Reading Between the Lines: Using SHOE to Discover Implicit Knowledge from the Web

Jeff Heflin, James Hendler, and Sean Luke

Department of Computer Science
University of Maryland
College Park, MD 20742
{heflin, hendler, seanl}@cs.umd.edu

Abstract

This paper describes how SHOE, a set of Simple HTML Ontological Extensions, can be used to discover implicit knowledge from the World-Wide Web (WWW). SHOE allows authors to annotate their pages with ontology-based knowledge about page contents. In previous papers, we discussed how the semantic knowledge provided by SHOE allows users to issue queries that are much more sophisticated than keyword search techniques, including queries that require retrieval of information from many sources. Here, we expand upon this idea by describing how SHOE's ontologies allow agents to understand more than what is explicitly stated in Web pages through the use of context, inheritance and inference. We use examples to illustrate the usefulness of these features to Web agents and query engines.

1. Introduction

The Web is a growing, uncontrolled, incomplete, and distributed information resource. Users are overwhelmed by information overload and need ways to efficiently access the information that is most valuable to them. Current retrieval tools are inadequate because Web content focuses on presentation for human consumption. Some researchers are concentrating on Natural Language Processing (NLP) techniques to allow machines to understand HTML body text. However, these techniques do not work well in general domains and will never be able to process the growing number of Web pages that consist mostly of images. With SHOE, we propose to solve these problems by allowing Web authors to annotate their pages with machine-readable knowledge that can then be used by agents and query engines.

SHOE is a set of Simple HTML Ontological Extensions. Like HTML, it is an application of the Standard Generalized Markup Language (SGML) (ISO 1986). As such, it has its own Document Type Definition (DTD) which specifies valid tags and allowable combinations of those tags. SHOE's tags are used to specify ontology-based knowledge. There are two types of SHOE enabled pages: those that define ontologies and those that declare instances. Ontologies define the valid elements that may be used to describe instances. These include:

- Categories that define an "is a" classification scheme
- Relations that may exist between instances or that can describe properties of an instance
- Renaming rules for objects borrowed from other ontologies
- Inference rules
- Constants
- Arbitrary data types

SHOE ontologies are made available to document authors and SHOE agents by placing them on the Web. Each typically extends the base SHOE ontology and may extend other pre-existing ontologies. Anyone can create an ontology, but for an ontology to be really useful it should be common to a community that wishes to communicate or share data with each other.

Instances are distinct entities that can be classified in categories and can have relationships to other instances. Every instance must state which ontologies it is using to make claims. A simple SHOE annotated page that describes an instance is given below:

```
<HTML>
<HEAD>
<TITLE>Jane Smith's Homepage</TITLE>
<META HTTP-EQUIV="SHOE"
      CONTENT="VERSION=1.0">
</HEAD>
<BODY>
…
<INSTANCE
   KEY="http://www.cs.umd.edu/users/jsmith/">
<USE-ONTOLOGY
   ID="cs-dept-ontology" VERSION="1.0"
   PREFIX="cs"
   URL="http://www.cs.umd.edu/ont/cs.html">
<RELATION NAME="cs.name">
   <ARG POS="TO" VALUE="Jane Smith">
</RELATION>
<CATEGORY NAME="cs.gradStudent">
<RELATION NAME="cs.advisor">
   <ARG POS="TO"
    VALUE="http://www.cs.umd.edu/users/jdoe/">
</RELATION>
</INSTANCE>
</BODY>
</HTML>
```

In this example, we have declared an entity identified by "http://www.cs.umd.edu/users/jsmith/". We use a URL here only for convenience in locating the instance on the

