

Information Retrieval on the Semantic Web*

Urvi Shah and Tim Finin

University of Maryland Baltimore County
Baltimore MD 21250
{ushah1,finin}@cs.umbc.edu

James Mayfield

Johns Hopkins University
Applied Physics Laboratory, Laurel MD
james.mayfield@jhuapl.edu

Abstract

We describe an approach for information retrieval over documents that consist of both free text and semantically enriched markup. In particular, we present the design and implementation prototype of a framework in which both documents and queries can be marked up with statements in the DAML+OIL semantic web language. These statements provide both structured and semi-structured information about the documents and their content. Our approach allows inferencing to be done over this information at several points: when a document is indexed, when a query is processed and when query results are evaluated. Our current prototype is based on a version of the HAIRCUT information retrieval system and uses a semantic web inferencing system implemented using DAML-JessKB.

Keywords

Semantic Web, Text Extraction, Query-Answering Systems, Hybrid Information Retrieval

Introduction

We envision the future web as pages containing both text and semantic markup. Current information retrieval techniques are unable to exploit the semantic knowledge within documents and hence cannot give precise answers to precise questions. We cannot automatically extract such content from general documents yet. Manually structuring web documents, for example, with XML lets us retrieve more precise information using string and structure matching tools, such as the web robots Harvest, WebSQL, and WebLog. However, for this approach the user needs to be well aware of the structure of the documents, their exact names and forms, and hence is not scalable. Knowledge representation languages like DAML+OIL that support logic inferences can help us achieve more flexible and precise knowledge representation and retrieval. Industry is currently developing many metadata languages to let people index web information resources with knowledge representations (logical statements) and store them in web documents.

*This research was supported in part by DARPA contract F30602-97-1-0215.
Copyright © 2002, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

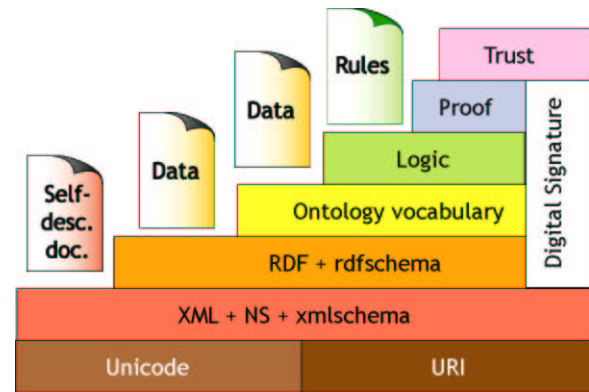


Figure 1: Tim Berners-Lee's vision of the Semantic Web

Web documents may contain free text along with some markup. There are many potential uses for annotation on the semantic web including workflow, image retrieval, database mediation and device interoperability. In this paper we will focus on three scenarios that involve semantically marked up web pages and text documents:

Information retrieval (IR) - e.g., identify and rank relevant pages or documents.

Simple question answering (Q&A) - e.g., who is the governor of Alaska?

Complex question answering - e.g., what is the current situation in Algeria?

Several general techniques have been developed to improve retrieval efficiency and performance. All three scenarios may involve some degree of reasoning and inference. Complex Q&A will often involve significant reasoning and summarization capabilities.

Background

DAML and the Semantic Web

The current web is primarily composed of pages with information in the form of natural language text and images intended for humans to view and understand. Machines are used primarily to render this information, laying it out on the screen or printed page. The idea behind the Semantic Web is

to augment these pages with markup that captures some of the meaning of the content on pages and encodes it in a form that is suitable for *machine understanding* (Hefflin, Hendler, & Luke 2001). This requires a new kind of markup languages; one that supports defining shared data models or ontologies for a domain and allows page authors to make statements using this ontology. The markup languages currently being used to explore this idea include RDF/S (RDF 1999; Lassila & (eds) 1999) and DAML+OIL (DAML 2001; Horrocks 2001).

