

Shneiderman (1998)], but these only work properly for simple hierarchies because they cannot show information about cross-links.

Another question concerning requirements that can be met by views concerns cases when the user's query is related to multiple ontologies: how can we combine different ontologies together to create appropriate views for end-users? Heterogeneity creates the same problems for view-based search as for form-based search. This question needs to be answered in order to progress current view-based search tools to the real semantic Web, where multiple ontologies are certain to be encountered.

5.4 Natural language systems

The attraction of natural language semantic search tools is easy interaction for the user. In the past, question answering (QA) using natural language approaches has largely been focused on retrieving the answer from raw text, and on supporting query expansion in information retrieval (Mc Guinness, 2004). However, the availability of semantic mark-up opens the way to novel QA systems. The novelty of these systems, with respect to traditional QA systems, is that they can make use of semantic information to provide precise answers to questions posed in natural language. With respect to other modes of interaction, their advantage is that they provide a way to formulate queries with several parameters, which is more flexible than form- and view-based systems, but does not require the user to learn a formal search language.

AquaLog[8](Lopez *et al.*, 2005) provides an example of a portable, ontology-based QA system (see Figure 7). A QA system must bridge between the terminology of the user and the concepts used by the underlying ontology. In a first step, by using linguistic techniques, the system classifies and breaks up the question into linguistic triples. Then, these terms are linked and mapped to semantic entities and relations, generating the semantic triples from which the answer is derived, sometimes by assembling multiple triples to generate an answer. The process exploits various linguistic methods and domain knowledge from the ontology to disambiguate between possible

[8] `http://kmi.open.ac.uk/technologies/aqualog/`

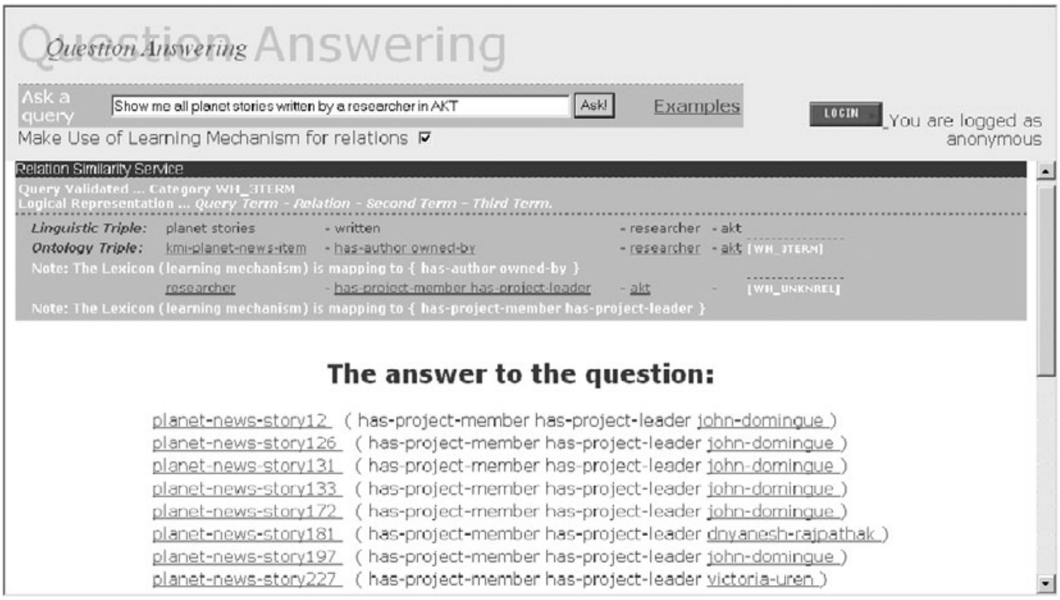

Figure 7 Aqualog interface showing at the top, the input natural language query, then intermediate triples identified by the system and finally answers. Links in the answers take the user to the original resources

interpretations of the user's natural language query. Whenever the right mapping cannot be found, the user is asked to choose between the alternative readings. Over time, the system can learn a particular user's 'jargon' from these interventions.

