Ontology-Based Knowledge Discovery on the World-Wide Web[1]

Sean Luke*
seanl@cs.umd.edu
http://www.cs.umd.edu/~seanl/

Lee Spector* †
lspector@hampshire.edu
http://hampshire.edu/~lasCCS/

David Rager*
rager@cs.umd.edu
http://www.cs.umd.edu/~rager/

*Department of Computer Science
University of Maryland
College Park, MD 20742

†School of Cognitive Science and Cultural Studies
Hampshire College
Amherst, MA 01002

Abstract

This paper describes SHOE, a set of Simple HTML Ontology Extensions. SHOE allows World-Wide Web authors to annotate their pages with ontology-based knowledge about page contents. We present examples showing how the use of SHOE can support a new generation of knowledge-based search and knowledge discovery tools that operate on the World-Wide Web.

Introduction

Imagine that you are searching the World-Wide Web for the home pages of a Mr. and Mrs. Cook, whom you met at a conference last year. You don't remember their first names, but you *do* recall that both work for an employer associated with the massive ARPA funding initiative 123-4567. This would certainly be sufficient information to find these people given a reasonably structured knowledge base containing all of the relevant facts. At first this also seems like enough information to find their home pages by searching the World-Wide Web, but you soon discover otherwise.

Using an existing man-made web catalog, you can find ARPA's home page but learn that hundreds of subcontractors and research groups are working on initiative 123-4567. Searching existing web indices for "Cook" yields thousands of pages about cooking, and searching for "ARPA" and "123-4567" provides you with hundreds and hundreds of hits about the popular initiative. Unfortunately, searching for all of them together yields nothing: apparently neither person lists the initiative on his or her web page. Just wandering through the Web is fruitless. What can you do?

This scenario is common to many people on the World-Wide Web. A major problem with searching on the Web today is that data available on the Web has little semantic organization beyond simple structural arrangement of text, declared keywords, titles, and abstracts. As the Web expands exponentially in size, this lack of organization makes it very difficult to efficiently glean knowledge from the Web, even with state-of-the-art natural language processing techniques, index mechanisms, or the assistance of an army of data-entry workers assembling hand-made Web catalogs. In short, there is no effective way use the World-Wide Web to answer a query like:

```
Find web pages for all x, y, and z such that
    x is a person, y is a person,
    z is an organization where
        lastName(x,"Cook") and
        lastName(y,"Cook") and
        employee(z,x) and
        employee(z,y) and
        marriedTo(x,y) and
        involvedIn(z,"ARPA 123-4567")
```

Searching the Web

The chief intent of HTML and HTTP is to assist user-level presentation and navigation of the Internet; automated search or sophisticated knowledge-gathering has been a much lower priority. Given this emphasis, relatively few mechanisms have been established to allow documents to be indexed with useful semantic information beyond document-oriented information like "abstract" or "table of contents". As a result, most common indexing mechanisms for the World-Wide Web have generally fallen into one of three categories:

- Keyword subject indices.
- Catalogs painstakingly built by hand.
- Private robots using ad-hoc methods to gather limited semantic information about pages (like "Everyone with links to me" or "All broken page links").

Each approach has disadvantages. Keyword indices suffer because they associate the semantic meaning of web pages with actual *lexical* or *syntactic content*. Using our previous example, if we were looking for a woman whose last name was Cook, searching a keyword index under "Cook" yields tremendous numbers of web pages, almost none of which are about living people named Cook. "Cook" has many uses besides being a last name.

[1] To appear in the Proceedings of the Workshop on Internet-based Information Systems, AAAI-96

On the other hand, a major disadvantage of hand-built catalogs is the man-hours required to construct them. Given the size of the World-Wide Web, and the rate at which it is growing, cataloging even a modest percentage of web pages is a Herculean task. Additionally, the criteria used in building any catalog may turn out to be orthogonal to those of interest to a user.

