# Hybrid Retrieval from the Unified Web

Trivikram Immaneni
Department of Computer Science and Engineering
Wright State University
3640 Colonel Glenn Hwy, Dayton, OH USA 45435
+1 937 775-5109
immaneni.2@wright.edu

Krishnaprasad Thirunarayan
Department of Computer Science and Engineering
Wright State University
3640 Colonel Glenn Hwy, Dayton, OH USA 45435
+1 937 775-5109
t.k.prasad@wright.edu

## ABSTRACT
The goal of Semantic Web initiative is to make the semantics of Web content accessible to machines. The Semantic Web has been evolving into a web of data separate from the existing HTML web. Our work focuses on establishing and exploiting connections between the two webs, especially hyperlink connections from the HTML web pages to the Semantic Web nodes, so as to enhance both data and document retrieval. We propose the Unified Web model to integrate the two webs, and a hybrid query language to retrieve data and documents from the Unified Web. Specifically, the query language amalgamates graph-based reasoning over RDF with keyword-based search.

## Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Search and Retrieval.

## General Terms
Languages, Theory.

## Keywords
Hybrid Query Language, Unified Web, Semantic Web, Semantic Search, Hierarchical Keyword Matching.

## 1. INTRODUCTION
The current web (the HTML Web) is a hyperlinked web of documents. The web browsers and the popular search engines provide convenient mechanisms to navigate, search, and retrieve information from the Web, thereby making the Web content human accessible. The Semantic Web (SW) is a labeled graph of resources and binary properties. The goal of SW initiative [1] is to "extend" the HTML Web to make the semantics of its content accessible to machines. But the SW has been evolving into a separate "web of data" parallel to the existing HTML web. The SW is built upon the Resource Description Framework (RDF) and its extensions. Database techniques have been extensively applied for storing and retrieving RDF data [2], and majority of the RDF query languages [3,4] resemble SQL (for example, SPARQL [4]).

One can incorporate documents into the SW by viewing them as data nodes and retrieving the documents using SPARQL queries (that enable RDF-graph traversal). In order to incorporate document *content* into the SW, we can encode it as string literal and use the regular expression matching features of SPARQL to retrieve it. But Data Retrieval (DR) via syntactic text matching ignores the context and the *semantics* of the document content and suffers from the well documented problems that Information Retrieval (IR) has been trying to address. On the other hand, making the semantics explicit by manual (re-)authoring of (legacy) documents employing SW formalisms such as RDF, OWL, etc, or by semi-automatically generating semantic annotations using the state-of-the-art NLP and Information Extraction techniques is infeasible in the general case.

Even if the web documents were to become a part of the SW (that is, their URLs and content occur in RDF triples), SPARQL-like query languages may be unsuited for human users ignorant of the underlying schema (such as exact URIs) for composing queries. Instead, the query language should be keyword-based with the provision to provide more precise information when available.

To address these issues, we propose the Unified Web (UW) model in Section 2 that encodes the two webs and the connections between them. The retrieval of data and documents on this UW is more effective than the separate data retrieval from the SW and document retrieval from the HTML web due to the exploitation of the connections between the two webs. We propose a hybrid, keyword-based query language for the UW in Section 3. It allows the users to explore the data and formulate more *precise* queries even when the schema information is not available. For document retrieval, it provides *convenient* keyword-based queries that can exploit available semantic information, especially the ISA relationships, for formulating *accurate* queries with explicit disambiguation information, and *expressive* queries for reasoning and "broadening" search. Section 4 describes related research. Section 5 concludes with suggestions for future work.

## 2. THE UNIFIED WEB MODEL
The Unified Web model aims to integrate the two separate worlds of the HTML web (documents) and the Semantic Web (data) into a single unified world and provide a framework for retrieving documents and data from it. Conceptually, the HTML Web is a graph with web documents as nodes connected by hypertext links. Likewise, the SW is a graph of resource nodes connected by property links. Recall also that, as it stands, the Semantic Web data is housed in HTML Web (RDF/XML documents), and the HTML Web documents can include URIs and URLs. Retrieval from the UW will exploit information on the two webs and marry the techniques developed for them to enable more effective retrieval of documents and data. The UW model consists of nodes and relationships between the nodes as discussed below.