This kind of interpretation is a key function of semantic QA systems. By translating the terms of the user's query into those of the semantic resources, each user can formulate queries in the way that is natural to them. Semantic QA has similar limitations to keyword search, in that, since the QA system does not constrain queries in any way, or support the user in understanding the domain, users must be familiar with the domain to be able to pose reasonable questions. One solution to this is provided by the GINO system (Bernstein & Kaufmann, 2006), which offers guidance to the user as they formulate a 'quasi-English' query step by step, ensuring that only valid queries are posed.

AquaLog can be ported to a new domain with minimal intervention, the user still needs to tell the system which ontology is going to be used and define which classes in the ontology would be suitable to answer particular kinds of queries—for example, which classes describe people, and so can be used to answer 'who queries'. Similar tactics for separating knowledge that can be reused in many domains (general linguistic knowledge and knowledge about the compositional construction of queries) from domain knowledge, facilitate portability in other semantic QA systems. For example, Orakel[9] (Cimiano, 2004) makes use of two different lexicons: the general lexicon and the domain lexicon. The general lexicon includes words such as determiners (e.g. a, the, every) as well as questions pronouns (e.g. who, which) and, thus, is domain-independent. The domain lexicon varies from application to application, and is generated automatically for each application out of the KB.

Going beyond portability, we believe that natural language QA in open scenarios (i.e. the semantic Web scenario with multiple ontologies) is technically feasible, although it has not yet been achieved by current systems. Research is underway on the linguistic mapping mechanisms required to interpret the possible senses of a query in several different vocabularies and, if necessary, compose answers from fragments described according to different ontologies (Lopez *et al.*, 2006). To work on the large-scale of the semantic Web, these methods need to be lightweight in order to give a real-time response. Current systems also require more user interactivity

[9] http://ontoware.org/projects/orakel/

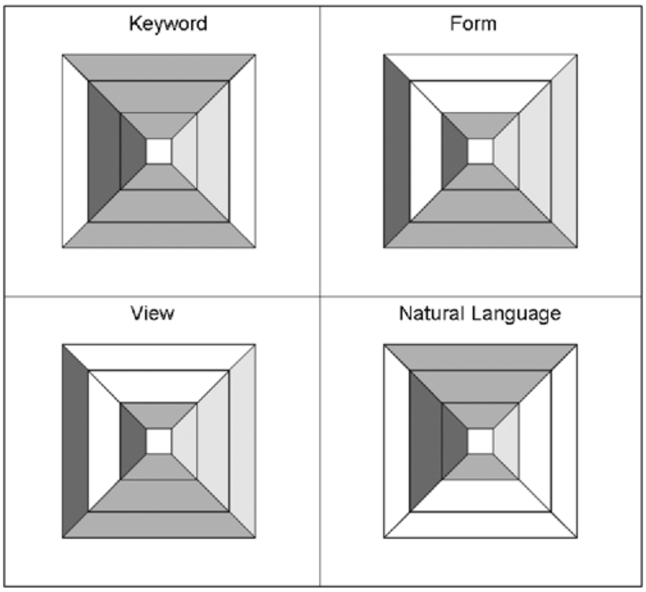

Figure 8 Modes of interaction matched to the requirements space. A dark bar represents a contribution of the system towards fulfilling the requirement. Clockwise from the top the sectors represent query type, search environment, intrinsic problems, iterative and exploratory. Figure 2 provides a key

to resolve problems with interpretation than would be acceptable in open scenarios, where the number of ambiguities can be expected to be much higher than in single domains.

Although posing questions in natural language provides a flexible way to compose queries that have several parameters, there is a limit to the number of clauses in a sentence that natural language parsers can interpret reliably. This produces in turn a limit to the number of parameters that can be queried. This problem is independent of the size of the ontologies being considered. Therefore, although the QA approach should scale to large ontologies, it cannot tackle very large parameterized queries.

5.5 Summary

In Figure 8, we summarize the features that the four modes of interaction could conceivably have. We have included some which are not present (to our knowledge) in the tools reviewed above, where we believe they are technically feasible for that mode of interaction.