Lastly, ad-hoc robots that attempt to gather semantic information from the web typically gather only the limited semantic information inferable from existing HTML tags. The current state of natural language processing technology makes it difficult to infer much semantic meaning from the body text itself at a reasonable rate (if at all). We have examined and developed web-wandering robots equipped with ad-hoc machinery for specialized searching tasks: recognizing and cataloging computer science web pages, for example. Unfortunately, even a small topic like this proves surprisingly difficult to implement, and like many ad-hoc methods, these robots' algorithms are extremely brittle.

Further, none of these approaches (except perhaps the last, for specific domains) allows for inferences about relationships *between* web pages, aside from simple facts about linkage. Sophisticated queries such as our initial example ("Find a man and a woman married to each other, whose last name is 'Cook', and who both work an organization involved with ARPA Initiative 123-4567") are therefore clearly out of reach.

Solution: Adding Semantics to HTML

Instead of trying to glean knowledge from existing HTML, another approach is to give HTML authors the ability to embed knowledge directly into HTML pages, making it simple for user-agents and robots to retrieve and store this knowledge. The straightforward way to do this is to provide authors with a clean superset of HTML that adds a knowledge markup syntax; that is, to enable them to use HTML to directly classify their web pages and detail their web pages' relationships and semantic attributes in machine-readable form.

Using such a language, a document could claim that it is the home page of a graduate student. A link from this page to a research group might declare that the graduate student works for this group as a research assistant. And the page could assert that "Cook" is the graduate student's last name. These claims are *not* simple keywords; rather they are semantic tags defined in an "official" set of attributes and relationships (an *ontology*). In this example the ontology would include attributes like "lastName", classifications like "Person", and relationships like "employee". Systems that gather claims about these attributes and relationships could use the resulting gathered knowledge to provide answers to sophisticated knowledge-based queries.

Moreover, user-agents or robots could use gathered semantic information to refine their web-crawling process. For example, consider an intelligent agent whose task is to gather web pages about cooking. If this agent were using a thesaurus-lookup or keyword-search mechanism, it might accidentally decide that Helena Cook's web page, and pages linked from it, are good search candidates for this topic. This could be a bad mistake of course, not only for the obvious reasons, but also because Helena Cook's links are to the rest of the University of Maryland (where she works). The University of Maryland's web server network is very very large, and the robot might waste a great deal of time in fruitless searching. However, if the agent gathered semantic tags from Helena Cook's web page which indicated that Cook was her last name, then the agent would know better than to search this web page and its links.

Related Work

HTML 2.0 (Berners-Lee and Connolly 1995) already includes several weak mechanisms for semantic markup (the REL, REV, and CLASS subtags, and the META tag). HTML 3.0 (Ragget 1995) advances these mechanisms somewhat, though it is not yet an official standard. Unfortunately, the semantic markup elements of HTML have so far been used primarily for document meta-information (such as declared keywords) or for hypertext-oriented relationships (like "abstract" or "table of contents"). Furthermore, relationships can only be established along hypertext links (using `<LINK>` or `<A>`). It appears that the intent of HTML's existing set of semantic markup tags is only to provide semantics that assist hypertext applications or other document-oriented functions.

To address some of these problems, Dobson and Burrill (1995) have attempted to reconcile HTML with the Entity-Relationship (ER) database model. This is done by adding to HTML a simple set of tags that define "entities" within documents, labelling sections of body text as "attributes" of these entities, and defining relationships from an entity to outside entities. Documents may contain as many entities as necessary. Dobson and Burrill associate with each entity a unique key, and establish relationships not between URL links but between keys.

Although Dobson and Burrill's ER scheme is a significant improvement over HTML's existing mechanism, it does not provide for any *ontological* declarations. For example, their scheme does not give any clear mechanism for classification through an "is a" hierarchy of classes. Yet one of the most significant uses for semantics in documents is to categorize them according to some classification scheme or taxonomy. For example, paper documents are often classified using hierarchical schemes like the Library of Congress subject headings, the Dewey Decimal system, or Universal Decimal Classification. Similarly, a good semantics mechanism for World-Wide Web documents needs the ability to do flexible, hierarchical classification. The ability to establish relationships between WWW entities is important, but secondary to the ability to classify those entities.