### 2.1 Node
Node is an abstract entity that is uniquely identified by its URI. A Node may or may not have a document associated with it. But there is at least one node (and hence one URI) associated with a document. A document is a concrete container of information. A node can be seen as an abstract container that "contains" the following categories of information.

The "*Home URI"* section contains the textual representation of the URI of the node. Additionally, it contains a bag of words and phrases called "URI Index Words" or UIW constituted from various sources. For example, the words can be substrings of the URI, or come from the object literal of a triple whose subject is the URI and predicate is, say, *rdfs:label*. They can come from the anchor text of the URI in some document. For example, the hyperlink *<a href= "mailto:bsmith@wright.edu">William</a>* can contribute *William* to the UIW of the node whose URI is *mailto:bsmith@wright.edu*.

The *"Document"* section contains the textual representation of the document associated with the node (if any). The *"Parameters"* section contains information about the document such as filename, date of creation, etc., which is usually not a part of the document itself and should be obtained from the server serving the document. The *"External Text"* section contains fragments of text from other documents (for example anchor text) whose nodes participate in a *linksTo* relationship (to be discussed below) with the current node. This section may have words in common with the UIW. The *"Outgoing Links"* section contains URIs of nodes to which the current node has an outgoing *linksTo* link. The *"Triples"* section contains the textual representation of the RDF triples asserted by the node.

The above "container" of information associated with a node is the information that the retrieval system should keep track of. The "Home URI", the "Document" and the "External text" sections can be represented as bags of words. The "Outgoing links" and the "Triples" sections can be represented as a bag of URIs and a bag of triples respectively. All of these serve to "annotate" the node and can be used to index the node for retrieval.

The system assigns a number to each blank node/literal it encounters in a document. The URI of the document is concatenated with "#blnk" or "#lit" and with the number assigned to the node/literal. The resulting URI is assigned to the node/literal after conflicts are resolved. The "Home URI" section of a blank node contains its URI and an empty UIW. The "Home URI" section of the literal contains its URI and UIW based on the literal. The "Parameters" section of the literal contains the literal data type if any.

## 2.2 Relationships

The *asserts* relationship exists between a node and each of the RDF statements found in the associated document. For example, if the document *http://www.abc.com/xyz.htm* contains the following RDF fragment.

```
<rdf:RDF…>
<owl:Class rdf:ID='http://www.abc.com/sw#Jaguar'/>
</rdf:RDF>
```

Then, *asserts* relationship exists between *xyz.htm* and the statement *www.abc.com/xyz.htm#blnk1*. See Figure 1. Of course, the statement itself has a subject, a property and an object. For our purposes, the node asserts every triple that the parser extracts from the (RDF fragment of the) document that is associated with the node.

The *hasDocument* relationship exists between a node and a literal. The literal is the string representation of the document associated with the node. A *hyperlinksTo* relationship exists from a node A to another node B if there is a hyperlink from the document of node A to the document of node B. The *linksTo* relationship exists from node A to node B if a *hyperlinksTo* relationship exists from node A to node B, or node B occurs in any of the triples asserted by node A (see Figure 1).

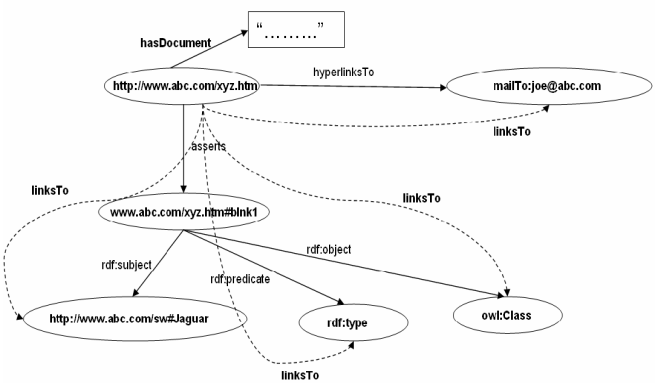


**Figure 1. Relationships**

The Unified Web is essentially an abstract model whose purpose is to encode the HTML web, the SW and the relationships between the two. The model can be implemented in different ways --- for DR, for IR, or for hybrid retrieval.