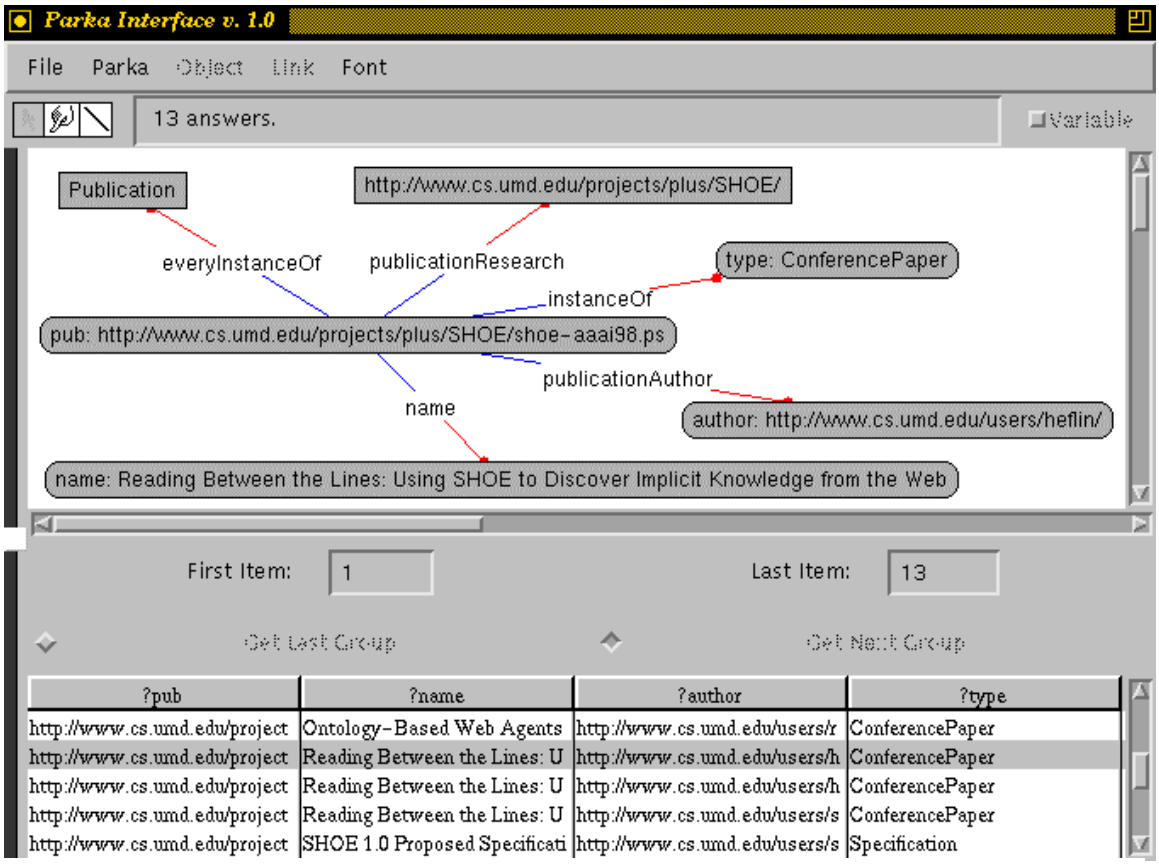

Figure 1. A Parka Query on the Knowledge Base Created by Exposé

Web; SHOE instances are not necessarily tied to specific physical HTML documents. This instance uses the "cs-dept-ontology" to declare categories and relationships. The name of the instance is "Jane Smith". Jane Smith is a graduate student and has an advisor whose instance is identified by "http://www.cs.umd.edu/users/jdoe/". Note that every category and relationship has a prefix that indicates it comes from this ontology. If we used more ontologies, then there would be additional categories and relationships with different prefixes. For a detailed description of SHOE's syntax see the SHOE Specification (Luke and Heflin 1997).

We have designed applications to demonstrate SHOE's promise. For example, we have developed Exposé, a web-crawler that looks for SHOE pages and loads them into a Parka knowledge base (Evett, Andersen, and Hendler 1993; Stoffel, Taylor, and Hendler 1997). The content can then be examined using Parka's query facilities. A Parka query concerning the authors of SHOE publications is shown in Figure 1. The Parka query tool tailored for use with SHOE includes a feature to view Web pages relevant to the query results with a Web browser. We have also built the Knowledge Annotator, a graphical application that makes it easy to annotate Web pages with SHOE. This tool prompts the user for information about instances, relations and categories, and automatically adds the appropriate SHOE tags to an existing HTML document. See Figure 2 for a screenshot from this tool. To support these and future applications, we have implemented C and Java libraries for parsing SHOE documents and manipulating the information contained within.

2. Related Work

HTML 2.0 (Berners-Lee and Connolly 1995) 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. 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>).

To address the limitations of HTML, Dobson and Burrill (1995) have attempted to reconcile it with the Entity-Relationship (ER) database model. This is done by adding to HTML a simple set of tags that define "entities" within documents, labeling sections of the body text as "attributes" of these entities, and defining relationships from an entity to outside entities. However, this scheme does not provide a means for uncovering knowledge other than that which is explicitly stated by the author of the document.