The XML standard (Bray, Paoli, & Sperberg-McQueen 1998) provides the necessary means to declare and use simple data structures, which are stored in XML documents and which are machine-readable. However, since XML is defined only at the syntactic level, machines cannot be relied upon to unambiguously determine the correct meaning of the XML tags used in a given XML document. The W3C Consortium has developed RDF/S with the goal of addressing the XML deficiencies by adding formal semantics on the top of XML. These two standards provide the representation frameworks for describing relationships among resources in terms of named properties and values, which are similar to rudimentary frame languages such as RDF Schema. Yet, both standards are still very restricted as a knowledge representation language due to the lack of support for variables, general quantification, rules, etc.

The goal of DAML+OIL is to enable the transformation of the currently human-oriented web, which is largely used as a text and multimedia repository only, into a Semantic Web as envisioned by Berners-Lee (Berners-Lee & Fischetti ; Berners-Lee, Hendler, & Lassila 2001). It follows the same path for representing data and information in a document as XML, and provides similar rules and definitions to RDF/S. In addition, DAML+OIL also provides rules for describing further constraints and relationships among resources, including cardinality, domain and range restrictions, as well as union, disjunction, inverse and transitivity rules. DAML+OIL is, therefore, an effort to develop a universal Semantic Web markup language that is sufficiently rich to provide machines not only with the capability to read data but also with the capability to interpret and infer over the data. DAML+OIL will enable the development of intelligent agents and applications that can autonomously retrieve and manipulate information on the Internet and from the Semantic Web of tomorrow.

Information Retrieval in the World Wide Web

Though an active area of research for over thirty years, information retrieval (IR) has only become ubiquitous with the advent of the World Wide Web. The most familiar application of text retrieval is ad-hoc querying where a query is used to search a static set of documents. This is the task that commercial web search engines such as AltaVista and Google are best known for addressing. Search engines operate on huge databases and carry out a keyword search. In most cases precision is low, not all retrieved documents answer a user's query. For example, when a query "Who is the President of USA", was posed to Google, few links retrieved contained the name of the current President somewhere in

the document, but documents describing "How to campaign for becoming a President" and the President of a newspaper "USA TODAY" were also retrieved.

Intelligent Search Engines evolved as a descendent to Meta Search Engine, which incorporate machine-learning techniques. Information will no longer be intended for human readers only, but also for processing by machines, enabling intelligent information services, personalized web sites, and semantically empowered search engines. Knowledge can be annotated on the page in such a way that automatic tools can collect and understand it. Ontologies make possible that software agents can understand knowledge, which is marked up and further draw inferences pertaining to the domain of interest (Staab *et al.* 2000).

Agent Paradigm is a promising technology for information retrieval. Some applications are intelligent IR interfaces and clustering and categorization. An agent-based approach means that IR systems can be more scalable, flexible, extensible, and interoperable. Agents need a way to process and "understand" their information, both on the level of individual documents/objects as well as collection-wide entities. Statistical approaches, for deriving metadata from information, such as n-grams and latent semantic indexing are particularly interesting for analyzing text objects, because they are independent of the language of the text, are resistant to misspellings, and allow the application of many known mathematical techniques to natural language analysis.

Answering queries on the Web

Query-Answering Systems has been an area of research in different fields such as knowledge representation, databases, information retrieval, user interfaces, natural language. The advantage of these precision based system coupled with the scope of search engines led to efforts in scaling these systems to the web. START (Katz 1997) is one of the first QA systems with a web interface, having been available since 1993. Focused on questions about geography and the MIT InfoLab, START uses a precompiled knowledge base in the form of subject-relation-object tuples, and retrieves these tuples at run time to answer questions. AskJeeves (ask) is a commercial service that provides a natural language question interface to the web, but it relies on hundreds of human editors to map between question templates and authoritative sites. MULDER (Kwok, Etzioni, & Weld 2001) claims to be the first automated question-answering system that uses the full web as its knowledge base. MULDER utilizes several natural-language parsers and heuristics in order to return high-quality answers. Using this framework the MULDER system can be modeled to suit the requirements of answering queries on the web.