The familiarity of the keyword search is a real advantage for users. The simplicity of the input also means that achieving search in heterogeneous environments should be possible, given appropriate support for disambiguation, since no interface modifications are needed to adapt the interface for different domains. Recommendation could be partly supported, despite the limited information in a keyword query by reusing the recommender techniques developed in the information retrieval (IR) community. Reuse is certainly possible, but may not be very useful in the keyword search environment where query formulation is simple. We have left out two features, which we believe keyword search cannot support. These are parameterized search and understanding. As SemSearch demonstrates, we can formulate a relation search from a set of keywords, but there would be too many possible ways of combining terms if the same approach were tried for parameterized search. Understanding of the domain, also, cannot be supported by keyword search. Of course, the user can learn from examining the results, but this is not a feature of the search mechanism.

The form interfaces we reviewed worked well for entity search and parameterized search. However, form-based search is not flexible enough to allow unknown relation types, so relation search is omitted. There is no obvious barrier to refinement or recommendation in forms, and reuse of complex parameterized searches of the type that can be produced in forms is an obvious feature to provide. We have discussed, above, how form-based search can overcome the intrinsic problems of semantic search by 'showing the user what there is'; however, the same interface metaphors that overcome these problems make form-based search unusable in large-scale ontologies and open, heterogeneous scenarios. At large-scale, there is simply a limit to how many options users would accept having to choose between to formulate a search. In heterogeneous environments, there is no obvious way to generate structured forms.

Regarding support for different query types, view-based tools work well for semantic entity search and can be used to specify parameterized searches. However, navigation-based query routines are not designed for relation-based search. It is in their nature to specify the relation type at each point, so the issue of undefined relations simply does not arise. As Figure 8 illustrates, the limitations of view-based approaches are similar to form-based approaches. As well as supporting the same query types, they too have difficulties with scalability and heterogeneity. In both modes of interaction, the main asset of the system (that it shows the user exactly what is there) becomes its fundamental weakness (when the volume of information the user would have to see become too large to make sense of).

Semantic QA systems are similar to keyword systems in being able to answer entity queries and relation queries. They can tackle parameterized queries, provided they have limited numbers of clauses and, in single domain environments systems like GINO, can be used to support the user in formulating valid parameterized queries. Importantly, although only single domain systems have been built so far, semantic QA systems are potentially flexible about the search environment they can operate in. This makes them appropriate for the open semantic Web scenario with its heterogeneous ontologies.

6 Challenges for the future

This review has shown that all three of the query types, identified in Section 4.a., are present in some of the four modes of interaction. However, as Figure 8 clearly illustrates, no single mode lends itself to supporting all three. Straightforward searching for entities is ubiquitous, and its effectiveness depends primarily on the efficiency of matching and ranking methods. However, on one hand, keyword search and natural language querying both support relation searching, because they offer a means for specifying two or more entities, without having to specify the exact nature of the link. On the other hand, view- and form-based search interfaces both show the user the exact options available in the ontology to assist them in constructing parameterized searches. The solution is obvious. Semantic search tools need to offer a range of different modes of search formulation, to allow users to pick the method that best suits their task. This need for building multimodal semantic search tools provides a first dimension in mapping the space that development of systems needs to address.

The second dimension relates to the hard challenge posed by moving from the controllable, single domains, addressed by semantic portals and semantic intranets, into the uncontrolled, large-scale and heterogeneous environment of the open semantic Web. In a search system for the open semantic Web, we should not expect complex reasoning over very expressive ontologies, because this requires detailed knowledge of ontology structure. However, systems that will balance mapping searches to multiple large ontologies while giving results in real time are feasible. This challenge has begun to be addressed by some matching techniques, particularly in the natural language area, but much remains to be done.

Combining these two dimensions divides the world of semantic search systems into four areas. We summarize each of these below, considering their state of development and what remains to be done.

6.1 Single mode—single domain

This is the area that has been addressed by the majority of the research done to date. Although incremental improvements are needed to perfect existing systems, the '80:20 rule' applies and gains will now be hard won. Specific areas where usability could be improved by technical advances are in matching, where lessons learnt from the natural language approaches we discussed, in Section 9, could be applied to other modes of interaction, and in ranking (e.g. Aleman-Meza *et al.*, 2005; Ding *et al.*, 2005).

6.2 Multiple mode—single domain

As we have already stated, this is an obvious route to follow to exploit the complementary affordances of the different search modes. Some existing search tools, for example, Corese and SEWASIE, offer a selection of modes, but the real challenge lies in integrating modes, so that users segue between them. Substantial gains in usability should be possible within a few years, with the right interface design. A key area to target will be the support of multimodal search refinement. Done properly, this should be able to support users in building searches and understanding the search space.