## 2.3 Implementing a DR System for UW

The UW model can be specified using RDF and the system can be implemented as an RDF database. A node is an instance of *rdfs:Resource*. The relationships *asserts*, *hasDocument*, *linksTo* and *hyperlinksTo* are instances of *rdf:Property*. The *asserts* property's domain is *rdfs:Resource* and its range is *rdf:Statement*. The *hasDocument* property's domain is *rdfs:Resource* and range is *rdfs:Literal*. The *linksTo* and *hyperlinksTo* are general properties – their domain and range are *rdfs:Resource*. The namespace of these properties can be the namespace of the system itself (for example, the fictional *http://www.system.org/web*).The relationships *rdf:Subject, rdf:Predicate,* and *rdf:Object,* naturally exist between an RDF statement and its components. The *asserts, linksTo, hyperlinksTo,* and *hasDocument* are called *system relationships.* Triples involving these relationships are the implicitly asserted *system triples*. These triples, along with those involving *rdf:Subject, rdf:Predicate, rdf:Object* form the UW.

The UW is the reified Semantic Web. Agents can reason with the data on UW. The RDF statements asserted by the resources (as opposed to the system) are called *user triples*, which form the conventional Semantic Web. The SPARQL queries for the SW can easily be transformed to SPARQL queries for the UW. Therefore, retrieving data from the UW using SPARQL queries is straight forward. Similarly, web documents appear as literals on the UW and SPARQL (regular expression) queries can be used to retrieve documents as data from the UW. Furthermore, an agent operating on the UW will have both the declarative knowledge and the indicative knowledge available to it, so the users can compose SPARQL queries to retrieve documents based upon the link structure (like WebSQL [5]).

## 2.4 Implementing an IR System for UW

The UW is a collection of nodes (with annotations) connected by links. An IR system implementing the model can index the nodes based upon the content of the various sections such as the words that describe its URI, the document, the URIs that it *linksTo*, the triples that it *asserts* and so on. At the time of retrieval, the system

can use any of the above annotations and the link structure to retrieve and rank nodes. The following section discusses how an IR system for the UW can exploit the *linksTo* and *asserts* information (the link between the HTML web and the SW) to use SW data to enhance document retrieval. Note that if the node happens to be an OWL ontology document, only the base URI will have a document associated with it and all the other nodes defined in it will be non-document nodes. See Figure 1 above.

### *2.4.1 Hyperlinks as Semantic Markup*

The SW is physically enclosed in web pages on the HTML web (as the RDF data is contained in files located on the Web). HTML markup tells the browser how to display a document. In contrast, semantic markup of content promotes its machine comprehension. Consider the following fragment from a document located at *http://www.one.com/A.html* that basically says that *B.html* is authored by John Smith.

```
<rdf:RDF…>
<rdf:Desc.  rdf:about="http://www.two.com/B.html">
<mydomain:author>  John  Smith  </mydomain:author>
</rdf:Desc.> </rdf:RDF>
```

The physical location of this fragment (that is, the file in which it resides) is irrelevant to the resource that it is describing. So, "description" (or metadata) is a better term to describe this fragment than "markup". There are systems that perform Semantic Web Document (SWD) retrieval on the Web viewing a document as a bag of URIs [6]. This is akin to retrieving databases (as opposed to data) from the web based upon their contents. This approach makes sense for searching for ontologies and SW data ("*retrieve documents that contain the URI XXX*"), but is not appropriate for document retrieval because the *location* of the semantic description has nothing to do with the document that it is describing. What is needed for this *bag of URIs* model to be effective for document retrieval is markup technology that physically ties in the semantic description of a document with the document being described.