DAML queries

The purpose of ontologies and annotations on web pages is to enable a level of query capability and performance that is not available with current web search technology. RDQL is an implementation of an SQL-like query language for RDF. It treats RDF as data and provides query with triple patterns and constraints over a single RDF model. TRIPLE

is an RDF-based logic programming language for the Semantic Web at the Stanford University Database Group. TRIPLE's (Sintek & Decker 2001) layered architecture allows simple object-oriented extensions like RDF Schema, directly implemented with the extended Horn logic features and DAML+OIL type modules can be realized via interaction with external reasoning components.

DQL, a DAML+OIL Query Language, is a simple language for querying DAML+OIL knowledge bases. The language is specified as DAML+OIL ontology so that both queries and the results obtained from asking a query are represented in DAML+OIL. In order to query a DAML+OIL KB, one expresses a query in DAML+OIL and the results are returned in DAML+OIL. An instance of class Query represents a question posed to a reasoner. A query pattern is in affect a conjunction of one or more triples. Each triple corresponds to an RDF Statement except that its predicate, subject, and/or object can be a variable.

Related Work

The web is currently a distributed mass of simple hypertext documents. Large-scale web search engines effectively retrieve entire documents, but they are imprecise, because they do not exploit semantics of content. WebKB (Martin & Eklund 1999) is a tool, that interprets semantic statements stored in web-accessible documents. WebKB advocates the use of Conceptual Graphs and simpler notational variants for ontology and control commands, that enhance knowledge readability and lets its users combine lexical, structural, and knowledge-based techniques to exploit or generate web documents. In an operational context, these knowledge-based features need to be combined with more traditional information retrieval ideas that give both coarse-grained search capabilities and the fine-grained, precision-based knowledge retrieval.

Quest (Bar-Yossef *et al.* 1999) was designed and implemented for querying and manipulating documents written in the OHTML (Kogan *et al.* 1997) markup language. OHTML supports fine granularity semantic tagging of HTML pages. OHTML annotations are viewed as OEM objects, expressed in textual form within an HTML page. Quest uses the W3Lorel query language, based on the Lorel (Abiteboul *et al.* 1997) language to query the OEM objects (semantic view), as well as the hypertext view (HTML tags) of the document. Following this semi-structured approach, Quest allows for arbitrary tagging of HTML pages, offering flexibility to the user on the choice of semantic tags.

The SIGIR workshop on XML and information retrieval targeted various issues, which define the most relevant topics in the relation between these two technologies. ELIXIR (Chinenyanga & Kushmerick 2001), an Expressive and Efficient Language for XML Information Retrieval, extends the query language XML-QL (Deutsch *et al.* 1999) with a textual similarity operator. ELIXIR is a general-purpose XML information retrieval language, sufficiently expressive to handle querying XML documents both on content and structure. Based on the document-centric view of XML, XIRQL (Fuhr & Grojohann 2000) is an extension of XQL that supports IR-related features, which are weighting and

ranking, relevance-oriented search, data types with vague predicates, and semantic relativism. XIRQL integrates these features by using ideas from logic-based probabilistic IR models, in combination with concepts from the database area. For processing XIRQL queries, path algebra is presented, that also serves as a starting point for query optimization. XYZFind (Egnor & Lord) is a system for structured information retrieval using XML. It incorporates techniques for exploiting semantically structured XML to increase precision and recall and an extension to the classic inverted index to support structured Information Retrieval.

Design and Implementation of HOWLIR

There is a fundamental conflict between a person's view of the Semantic Web and a software agent's view of it that must be resolved for the Semantic Web to adequately support retrieval. Documents that contain semantic markup may be viewed in two ways. People will tend to view them as text documents that happen to contain some additional information that is not directly accessible or useful to them. Software agents on the other hand will view them as propositional stores over which to perform inference. These disparate views are incompatible on the surface. If they are not reconciled, they may lead to the development of a Semantic Web that is divorced from the current human-accessible Web.