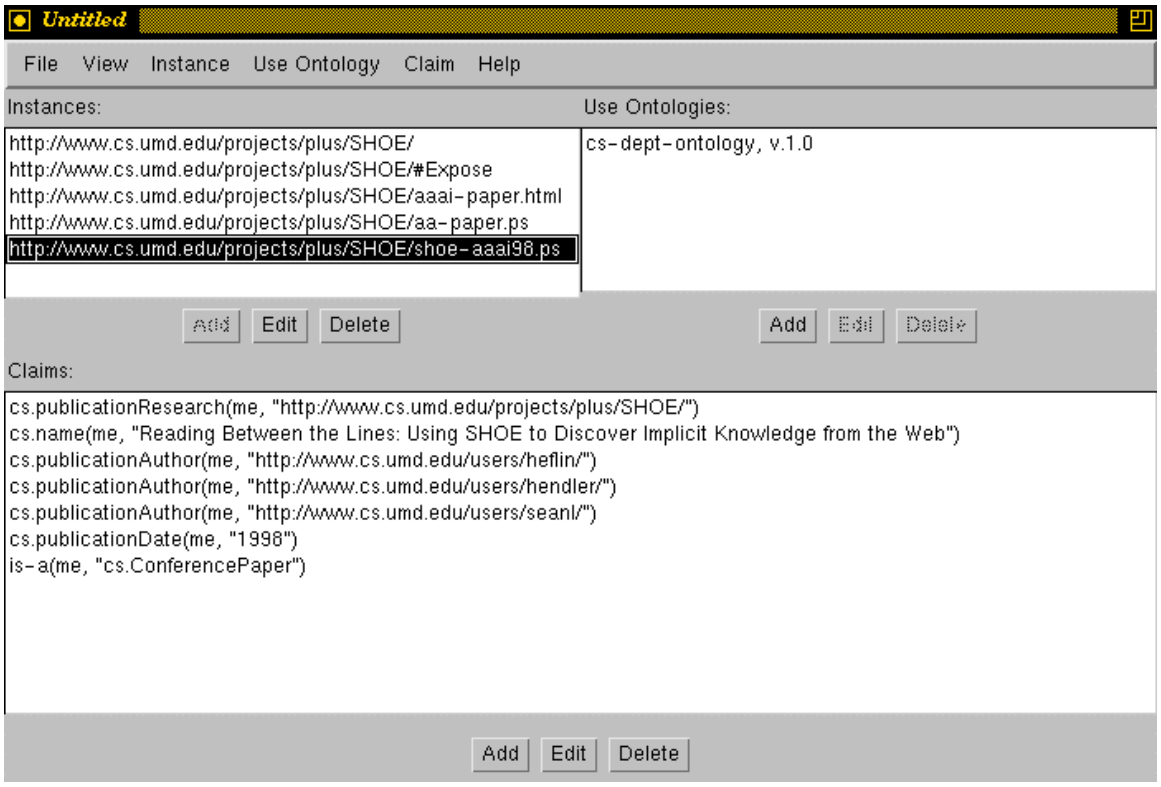

Figure 2. The Knowledge Annotator

Many research projects aim to make the Web more productive by adding database functionality. Some projects focus on creating query languages for the Web (Arocena, Mendelzon, and Mihaila 1997; Konopnicki and Shmueli 1995). However these approaches are limited to queries concerning the HTML structure of the document and the hypertext links. They also rely on index servers such as Alta Vista or Lycos to search for words or phrases, and thus suffer from the limitations of keyword search. Yet another approach involves mediators (or *wrappers*), custom software that serves as an interface between middleware and a data source (Wiederhold 1992; Papakonstantinou et al. 1995; Roth and Schwarz 1997). When applied to the Web, wrappers allow users to query a page's contents as if it was a database. However, the heterogeneity of the Web requires that a multitude of custom wrappers must be developed, and it is possible that important relationships cannot be extracted from the text based solely on the structure of the document. Semi-automatic generation of wrappers (Ashish and Knoblock 1997) is a promising approach to overcoming the first problem, but is limited to data that has a recognizable structure.

The projects most similar to SHOE are creating languages to help machines process and understand Web documents. The Resource Description Framework (RDF) (Lassila and Swick 1998) is a work in progress by the World-Wide Web Consortium (W3C). RDF is similar in concept to SHOE, but has few inferential capabilities and is limited to binary relations. The Web Analysis and Visualization Environment (WAVE) project (Kent and Neuss 1995) has designed the Ontology Markup Language (OML) and Conceptual Knowledge Markup Language (CKML), both of which are in part based on SHOE.

The Extensible Markup Language (XML) (Bray, Paoli, and Sperberg-McQueen 1998) has been proposed as a restricted form of SGML intended to ease the processing of documents on the Web. XML allows web authors to create customized sets of tags for their documents. Style sheets can be designed to display each set in a standard manner. However, the flexibility of XML will make it difficult for general intelligent agents to work with these documents since the tags developed by different authors can have the same semantics. Additionally, agents can only make use of the structure and knowledge provided explicitly on the page. SHOE was designed to use the same subset of SGML as XML, and thus can be thought of as an XML application that provides a standard way to encode useful knowledge in Web documents.

3. Discovering Implicit Knowledge

When people read documents, they draw on their knowledge of the domain and general language to interpret individual statements. Queries that can be answered easily by a human reader are difficult for most query software because the software does not have the benefit of this implicit knowledge. However, when a SHOE instance

makes specific claims based on the semantics in a particular ontology, a Web agent can then draw on that ontology to infer additional knowledge not directly stated. All of SHOE's implicit knowledge arises from its use of ontologies. Three types of implicit knowledge are provided.