Keeping the above discussion in mind, we propose an approach to improve document retrieval for legacy documents using SW data. We treat hyperlinks as semantic markup. A hyperlink from a document to a node on the SW links the document to the node and at the same time annotates the document with the URI of the node. On the UW, it is likely that there will be hyperlinks from HTML documents to resources that are part of the SW (that is, participate in triples). We propose that this valuable information be utilized to enhance document retrieval from the UW. For example, if a document contains a hyperlink to *mailto:bsmith@wright.edu*, and if there is a triple in the database that tells us that <*mailto:bsmith@wright.edu rdf:type univ:prof*> then this information can be used to enhance document retrieval. Specifically, a search for an instance of a *univ:prof* can uncover the document containing *mailto:bsmith@wright.edu.* Effectively, ISA relationship encoded in the SW can be used to broaden the search results. Thus, a hyperlink connecting an HTML page to the Semantic Web can be valuable from IR perspective.

Consider another example. On the web, it is not uncommon to see a document with hyperlinks from terms in the document to standard web pages (such as dictionary.com, Wikipedia, etc) that describe those terms.

"..The <*a href=*"*http://dictionary.com/search?q=jaguar*"> *J*aguar <*/a*> God of the Underworld …."

Here the hyperlink is from the term *Jaguar* to a webpage in an online dictionary [7] that describes/defines the term. The dictionary webpage can be said to *annotate* the term *Jaguar*. Similarly, on the UW, the author of a webpage can provide a hyperlink to the appropriate URI to annotate a term as illustrated below.

"..The <*a href = "http://www.animalOnto.com/Jaguar"*>Jaguar <*/a*>God of Underworld…."

This annotation is meant for machine agents rather than humans. This is a simple and elegant way of annotating a web page with SW data that can improve retrieval using the bag of URIs model. But it interferes with the human web navigation. To enable both human and machine consumption, we can use the combination.

<*a href ="http://dictionary.com/search?q=jaguar"*>Jaguar <*/a*> <*a href = "http://www.animalOnto.com/Jaguar"*> <*/a*>God of the Underworld….."

Here the empty hyperlink (rendered invisible by the browsers) next to *Jaguar* captures the sense of the term *Jaguar*. However, this approach is not viable for legacy documents because it requires physical modification of the documents (similarly to what is enabled by RDFa [8]). But, consider the following proposal of annotating the dictionary page defining *Jaguar* [7].

**jaguar** <*a href = "http://www.animalOnto.com/Jaguar"*> <*/a*> A large feline mammal *(Panthera onca)* of Central and South America, closely related to the leopard and having a tawny coat ...

The empty hyperlink annotation with *http://www.animalOnto.com/Jaguar* can explicitly state the animal sense and disambiguate it from the potential car or football team sense. Now, pages that hyperlink to this dictionary page can be inferred to be relevant to *Jaguar* the animal context and *http://www.animalOnto.com/Jaguar* can be considered to annotate those pages. In summary, by adding annotations to the pages in a *single* web site (for example, dictionary.com), we can annotate a host of legacy documents which link to the pages on the web site. This is an improvement over the previous approach but requires modification of the dictionary.com pages. We can further achieve scalability for extant legacy documents simply by adding the following triple to the IR system's database. <*http://dictionary.com/search?q=jaguar owl:Sameas http://www.animalOnto.com/Jaguar* >

This information can be used to conclude that the (unmodified) web pages linking to *http://dictionary.com/search?q=jaguar* (which is also unmodified) are talking about Jaguar, the animal. This idea can be extended to create ontology websites where each web page corresponds to an entity in the ontology. A user can annotate a document simply by adding a hyperlink to one of the pages in the web site.

A web page can be considered to have semantic annotation simply because it has a hyperlink to a Semantic Web data node or because it is linking to another web page that has explicit semantic annotations. Therefore, the existing hyperlink structure can be harnessed and used in conjunction with semantic descriptions to enhance document retrieval. The UW provides a framework where this is possible (due to the *linksTo* relationship). In essence, our approach is an application of the Pareto principle.

# 3. HYBRID QUERY LANGUAGE (HQL)

Our goal is to build a hybrid retrieval system based on UW that combines DR and IR paradigms. The goals of the system are: i) It should store and retrieve the SW data (*user triples*), and use information available in the documents to enhance data retrieval. ii) It should store and retrieve documents, and use available SW data to enhance document retrieval. The following description is informal in the interest of readability.