To draw these disparate views together, and thereby increase the value of markup for people and the value of text for software agents, we argue that search and inference should be tightly bound. People will want to use the semantic Web to search not just for documents, but also for information about specific semantic relationships. Doing so will naturally require that inferences be drawn along the way. Software agents want to draw inferences on a topic of interest. Yet, because it is no longer practical to assume a monolithic knowledge base, drawing appropriate inferences necessitates rules and facts that will support the desired inferences.

There is a wide spectrum of techniques, which can be applied to address querying, and retrieval of semantically marked documents. HOWLIR is intended to provide a framework, which is able to extract and exploit the semantic information from these documents, perform sophisticated reasoning and filter results for better precision.

HOWLIR can be described in terms of two primary components: a set of ontologies and a hybrid information retrieval mechanism. HOWLIR defines ontologies encoded in DAML+OIL allowing users to specify their interests in different events. These ontologies are also used to annotate the event announcements. The information retrieval engine is based on the use of HAIRCUT. It employs text extraction, annotation, and inferencing mechanism, by utilizing the knowledge expressed in the ontologies. We now present the design and implementation aspects of every component of the framework.

EVENT Ontology

Ontology is an explicit specification of a representational vocabulary for a shared domain of discourse - definitions of

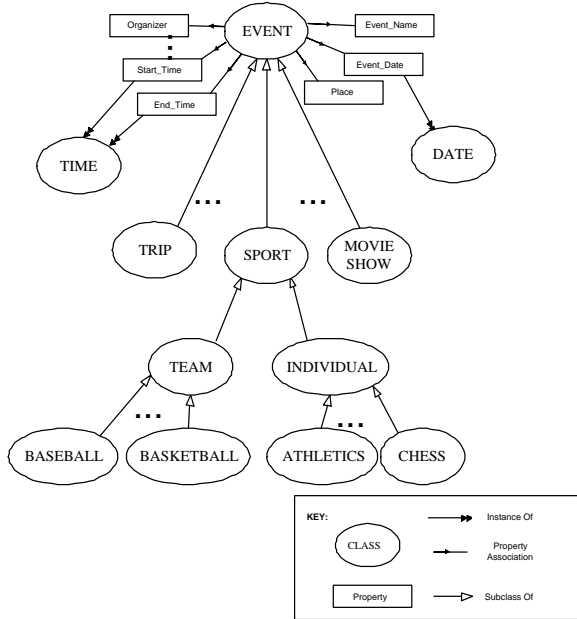


Figure 2: Snapshot of the Event Ontology

classes, relations, functions, constraints and other objects. Pragmatically, a common ontology defines the vocabulary with which queries and assertions are exchanged among software entities. Ontologies are not limited to conservative definitions, which in the traditional logic sense only introduce terminology and do not add any knowledge about the world. To specify a conceptualization we need to state axioms that put constraints on the possible interpretations for the defined terms.

Ontologies help to overcome the bottlenecks in information access, which is primarily based on keyword searches. They support information retrieval based on the actual content of a page and help navigate the information space based on semantic concepts. Ontologies enable advanced query answering and information extraction service, integrating heterogeneous and distributed information sources enriched by inferred background knowledge.

The main goal of our ontology development was to develop an ontology, which will help users interested in different events in the university, retrieve relevant information. The *Event Ontology*, an extension to ITTalks (Cost *et al.* 2001) is built, following the concept of “Natural Kinds OF” (Quine 1977) from the field of philosophy. We first identify the natural kinds in the phenomena under study, “EVENTS”, and then figure out what their most important characteristics are. The Event ontology we built is within a University domain. Event categories follow the natural kind of events that are prominent in a university e.g. movie showing, seminars, sport events etc. Events may be academic or non academic, free or paid, open or by invitation, but these describe the events in general and are not identifying characteristics of any particular type of event. Figure 2 sketches a snapshot of the event categories and properties defined in the ontology. An event announcement made within the campus is identi-

fied as an instance of one of the natural kind of events or subcategories. Instances of subcategories are inferred to be a subtype of one of the natural kind of events.