Moreover, the ER scheme does not allow one to specify inferences that can be drawn from relationships given in web pages. Even simple specifications such as transitive closure inferences can be helpful: if Helena Cook's home page claims that she works for the PLUS research group, and this research group is part of the Computer Science Department, part of the College of Computer, Mathematical, and Physical Sciences, part of the University of Maryland at College Park, part of the University of Maryland at College Park, part of the State of Maryland, she should not have to declare that she works for *all* of these entities; such a fact should be inferable. Invertable relationships are also useful: if George Cook is known to be married to Helena Cook, the inverse should be automatically inferable, without George or Helena having to say it. Through the addition of more powerful inferential rule capabilities, full knowledge base semantics could be provided.

Several advances will be required to provide full knowledge-base semantics on the World-Wide Web. Although the knowledge representation literature describes many systems that could be adapted to this purpose, unique features of the Web will mandate significant changes. For example, assertions on the Web will be made by many different people with differing authority to make such assertions. These assertions must therefore be interpreted as *claims*, of which the authorship is a significant part. In addition, the distributed nature and unknown correctness of knowledge on the Web poses new challenges. The work described in this paper is a first step in the process of solving these problems to provide full knowledge-base semantics for World-Wide Web contents.

A SHOE Overview

We present here an introduction to a small superset of HTML that provides many of these mechanisms. This scheme is called SHOE: Simple HTML Ontology Extensions. Among other things, SHOE provides the ability to:

- Define ontologies using HTML.
- Declare entities for both whole documents and for document subsections
- Declare relationships between entities.
- Declare entity attributes.
- Classify entities under an "is a" classification scheme.

The full specification of this language is located at *http://www.cs.umd.edu/projects/plus/SHOE/spec.html.* The specification does not as yet provide inferential rules other than transitive "is a" classification, but is designed to be consistent with such rules when they are added later. The specification adds the following tags to HTML:

Specifying Ontologies

- `<ONTOLOGY ... > ... </ONTOLOGY>`
  Declares a new ontology.
- `<ONTOLOGY-EXTENDS ... >`
  Indicates that our ontology extends another ontology.
- `<ONTDEF ... >`
  Defines a relation, an "is a" classification, or a renaming rule.

Annotating an HTML Document Using One or More Ontologies

- `<USE-ONTOLOGY ... >`
  Indicates that the document uses one or more ontologies.
- `<META ... >`
  Used to declare the document as an entity.
- `<INSTANCE ... > ... </INSTANCE>`
  Declares a subsection of a document to be an "instance" (an entity).
- `<CATEGORY ... >`
  Classifies an instance under one or more classes (categories).
- `<RELATION ... >`
  Declares a relation between entity instances or between an instance and data.
- `<ATTRIBUTE ... > ... </ATTRIBUTE>`
  An alternative mechanism for declaring a relation between an entity instance and data: the data in question is the body text between the two attribute tags.

A Detailed Example

To illustrate SHOE, we'll annotate the home page of George Cook (Helena Cook's husband). This example does not describe all the capabilities of our specification, but gives a taste of much of it. Before we can annotate George's home page, we need an ontology that:

- Provides a "Person" classification
- Provides an "Organization" classification
- Provides the "marriedTo" relationship between people
- Provides the "firstName" and "lastName" attributes for people
- Provides the "employee" relationship between organizations and people

For the sake of this example we'll build a new ontology that provides some of the necessary classifications and relationships. Ordinarily we wouldn't have to do this; instead, we'd rely on existing ontologies from common libraries on the web. Such ontologies