The challenge of retrieving information (documents or otherwise) from the UW is to design a query mechanism that allows users to harness structural information when available and rely on keyword-based searches when the structural information is not available. For example, to search for documents created by an individual named John in a typical RDF database, the users have to submit the following SPARQL query.

```
Select ?x   Where{
?x  http:/purl.org/../creator mailTo:john@abc.com}
```

What we want is to allow users to submit the query: *"?x creator John",* to accommodate lack of complete information for formulating unambiguous query involving *creator* or *John*. Specifically, several different ontologies or databases may define *creator* or *John*. However, if the user has more detailed information about what kind of *John* she is looking for, this should be expressible too, such as by specifying *John* is a person via *"?x creator person :: john"*. Again, the user is not really specifying *person* unambiguously. Furthermore, *John* can be direct instance of *person* or its descendent subclass. Contrast this IR-like approach to the DR-like approach in SPARQL that requires exact URI of the resources.

To summarize, we advocate a convenient keyword-based query language that can assist in formulating accurate queries with disambiguation information whenever possible. We now describe HQL, focusing on the main components, due to space constraints.

## 3.1 Word set queries

These queries allow users to search for nodes (URIs) based upon the words and phrases in their UIWs. A "word set" is a set of words and phrases (multiple words enclosed in quotes) enclosed in angular brackets. Given a word set, the system retrieves all the nodes in the UW such that all of the words in the word set appear in the node's UIW.

**Query:** *getNodes (<w1 w2 ... wn>)*

For example, let the Home URI of a node be *mailto:bsmith@microsoft.com.* Let this node be referenced from another HTML document

```
<a  href=mailto:bsmith@microsoft.com>  Research
Scientist </a>
```

Also, let the following triple be asserted by some node.

```
<mailto:bsmith@microsoft.com  rdfs:label  "William
Smith">
```

Then the UIW of the node will (perhaps) be: *{"bsmith" "microsoft" "Research Scientist" "Research" "Scientist" "William Smith" "William" "Smith"}*. This node will be retrieved by the query *getNodes(<Smith Research>),* but *not* by *getNodes(<Smith Research Bill>).* Thus multiple words inside angular brackets have implicit conjunction. A query can have multiple word sets (*ws*) separated by blank spaces. The blank space is an implicit disjunction and the answer is the union of the sets retrieved by each word set. The user can also explicitly search for literals or triples.

**Query:** *getNodes (ws1 ws2 ... wsn)*
**E.g.:** *getNodes(<Bill microsoft > <microsoft "William Smith">)*
**Query:** *getLiterals(<w1 w2 ... wn>)*
**Ans:** Literals in whose UIW all the words in the wordset appear.
**Query:** *getTriples (getNodes(<w1 w2 w3 ... wn>))*
**Ans:** Triples containing URIs retrieved by the inner query.

## 3.2 Hierarchical Keyword Matching

In order to deal with the problem of polysemy, the user can provide the system with disambiguation information available in an ontology to retrieve nodes. The keyword based search mechanism and the scope resolution operator to "connect" two word sets can permit the system to determine the relevant URIs. These novel queries are referred to as "word set pair" (*wsp*) queries, with the first of the pair referring to the class/superclass and the second of the pair to the instance/subclass.

**Query:** *getNodes(<w11 w21 ... wn1>* **::***<w12 w22 ... wm2>)*
**E.g.:** *getNodes(<person>::<john>)*

This would retrieve a node (URI) whose UIW contains "john" *and* which is a direct or indirect instance of a URI whose UIW contains "person". The user can place additional constraints by formulating conjunction queries with wordset pairs.

**Query:** *getNodes( wordset1::wordset2 AND wordset3::wordset2)*
**E.g.:** *getNodes(<person>::<john> AND <professor>::<john>)*

The user can formulate queries using *triplets* to explore the data. A *triplet* is a sequence of three word sets, word set pairs, URIs or variables (unknown quantities - prefixed with a *'?')* or any combination thereof. A *triplet* with no variables is "full triplet" and a triplet with one unknown quantity is called "partial triplet".