6.3 Single mode—heterogeneous

Research on heterogeneous systems will be driven by semantic Web developments and, given the difficulty of some of the problems that need to be tackled here, keyword and natural language systems will be built first, since heterogeneous scenarios are problematic for form- and view-based systems. The foundation will be matching. In heterogeneous environments, the problem of identifying the right entity is an order of magnitude more difficult than in single domains. There are two main approaches: formal ontology mapping (Euzenat & Shvaiko, 2007) and light weight mapping, as is used, for example, in PowerMap (Lopez, Sabou *et al.*, 2006). It may even be possible to exploit linguistic and domain knowledge from matching to help users to understand the intersections between different sources of semantic mark-up. Recommendation in heterogeneous environments may draw on the kind of implicit measures of user preferences, reviewed in Kelly (2003), such as browser behaviour and the content of user generated documents, which are being investigated for information retrieval on the Web.

6.4 Multiple mode—heterogeneous

The development of multiple-mode systems will follow on from single-mode systems, but we would not expect them to start appearing until there are robust solutions for the matching problem. A key problem for heterogeneous scenarios is that there is no obvious way to build form- and view-based systems, which are best suited to formulating parameterized searches. Multiple-mode systems may offer solutions. These will require innovative solutions for building forms and views on the fly to suit the user's search context.

Acknowledgements

This work was partly funded by the X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978, and partly funded by the Advanced Knowledge Technologies (AKT) Interdisciplinary Research Collaboration (IRC), sponsored by the UK Engineering and Physical Sciences Research Council under grant number GR/N15764/. We are grateful to Professor Dieng-Kuntz for permission to use the screenshot of INRIA's online demo of CORESE, and to Professor Qu and his colleagues at XObjects Group for permission to use the screenshot of Falcon_S.

References

Aleman-Meza, B., Halaschek-Weiner, C., Arpinar, I. B., Ramakrishnan, C. & Sheth, A. P., 2005 Ranking complex relationships on the semantic Web. *IEEE Internet Computing* **9**(3), 37–44.

Athanasis, N., Christophides, V. & Kotzinos, D. 2004 Generating on the fly queries for the semantic web: the ICS-FORTH graphical RQL interface (GRQL). In *3rd International Semantic Web Conference (ISWC'04)*. Hiroshima, Japan, pp. 486–501.

Bamba, B. & Mukherjea, S. 2005 Utilizing Resource Importance for Ranking Semantic Web Query Results, In *Semantic Web and Databases, LNCS Volume 3372/2005*. pp. 185–198.

Bernstein, A. & Kaufmann, E. 2006 GINO-a Guided Input Ontology Editor. In *Proceedings of the International Semantic Web Conference*. pp. 144–157.

Buscaldi, D., Rosso, P. & Sanchis Arnal, E. 2005 A WordNet-based queryexpansion method for geographical information retrieval. In *CLEF 2005 Workshop at GeoCLEF 2005*. Vienna, Austria.

Catarci, T., Di Mascio, T., Franconi, E., Santucci, G. & Tessaris, S. An ontology based visual tool for query formulation support. In *16th European Conference on Artificial Intelligence (ECAI-04)*. 2004. Valencia, Spain, pp. 308–312.

Cimiano, P. 2004 ORAKEL: A natural language interface to an f-logic knowledge base. In *9th International Conference on Applications of Natural Language to Information Systems (NLDB)*. pp. 401–406.

Corby, O., Dieng-Kuntz, R., Faron-Zucker, C. & Gandon, F. 2006 Searching the semantic web: approximate query processing based on ontologies. *IEEE Intelligent Systems* **21**(1), 20–27.

Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., Reddivari, P., Doshi, V. C. & Sachs, J. 2004 Swoogle: a search and metadata engine for the semantic web. In *Thirteenth ACM Conference on Information and Knowledge Management*. pp. 652–659.

Ding, L., Pan, R., Finin, T., Joshi, A., Peng, Y. & Kolari, P. 2005 Finding and ranking knowledge on the semantic web. In *4th International Semantic Web Conference*, *LNCS 3729 2005* pp. 156–170.