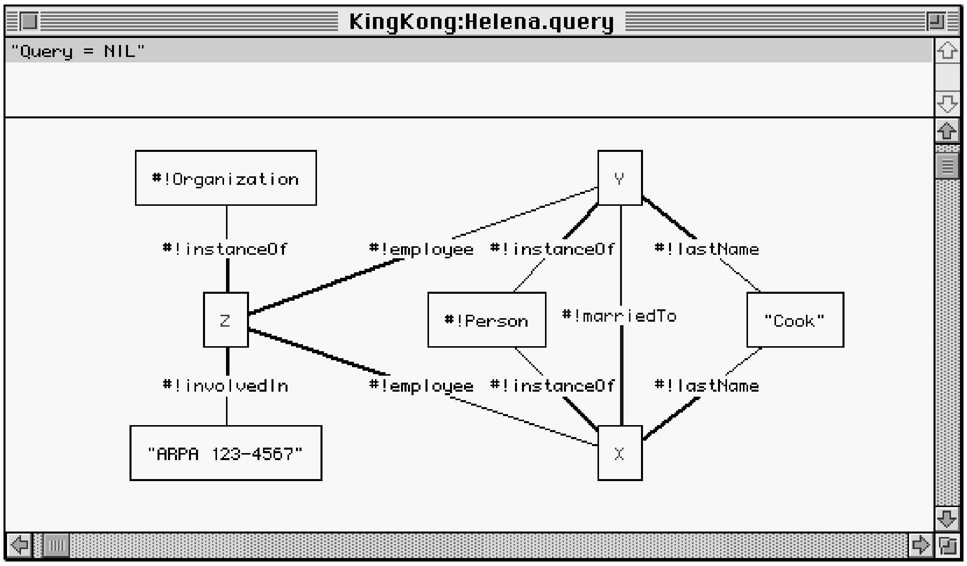

Figure 1: A Knowledge-based World Wide Web Query in PARKA

will offer a unified structure for sharing knowledge on the World-Wide Web.

Let's assume there already exists an ontology called `organization-ontology` version 2.1 which defines the classifications `Organization` and `Thing`, and that this particular ontology is available at *http://www.ont.org/orgont.html*. We'll *extend* the `organization-ontology` ontology to include our other needed classifications and relationships. Namely, we'll borrow `Organization` directly, and when we define `Person` we'll claim that `Person` "is a" `Thing`. Let's call our extension the `our-ontology` ontology, version 1.0. We write our new ontology as a piece of HTML:

```
<ONTOLOGY "our-ontology" VERSION="1.0">
<ONTOLOGY-EXTENDS "organization-ontology"
  VERSION="2.1" PREFIX="org"
  URL="http://www.ont.org/orgont.html">
<ONTDEF CATEGORY="Person" ISA="org.Thing">
<ONTDEF RELATION="lastName"
  ARGS="Person STRING">
<ONTDEF RELATION="firstName"
  ARGS="Person STRING">
<ONTDEF RELATION="marriedTo"
  ARGS="Person Person">
<ONTDEF RELATION="employee"
  ARGS="org.Organization Person">
</ONTOLOGY>
```

This indicates that `Person` is a subcategory of `Thing` as defined in the `organization-ontology` ontology, that people have first and last names which are strings, that people can be married to other people, and that people can be employees of organizations. These tags are embedded in an HTML document, which in turn might be promulgated as an "official" person-relationships ontology.

The "official" location of our ontology is the HTML document at *http://ont.org/our-ont.html*. George Cook can now use this ontology to describe his home page. Assume that, using this ontology, Helena Cook's page has already been classified as a `Person`, and that its unique key is the same as its official URL: *http://www.cs.umd.edu/~helena*. Furthermore, the place Helena and George work for, the University of Maryland's Computer Science Department, has its home page classified as an `Organization`, and that its unique key is the same as its official URL: *http://www.cs.umd.edu*.

To annotate George's home page, we begin by assigning his home page a key that is the same as its official URL: *http://www.cs.umd.edu/~george*. In the HEAD section of George's web page, we add:

```
<META HTTP-EQUIV="Instance-Key"
  CONTENT="http://www.cs.umd.edu/~george">
<USE-ONTOLOGY "our-ontology"
  VERSION="1.0" PREFIX="our"
  URL="http://ont.org/our-ont.html">
```

This declares George's web page to be a data entity with a unique key, and indicates that it will use the ontology `our-ontology` to describe itself. Furthermore, every time elements from `our-ontology` are used, they will be labelled with the prefix `our`.