**Full Triplet: [** *ws/wsp ws/wsp ws/wsp* **]**
**E.g.:** *getTriples([*
*<"john smith" manager> <relationship>***::***< sonof> <steve>])*
**Ans:** Triples matching the above pattern.
**Partial Triplet: [***ws/wsp ws/wsp ?x***] , [***ws/wsp ?x ws/wsp* **], [***?x ws/wsp ws/wsp***]**
**E.g.:** *getNodes([<john manager><relationship>***::***< sonof> ?x ])*
**Ans:** Set of URIs binding to the variable *?x*.

A query can have several *partial* triplets separated by AND. These queries, called "partial triplet queries" or "answer extraction queries", enable composition of primitive relationships and thereby perform rudimentary reasoning via RDF graph traversal.

**Query:** *getNodes (partial triplet AND partial triplet)*
**E.g.:** *getNodes([<john><sonof>?x] AND[?x <wifeof><steve>])*

We now focus on queries aimed primarily at retrieving documents. The system uses the available semantic data to enhance document retrieval. For example, a user searching for Jaguar the animal can either type "Jaguar" or she can specify the kind of Jaguar she is interested in using "animal::Jaguar".

**Query:** *getLinkingNodes*(*URI)*
**Ans:** The URI of the node itself and the URIs of the nodes which have an outgoing *linksTo* link to the URI node. Note that the ontology documents are also retrieved here.
**Query:** *getAssertingNodes* (*Triple1 Triple2 Triple3…)*
**Ans:** URIs of the nodes that *assert* the triples.

**Query:** *getDocNodes( k1 … kn )*
**Ans:** Nodes whose document section contains keywords *k1…kn.*

The following high level constructs are designed to help users in document search by enabling them to combine information about the document with information within the document.

**Query:** *docSearch(ws/wsp/keywords)*
**E.g.:** *docSearch(ws1 ws2 … wsn k1 k2 … kn)*
**Ans:** Equivalent to *getLinkingNodes(getNodes(ws1 ws2…wsn)) INTERSECTION getDocNodes(k1 k2 … kn)*

Note that *getLinkingNodes(getNodes(ws…wsn))* retrieves Union of the nodes retrieved by the inner *getNodes* and the nodes that have an outgoing *linksTo* link to those nodes.

**Query:** *docSearch(<w11 w21…wn1>***::***<w12 w22…wm2> k1 k2 … kn)*
**Ans:** Similar to the above query.
**E.g.:** *docSearch(<animal>::<jaguar> Maya God)*
**Query:** *docSearch( ws1::ws2 AND ws3::ws4 k1 k2 … kn)*
**Query:** *docSearch ([ws/wsp ws/wsp ws/wsp] k1 k2… kn)*
**Ans:** Equivalent to *getAssertingNodes(getTriples(full triplet)) INTERSECTION getDocNodes(k1 k2 … kn)*

**Query:** *docSearch( [partial triplet] k1 k2 … kn)*
**Ans:** Intersection of the following sets - the set of nodes retrieved by the query *getNodes(partialTriplet)* and the set of nodes containing the keywords in their document section.

**Query:** *docSearch([partial triplet] AND [partial triplet] k1… kn)*
**Ans:** Similar to the above.

In addition to the above constructs, the user can search for documents by using the special keyword query (set of keywords).

**Query:** *k1 k2… kn*
The semantics of this query is equivalent to the following query.
*getLinkingNodes(getNodes(<k1 k2 … kn ><k1><k2>…<kn> ) ) UNION getDocNodes (k1 k2… kn)*

# 4. RELATED RESEARCH

There are many formal query languages [3,4] designed to query RDF data. *HQL* is different from them in that it is keyword-based and therefore brings in uncertainty along with user convenience necessitating ranking. We are applying IR techniques to a DR framework – the rationale being that the heterogeneity of the data warrants the trade-off and that exploration of the data will help users compose more accurate queries.