Euzenat, J. & Shvaiko, P. 2007 *Ontology Matching*, Berlin Heidelberg: Springer-Verlag.

Gruber, T. R. 1995 Towards principles for the design of ontologies used for knowledge sharing. *International Journal of Human-Computer Studies* **43**(5-6), 907–928.

Guha, R., McCool, R. & Miller, E. 2003 Semantic search. In *12th International Conference on World Wide Web*. pp. 700–709.

Hyvonen, E., Saarela, S. & Viljanen, K. 2003 Ontogator: combining view-and ontology-based search with semantic browsing. In *XML Finland 2003, Open Standards, XML, and the Public Sector*. Kuopio, Finland, pp. 82–85.

Kelly, D. & Teevan, J. 2003 Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum* **32**(2), 18–28

Lee, J. & Goodwin, R. 2005 The semantic webscape: a view of the semantic web. In *WWW'05: Special interest tracks and posters of the 14th International Conference on World Wide Web*. New York, pp. 1154–1155.

Lei, Y., Uren, V. & Motta, E. 2006 SemSearch: a search engine for the semantic web. In *15th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks (EKAW 2006)*. Podebrady, Czech Republic, pp. 238–245.

Lopez, V., Pasin, M. & Motta, E. 2005 AquaLog: an ontology-portable question answering system for the semantic web. In *2nd European Semantic Web Conference (ESWC 2005)*. Heraklion, Crete, Greece, pp. 546–562.

Lopez, V., Motta, E., Uren, V. 2006 PowerAqua: fishing the Semantic Web. In *European Sematic Web Conference (ESWC 2006)*. Budva, Montenegro, pp. 393–410.

Lopez, V., Sabou, M., & Motta, E. 2006 PowerMap: mapping the real semantic web on the fly. In *ISWC*. Athens, GA, USA, pp. 414–427.

Marchioni, G. 2006 Exploratory search: from finding to understanding. *Communications of the ACM*, **49**(4), 41–46.

Mc Guinness, D. 2004 Question answering on the semantic web. *IEEE Intelligent Systems* **19**(1), 82–85

Mihalcea, R. & Moldovan, D. 2005 Semantic indexing using wordnet senses. In *Proceedings of the ACL-2000 workshop on Recent advances in natural language processing and information retrieval: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics*. pp. 35–45.

Motta, E. & Sabou, M. 2006 Next generation semantic web applications. In *1st Asian Semantic Web Conference*. Beijing.

schraefel, m.c., Wilson, M., Russell, A., & Smith, D.A., 2006 mSpace: improving information access to multimedia domains with multimodal exploratory search. *Communications of the ACM* **49**(4), 47–49.

Shneiderman, B. 1998 *Designing the user interface: strategies for effective human-computer interaction*, Reading, MA; Harlow: Addison Wesley.

Sinha, V., Karger, D. R. 2005 Magnet: supporting navigation in semistructured data environments. In *2005 ACM SIGMOD International Conference on Management of Data*. Baltimore, Maryland, ACM Press, pp. 97–106.

Stuckenschmidt, H., Giunchiglia, F. & van Harmelen, F. 2005 Query Processing in Ontology-Based Peer-to-Peer Systems, In Tamma, V. Cranefield, S. Finin, T. Willmott, S. (eds.), *Ontologies for Agents: Theory and Experiences*, Birkhäuser. pp. 145–168.

Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E. & Ciravegna, F. 2006 Semantic Annotation for Knowledge Management: requirements and a Survey of the State of the Art. *Journal of Web Semantics* **4**(1), 14–28.

Vallet, D., Fernández, M. & Castells, P. 2005 The Quest for Information Retrieval on the Semantic Web. *Upgrade Journal, Monograph: The Semantic Web* **6**(6), 19–23.

White, R.W., Kules, B. & Bederson, B. 2005 *Exploratory Search Interfaces: Categorization, Clustering and Beyond. Report on the XSI 2005 Workshop at the Human-Computer Interaction Laboratory, University of Maryland. ACM SIGIR Forum* **39**(2), pp. 52–56.

Wu, H., Cheng, G., Qu, Y. 2006 Falcon-S: An ontology-based approach to searching objects and images in the Soccer domain. *Supplemental Proceedings of ISWC*, Nov. 2006.