In the BODY section we now declare facts about George's home page, namely George's name, that George is a person, that he is married to Helena, and

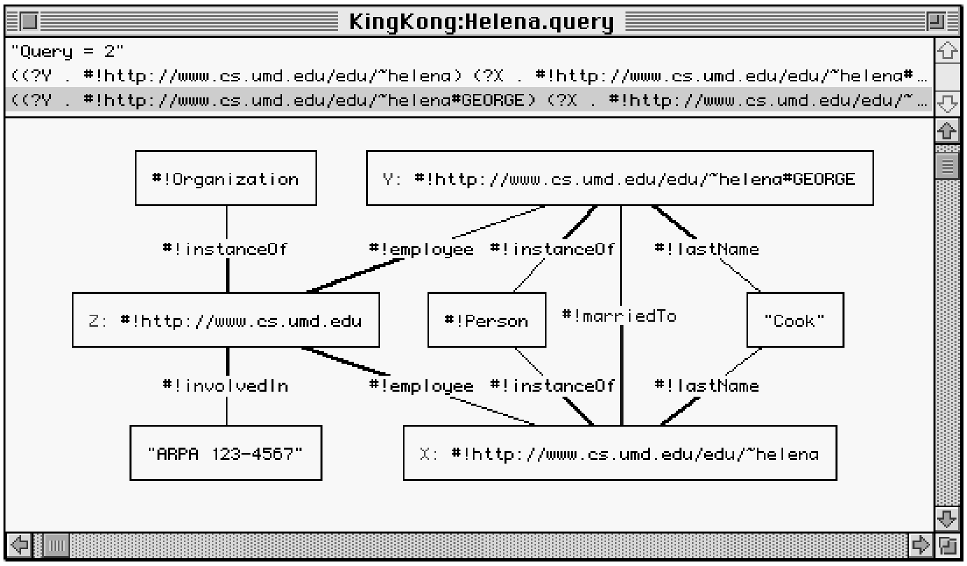

Figure 2: Query Results

that he works for the University of Maryland's Computer Science Department:

```
<CATEGORY "our.Person">
<RELATION "our.firstName" TO="George">
<RELATION "our.lastName" TO="Cook">
<RELATION "our.marriedTo"
  TO="http://www.cs.umd.edu/~helena">
<RELATION "our.employee"
  FROM="http://www.cs.umd.edu">
```

The category declaration indicates that George is a `Person`. The first two relations declare that George's name is "George Cook". The next relation declares that George is married to Helena. The last relation declares the relationship `employee` *from* George's employer *to* George.

Alternatively, if George's name were mentioned in the text of his home page, we could replace the first two relation declarations with something like:

```
My name is <ATTRIBUTE "our.firstName">
  George </ATTRIBUTE>
  <ATTRIBUTE "our.lastName"> Cook
  </ATTRIBUTE> and I live at...
```

If George didn't have his own web page but instead resided on a small part of his wife's web page, it would still be possible to provide George with his own unique identity and describe these relationships. In this case, we'll use *http://www.cs.umd.edu/~helena#GEORGE* as George's unique key. We add to the HEAD section of his wife's web page (if it's not already there):

```
<USE-ONTOLGY "our-ontology"
  VERSION="1.0" PREFIX="our"
  URL="http://ont.org/our-ont.html">
```

And in the BODY section we declare George to be an entity instance by adding (near the section on Helena's page that deals with George):

```
<INSTANCE
  "http://www.cs.umd.edu/~helena#GEORGE">
<CATEGORY "our.Person">
<RELATION "our.firstName" TO="George">
<RELATION "our.lastName" TO="Cook">
<RELATION "our.marriedTo"
  TO="http://www.cs.umd.edu/~helena">
<RELATION "our.employee"
  FROM="http://www.cs.umd.edu">
</INSTANCE>
```