Text Extraction

Event announcements are currently in free text. We need these documents to contain semantic markup. We take advantage of the AeroText™ system for text extraction of key phrases and elements from free text documents. Document structure analysis supports exploitation of tables, lists, and other elements and complex event extraction to provide more effective analysis. Unlike other text processing systems, the system also provides tools for the end-user to customize the application and optimize its performance on site.

We use the domain user customization tool to fine-tune extraction performance. The extracted phrases and elements play a vital role in identifying type of events and adding semantic markup. The AeroText™ system has a Java API that is used to access an internal form of the extraction results. We have built DAML generation components that access this internal form, and then translate the extraction results into a corresponding RDF triple model that utilizes the DAML+OIL syntax. This is accomplished by referencing the Event ontology that directly correlates to the linguistic knowledge base used in the extraction process.

<p>AeroText™ Capabilities</p> <p style="text-align: center; color: red;">Who, what, when, where</p> <p>ABC123 Corporation to Donate to Charities</p> <p>ASSOCIATED PRESS Monday, Dec 4, 2000, 5:00 pm EST</p> <p>Philadelphia -----ABC123 Corporation Plans to donate \$10 million in cash and software to various charities, CEO Benjamin Romero said in Philadelphia on Monday. By the end of the year, 100 organizations will receive the technology they need. Mr. Romero said, "Last year the corporation gave more than \$25 million in cash to nearly 200 nonprofit organizations."</p>	<p>Proper Names</p> <p>Key Phrases</p> <p>Grammatical Phrases</p> <p>Entity Co-references</p> <p>Entity Association</p> <p>Event Extraction</p> <p>Topic Categorization</p> <p>Disambiguation</p> <p>Location Resolution</p> <p>Block Finder™</p> <p>Multilingual</p>
--	---

Figure 3: AeroText™ Capabilities

Inference System

HOWLIR uses the metadata information added during the text extraction process to infer additional semantic relations. These relations are used to decide the scope of the search and to provide more relevant responses. HOWLIR bases its reasoning functionality on the use of DAMLJessKB (Kopena). DAMLJessKB facilitates reading DAML+OIL files, interpreting the information as per the DAML+OIL language, and allowing the user to reason over that information. The software employs the SiRPAC RDF API to read in the DAML+OIL file as a collection of RDF triples and Jess (Java Expert System Shell) (Friedman-Hill 2000.) as a forward chaining production system to carry out the rules of the DAML+OIL language. Jess is a rule engine and scripting environment written in the Java language that can be used

to write applications that have the capacity to "reason" using knowledge supplied in the form of declarative rules. Jess uses the Rete (Forgy 1982) algorithm to process rules, a very efficient mechanism for solving the difficult many-to-many matching problem.

DAMLJessKB provides basic facts and rules that facilitate drawing inferences on relationships such as Subclasses and Subproperties. We enhance the existing inferential capabilities of DAMLJessKB and supplement it by filtering out facts that are of relevance to our system and applying domain specific rules. For example, DAMLJessKB does not import facts from the ontology that is used to create instances, thereby limiting its capacity to draw inferences. We have addressed this issue by importing the base Event ontology and providing relevant rules for reasoning over instances and concepts of the ontology. This combination of DAMLJessKB and domain specific rules has provided us with an effective inference engine.

The HAIRCUT Retrieval System

The Hopkins Automated Information Retriever for Combining Unstructured Text (HAIRCUT) (Mayfield, McNamee, & Piatko 1999) is a information retrieval system we have developed at the Johns Hopkins University Applied Physics Laboratory (JHU/APL). A language modeling approach to reasoning document similarity is used in lieu of traditional Boolean or vector-space models, a variety of tokenization schemes is supported, including overlapping character n-grams, and a novel term similarity measure is used to support various linguistic operations. The system is implemented in Java for ease of development and portability. We have further enhanced HAIRCUT for indexing DAML+OIL and RDF Triples, with or without wildcards. HAIRCUT allows the user to specify required, allowed and disallowed query terms. This gives the user flexibility in querying, at the same time increases precision. The combination of several complementary technologies in a single system makes HAIRCUT distinctive among retrieval systems.