There are many systems that retrieve documents based upon their semantic annotations/descriptions [6,9,10,11,12,13,14]. Some of them provide hybrid languages which have both "formal" and keyword components to the users [11,13]. But in *HQL*, the user can formulate even the traditional RDF query (the "formal" component) using keywords in lieu of the exact URI. Another important distinction of this work is the suggestion that keywords used to index a URI can be derived from other HTML documents (anchor text for instance). Also, the simple and unique concept of hierarchical keyword matching (word set pairs) can be used to tackle ambiguity that plain keyword to URI matching suffers from. Another aspect that separates this work is our suggestion that *existing* HTML technology (without any enhancements) can be used to annotate documents and that existing *outgoing* hyperlinks can in *some cases* be treated as semantic markup.

# 5. CONCLUSION AND FUTURE WORK

We have presented the Unified Web model that integrates the Semantic Web and the HTML web, enabling exploitation of SW data to retrieve documents. In particular, we illustrated a scalable approach to semantic annotation of legacy documents using hyperlinks that improves both precision and recall of document retrieval. The query language *HQL,* to retrieve data and documents on the UW, is user-friendly because it is keyword-based, and is flexible and expressive because it provides a range of alternatives to formulate both exploratory browsing queries and research queries based on the available information with the user and in the SW. Specifically, the novel word set pair query enables formulation of more accurate queries using the ISA relationship.

We have a Java based in-memory implementation of the system, SITAR, that can hold around 1 million triples with 500M allocated to JVM. We are currently upgrading the system to store triples persistently. We are unable to discuss the implementation details due to space limitations. We are also working on a trust-based ranking algorithm that utilizes the *asserts* and *linksTo* information to rank URIs, triples and documents.

# 6. REFERENCES


[1] Semantic Web Activity, [Webpage], http://www.w3.org/2001/sw/.
[2] Beckett, D. SWAD-E Deliverable 10.2: Mapping Semantic Web Data with RDBMSes. [Online Document] 2003, http://www.w3.org/2001/sw/Europe/reports/scalable_rdbms_mapping_report/
[3] Bailey, J., Bry, F., Furche, T., and Schaffert, S. Web and Semantic Web Query Languages: A Survey. *Reasoning Web*, Eds., N. Eisinger and J. Maluszynski , Springer-Verlag, 2005.
[4] Prud'hommeaux, E., and Seaborne, A., Eds., SPARQL Query Language for RDF. [W3C Candidate Recommendation – Online document] April 2006, http://www.w3.org/TR/rdf-sparql-query/.
[5] Mendelzon, A., Mihaila, G., and Milo, T. Querying the World Wide Web. *Journal of Digital Libraries* vol. 1, no. 1, pp. 68-88, 1997.
[6] Ding, L., et al., Finding and Ranking Knowledge on the Semantic Web. In *Proc. of the 4th Intl. Sem. Web Conf.*, November 2005.
[7] Online Dictionary, [Website], http://dictionary.reference.com/.
[8] Adida, B., and Birbeck, M., Eds., RDFa Primer 1.0. [W3C Working Draft] May 2006, http://www.w3.org/TR/xhtml-rdfa-primer/
[9] Davies, J., Weeks, R., and Krohn, U. QuizRDF: Search Technology for the Semantic Web. In *Workshop on Real World RDF and SW Applications*, *Proc. of WWW'02*, Hawaii, USA, 2002.
[10] Guha, R., McCool, R., and Miller, E. Semantic search. In *Proc. of the 12th Intl. Conf. on World Wide Web*, May 2003.
[11] Mayfield, J., and Finin, T. Information retrieval on the semantic web: Integrating inference and retrieval. In *Proceedings of the SIGIR 2003 Semantic Web Workshop,* 2003.
[12] Rocha, C., Schwabe, D., and Aragao, M.P. A Hybrid Approach for Searching in the Semantic Web. In *Proceedings of the 13th Intl. World Wide Web Conference*, New York, May 2004, pp. 374-383.
[13] Zhang, L., Yu, Y., Zhou, J., Lin, C., Yang, Y. An enhanced model for searching in semantic portals. In *Proc. of the 14th Intl. World Wide Web Conference*, Chiba, Japan, NY: ACM Press, May 2005.
[14] Vallet, D., Fernández, M., and Castells, P. An Ontology-Based Information Retrieval Model. In *Proc. of 2nd European Semantic Web Conference (ESWC 2005)*, Berlin Heidelberg, 2005.