Applications

At the University of Maryland at College Park, we are developing a web-crawling robot, Exposé, which parses SHOE-enabled HTML documents and adds claims to its internal knowledge-base. Exposé runs on Macintosh Common Lisp or C, using PARKA (Evett, Anderson, and Hendler 1993), University of Maryland's massively-parallel semantic network system, for its knowledge representation. We can then use this knowledge to answer sophisticated queries about these documents and their relationships

For example, after Exposé has gathered claims from Helena Cook's web page, we can query PARKA to find her and her husband. Figure 1 shows the query we in-

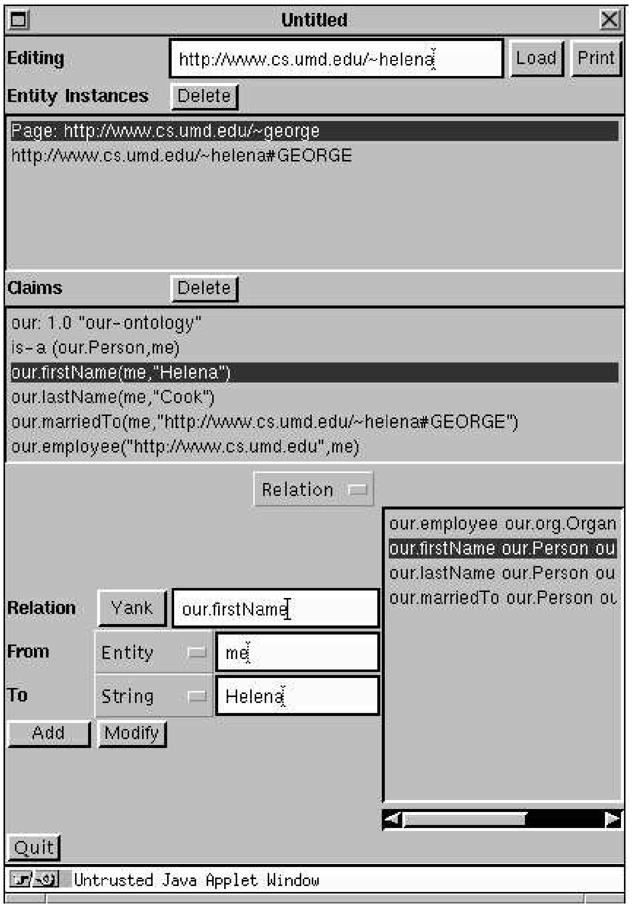

Figure 3: Graphically Annotating Helena's Web Page

troduced in the beginning of this paper, as laid out using PARKA's Graphical Query mechanism. This is the equivalent of querying PARKA with:

```
(query!  '(:and
    (#!instanceOf ?X #!Person)
    (#!instanceOf ?Y #!Person)
    (#!instanceOf ?Z #!Organization)
    (#!lastName ?X "Cook")
    (#!lastName ?Y "Cook")
    (#!employee ?Z ?X)
    (#!employee ?Z ?Y)
    (#!marriedTo ?X ?Y)
    (#!involvedIn ?Z "ARPA 123-4567")))
```

In Figure 2, PARKA has filled in the variables with actual results — selecting Helena fetches her web page directly.

We are also developing Java applications to make it easier for users to annotate web pages with semantic knowledge and to query robot servers using SHOE. For example, our graphical annotator is shown in Figure 3, assisting in embedding semantic knowledge into Helena's web page. In conjunction with this effort, we are investigating the use of knowledge-representation standards like KQML (Finin et al. 1994) and KIF (Genesereth and Fikes 1992) to facilitate communication between clients and servers in retrieving results, or between servers and slave servers in building up results from a number of sources.

Future Work

Although we feel our current specification provides much of the expressiveness needed for more advanced World-Wide Web agents, it still lacks many features found in sophisticated knowledge-representation systems. We are adding such features conservatively, seeking a compromise that provides some of the power of sophisticated knowledge representation tools while keeping the system simple, efficient, and understandable to the lay HTML community.