Scanning gigabytes of data and creating index files that are used to support efficient retrieval is a computationally challenging area. Research into the performance issues of search has studied minimizing disk space, and optimizing both index construction and run-time query performance. Since the ordering of words is usually ignored when an index is built, the resulting data structures usually require less space than the documents. Depending on the algorithms and implementation, space of only 15-30% the size of the original text may be required.

Hybrid Information Retrieval

The addition of semantic markup to Web documents makes it possible to perform inference over document content. However, markup is also useful in another way. Traditional text retrieval characterizes documents by the indexing terms they contain. These indexing terms are typically words, but they need not be. One common variant is stemmed words; stems are words that have had suffixes removed to allow similar words (e.g., juggler, juggles and juggling) to be treated as a single indexing term. Less commonly used, but no less

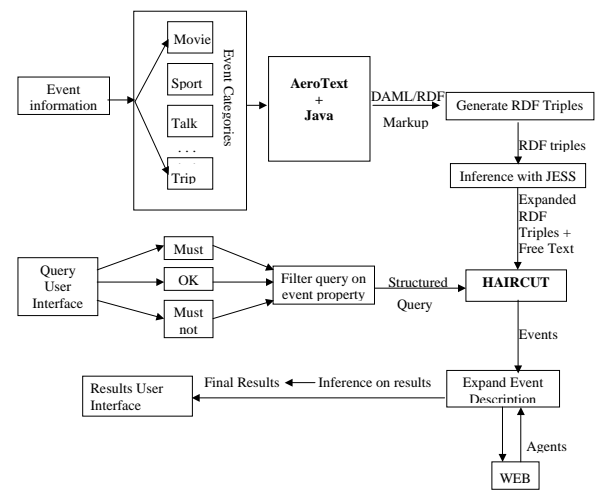


Figure 4: HOWLIR Process Flow

powerful, are characters n-grams, overlapping sequences of n contiguous characters. The efficacy of these different types of indexing terms demonstrates that traditional approaches to text retrieval can be effective over a variety of term types. This suggests that semantic markup, if present, might serve as indexing terms for a traditional IR engine. That is, in addition to indexing documents according to the text of their words, stems or n-grams, we might also index them according to the text of their semantic markup. We could, for example, treat each distinct DAML+OIL tag as an indexing term. Or, we might reduce document markup to RDF triples, and treat each distinct triple as an indexing term; this is our current approach.

By including semantic markup as indexing terms, we are exploiting the statistical associations between semantic markup and text. For example, given a way to find strongly associated indexing terms (e.g., through mutual information), we could suggest markup for a word or phrase by finding semantic markup that is strongly associated with it in an indexed collection. Alternatively, we could identify text that characterizes a given markup tag or triple; such text characterizations might serve as a basis for automated or semi-automated ontology mapping.

HOWLIR Process Flow

Information about events in the university is collected for retrieval and analyzed for Natural Kinds of Events. To make possible analysis and understanding of meaning of data by software, a population phase uses the Event ontology to describe the documents in DAML+OIL. Documents containing RDF triple patterns generated from the DAML+OIL markup and triples that are inferred through DAMLJessKB inference system, form the knowledge base of HAIRCUT. As an example see Figure 5 for documents structure.