For example, the current specification does not yet provide for annotations that allow inference of transitive closure, negation, or inverted (reversed) relations. We are currently working to refine a small set of tags that will be easy for HTML authors to understand while allowing agents to use these inferences to derive useful new facts from the basic claims made in HTML pages. The knowledge representation literature provides many insights into the design of such tags, but the unique demands of the World-Wide Web (such as the distribution of knowledge and the varying authority of authors) require that this literature be examined in a new light.

Conclusion

The Web is a disorganized place, and it is growing more disorganized every day. Even with state-of-the-art indexing systems, web catalogs, and intelligent agents, World-Wide Web users are finding it increasingly difficult to gather information relevant to their interests without considerable and often fruitless searching. Much of this is directly attributable to the lack of a coherent way to provide useful semantic knowledge on the Web in a machine-readable form.

SHOE gives HTML authors an easy but powerful way to encode useful knowledge in web documents, and it offers intelligent agents a much more sophisticated mechanism for knowledge discovery than is currently available on the World-Wide Web. If used widely, SHOE could greatly expand the speed and usefulness of intelligent agents on the web by removing the single most significant barrier to their effectiveness: a need to comprehend text and graphical presentation as people do. Given the web's explosive growth and its predominance among Internet information services, the ability to directly read semantic information from HTML pages may soon be not only useful but necessary in order to gather information of interest in any reasonable amount of time.

Acknowledgements

We are grateful to Dr. James Hendler for his assistance in the development of this paper.

This research was supported in part by grants from NSF(IRI-9306580), ONR (N00014-J-91-1451), AFOSR (F49620-93-1-0065), the ARPA/Rome Laboratory Planning Initiative (F30602-93-C-0039), the ARPA I3 Initiative (N00014-94-10907) and ARPA contract DAST-95-C0037.

References

Dobson, S.A. and V.A. Burrill. 1995. Lightweight Databases. In *Proceedings of the Third International Worldwide Web Conference (special issue of Computer and ISDN Systems).* v. 27-6. Amsterdam: Elsevier Science. URL: *http://www.igd.fhg.de/www/www95/papers/54/darm.html* See also: *http://www.cis.rl.ac.uk/proj/www/docs/lightweight/index.html*

Evett, M.P., W.A. Andersen, and J.A. Hendler. 1993. Providing Computational Effective Knowledge Representation via Massive Parallelism. In *Parallel Processing for Artificial Intelligence.* L. Kanal, V. Kumar, H. Kitano, and C. Suttner, Eds. Amsterdam: Elsevier Science Publishers. URL: *http://www.cs.umd.edu/projects/plus/Parka/parka-kanal.ps* See also: *http://www.cs.umd.edu/projects/plus/Parka/*

Finin, T., D. McKay, R. Fritzson, and R. McEntire. 1994. KQML: An Information and Knowledge Exchange Protocol. In *Knowledge Building and Knowledge Sharing.* K. Fuchi and T. Yokoi, Eds. Ohmsha and IOS Press. URL: *http://www.cs.umbc.edu/kqml/papers/kbks.ps* See also: *http://www.cs.umbc.edu/kqml/*

Genesereth, M. R., and R. E. Fikes, Eds. 1992. *Knowledge Interchange Format, Version 3.0 Reference Manual.* Technical Report Logic-92-1. Computer Science Department, Stanford University. URL: *http://www-ksl.stanford.edu/knowledge-sharing/papers/kif.ps* See also: *http://www-ksl.stanford.edu/knowledge-sharing/kif/*

Berners-Lee, T. and D. Connolly. 1995. *Hypertext Markup Language - 2.0.* IETF HTML Working Group. URL: *http://www.cs.tu-berlin.de/~jutta/ht/draft-ietf-html-spec-01.html*

Ragget, D. 1995. *HyperText Markup Language Specification Version 3.0.* W3C (World-Wide Web Consortium). URL: *http://www.w3.org/pub/WWW/MarkUp/html3/CoverPage.html*