Keeping different kinds of users who have varied preferences, we built a set of DAML queries similar to DQL Query, refer to Figure 6. Query provides one way in which the programmer can write a more declarative statement of

```

<DOC>
<DOCNO>http://gentoo.cs.umbc.edu/howlir/announcements/charity#charity_001
</DOCNO>
<TEXT>UMBC Blood Drive!!
Office of Student Life launches its annual Blood Drive for the Red Cross
on Mon, Nov 20 in the UC Ballroom from 10am - 4pm. </TEXT>
<TRIPLE>triple(charity_001)(
'http://gentoo.cs.umbc.edu/howlir/announcements/charity#charity_001_place',
'http://gentoo.cs.umbc.edu/ontologies/event_ont#Building',
'University Center').
triple(charity_001)(
'http://gentoo.cs.umbc.edu/howlir/announcements/charity#charity_001',
'http://gentoo.cs.umbc.edu/ontologies/event_ont#Organizer',
'Office of Student Life').
triple(charity_001)(
'http://gentoo.cs.umbc.edu/howlir/announcements/charity#charity_001_date',
'http://gentoo.cs.umbc.edu/ontologies/event_ont#Day_of_week',
'Monday').</TRIPLE>
</DOC>

```

Figure 5: Document containing Triples and free text in TREC format

what is wanted, and have HAIRCUT retrieve it. Taking advantage of the HAIRCUT feature which allows the user to specify which terms in the query MUST, MUSTNOT and MAYBE considered, each query is expressed as a document consisting of triples and free text. Syntactically an XML markup, the query pattern identifies the necessary and sufficient conditions for the search.

Logical systems provide provably good answers, but don't scale to large problems; an aspect that search engines can handle remarkably well. On the Semantic Web we can imagine a combination of a logical system coupled with the information retrieval engine. During search operation the HAIRCUT retrieval system retrieves all the documents that reference the terms used in the query, and then a logical system acts on that closed finite world of information to determine a reliable solution if one exists.

```

<Query>
<required>
triple(query_001)(
'http://gentoo.cs.umbc.edu/howlir/query#query_001',
'http://gentoo.cs.umbc.edu/howlir/ontologies/event_ont#Movie_Name'
'Ocean's Eleven').
</required>
<allowed>
</allowed>
<disallowed>
'http://gentoo.cs.umbc.edu/howlir/query#query_001',
'http://gentoo.cs.umbc.edu/howlir/ontologies/event_ont#Organizer'
'SEB').
</disallowed>
</Query>

```

Figure 6: DAML Query

Conclusion and Future Work

In this paper we have presented an approach to information retrieval over the Semantic Web utilizing a set of ontologies and inference engine. DAML+OIL is used as the knowledge representation language and as an interface for the inference engine, thus fostering flexibility and interoperability.

The powerful support in formulating rules, constraints and answering queries over schema information is far beyond existing database technology. The inference services are the equivalent of SQL query engines for databases; however provide stronger support (for example, recursive rules). Such inference service can be used to answer queries about explicit and implicit knowledge specified by the ontology thus provide a query answering facility that performs deductive retrieval from knowledge represented in DAML+OIL. HOWLIR framework advocates the interdependency of search and inference for precise retrieval over semantic content. Our initial work on this framework is promising, and we have presented a prototype within the context of university events.

The ongoing work is to build a sophisticated inference engine, which can develop an enhanced knowledge base from implicit and explicit inferences made from the DAML+OIL marked up documents. The framework encourages the role of Agents, that route information and share metadata, facilitate document and query expansion, and evaluate query results. HOWLIR framework can be expanded to realize intelligent message routing, wherein event announcements can be routed to the users based on their preferences, which are expressed as user profiles. User feedback can be utilized for better precision.

References

- Abiteboul, S.; Quass, D.; McHugh, J.; Widom, J.; and Wiener, J. 1997. The lorel query language for semistructured data. *International Journal on Digital Libraries 1* 68–88.
- Askjeeves. <http://www.askjeeves.com>.
- Bar-Yossef, Z.; Kanza, Y.; Kogan, Y.; Nutt, W.; and Sagiv, Y. 1999. Quest: Querying semantically tagged documents on the world wide web. In *In Proc. of the 4th Workshop on Next Generation Information Technologies and Systems*, volume NGITS'99.
- Berners-Lee, T., and Fischetti, M. *Weaving the web: The original design and ultimate destiny of the World Wide Web by its inventor*. San Francisco: Harper.
- Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. *The Semantic Web*. Scientific American.
- Bray, T.; Paoli, J.; and Sperberg-McQueen, C. 1998. Extensible markup language (xml). *W3C (Worldwide Web Consortium)*. <http://www.w3.org/TR/1998/REC-xml-19980210.html>.
- Chinenyanga, T., and Kushmerick, N. 2001. Elixir: An expressive and efficient language for xml information retrieval. In *SIGIR Workshop on XML and Information Retrieval*.
- Cost, R.; Finin, T.; Joshi, A.; Peng, Y.; Nicholas, C.; Chen, H.; Kagal, L.; Perich, F.; Zou, Y.; and Tolia, S. 2001. ITTALKS: A Case Study in the Semantic Web and DAML. In *International Semantic Web Working Symposium (SWWS)*.
- Darpa agent markup language. <http://www.daml.org>.

- Deutsch, A.; Fernandez, M.; Florescu, D.; Levy, A.; and Suciu, D. 1999. Xml-ql: A query language for xml. In *In Proc. 8th Int. World Wide Web Conference*.
- Egnor, D., and Lord, R. Structured information retrieval using xml. XYZFind Corporation, Washington, USA.
- Forgy, C. 1982 Rete: A Fast Algorithm for the Many Pattern/ Many Object Pattern Match Problem. *Artificial Intelligence 19* 17–37
- Friedman-Hill, E. 2000. Jess, the java expert system shell. (<http://herzberg.ca.sandia.gov/jess/>)
- Fuhr, N., and Grojohann, K. 2000. Xirql: An extension of xql for information retrieval. In *SIGIR Workshop on XML and Information Retrieval*.
- DAML+OIL Design Rationale 2001. www.cs.man.ac.uk/horrocks/Slides/index.html.
- Heflin, J.; Hendler, J.; and Luke, S. 2001. Shoe: A prototype language for the semantic webs. *Linkyping Electronic Articles in Computer and Information Science*. <http://www.ep.liu.se/ea/cis/1997/013/>.
- Kopena, J. <http://plan.mcs.drexel.edu/projects/legorobots/design/software/DAMLJessKB/>
- Katz, B. 1997. From sentence processing to information access on the world wide web. In *Natural Language Processing for the World Wide Web* 77–94. Papers from the 1997 AAAI Spring Symposium.
- Kogan, Y.; Michaeli, D.; Sagiv, Y.; and O.Shmueli. 1997. Utilizing the multiple facets of www contents. In *3rd Workshop on Next Generation Information Technologies and Systems*, volume NGITS'97.
- Kwok, C.; Etzioni, O.; and Weld, D. 2001. Scaling question answering to the web. In *Proceedings of WWW10*.
- Lassila, O., and (eds), S. R. R. 1999. Resource description framework (rdf) model and syntax specification. *W3C Recommendation*. <http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>.
- Martin, P., and Eklund, P. 1999. Embedding knowledge in web documents. In *Proceedings of World Wide Web Conference (WWW8)*.
- Mayfield, J.; McNamee, P.; and Piatko, C. 1999. The jhu/apl haircut system at trec-8. *The Eighth Text Retrieval Conference (TREC-8)* 445–452.
- Quine, W. V. 1977. *Naming, Necessity and Natural Kinds*. University Press. chapter Natural Kinds.
1999. *Resource Description Framework (RDF) Model and Syntax Specification*. www.w3.org/tr/rec-rdf-syntax.
- Sintek, M., and Decker, S. 2001. Triple-an rdf query, inference, and transformation language. *DDL P*. Japan.
- Staab, S.; Erdmann, M.; Maedche, A.; and Decker, S. 2000. An extensible approach for modeling ontologies in rdf(s). Technical Report 401, AIFB, University of Karlsruhe.