First, the ontology provides context. For example, there might be a relation called "takes" in both the "university-ontology" and the "medicine-ontology". In the former, "takes" is meant as in to "take a class". In the later, it is meant as in to "take medicine". If an NLP system saw the word "takes", it would have to disambiguate between the many meanings of the word, relying on sentence structure and related words to accomplish its task. A SHOE agent, however, will not be confused because it knows what ontology is being used, and therefore it knows the context. Furthermore, SHOE requires that every category and relation have a prefix chain that identifies its original ontology. Thus, even if a page uses ontologies that have a namespace conflict, it is always clear which relationship is intended.

Second, an ontology provides a taxonomy, or categorization framework. For example, the "cs-dept-ontology" might state that a "ResearchAssistant" is a "GraduateStudent" which is a "Student" which is a "Person". A SHOE ontology would encode this information as follows:

```
<DEF-CATEGORY NAME="Person">
<DEF-CATEGORY NAME="Student" ISA="Person">
<DEF-CATEGORY NAME="GraduateStudent"
              ISA="Student">
<DEF-CATEGORY NAME="ResearchAssistant"
              ISA="GraduateStudent">
```

In the real world, and similarly on the Web, an object could belong to many categories. Therefore, SHOE allows multiple inheritance.

The categorization hierarchy can be used to generalize or specialize a query. For example, software implementing a query to find students can also return pages that had been declared "ResearchAssistants" or "GraduateStudents". Thus, users can specify queries at the hierarchical level that meets their needs; the query engine will return the right answer.

Third, a SHOE ontology allows inferences to be defined. This gives agents the tools to perform logical reasoning on Web pages. A SHOE inference is similar to a Horn clause in that it consists of a body (INF-IF) of one or more subclauses describing claims that entities might make, and a head (INF-THEN) consisting of one or more subclauses describing a claim that may be inferred if all claims in the body are made. Like Horn clauses, SHOE does not allow negations in its implications[1]. This restriction reduces computational complexity because polynomial-time inference on a set of Horn clauses can be accomplished using Modus Ponens. Furthermore, if SHOE allowed negations it would be possible for the conclusion of an inference to contradict a known fact. This is obviously problematic because reasoning with an inconsistent set of facts can lead to any conclusion desired.

The three types of SHOE subclauses are category, relation and comparison. The arguments of any subclause may be a constant or a variable. Variables are indicated by the keyword VAR. Constants must be matched exactly; variables of the same name must bind to the same value. The following example illustrates the construction of an inference that uses all three types of subclauses.

```
<DEF-INFERENCE>
<INF-IF>
   <CATEGORY NAME="Car" VAR FOR="X">
   <RELATION NAME="age">
      <ARG POS=1 VAR VALUE="X">
      <ARG POS=2 VAR VALUE="A">
   </RELATION>
   <COMPARISON OP="greaterThan">
      <ARG POS=1 VAR VALUE="A">
      <ARG POS=2 VALUE="25">
   </COMPARISON>
</INF-IF>
<INF-THEN>
   <CATEGORY NAME="Antique" VAR FOR="X">
</INF-THEN>
</DEF-INFERENCE>
```

Any SHOE inference can be expressed in first-order logic. Relation clauses simply become predicate expressions. Category clauses require unary predicates that establish membership. Comparison clauses require a binary predicate for each of the types of SHOE comparisons. Likewise, any sentence in first-order logic that can be expressed as a Horn clause can be expressed as a SHOE inference. The first-order logic equivalent of the above inference is:

$$\forall x,a\ \mathrm{Car}(x) \wedge \mathrm{Age}(x,a) \wedge \mathrm{greaterThan}(a,25) \Rightarrow \mathrm{Antique}(x)$$

The reader may have observed that SHOE notation is more cumbersome than first-order logic.[2] However, since SHOE is an SGML application, it has the advantage that it can be parsed quickly by a simple SGML parser. Additionally, XML tools not specifically designed for SHOE can process the tags to a limited extent. Still, since first-order logic is a more compact notation, all of the subsequent examples in this paper will be expressed using first-order logic.

Obviously, there are many types of rules that can be expressed using SHOE's inference mechanism. Below are some example inferences from a hypothetical university ontology:

- $\forall x\ \mathrm{Professor}(x) \Rightarrow \mathrm{hasDegree}(x, \text{"PhD"})$

[1] In conjunctive normal form a Horn clause can have no more than one positive literal.

[2] We are currently extending the Knowledge Annotator so that users can define inferences with a minimum of effort.

- ∀x,y Colleague(x,y) ⇒ Colleague(y,x)
- ∀x,y Teaches(x,y) ⇒ StudentOf(y,x)
- ∀x,y,z WorksFor(x,y) ∧ SubOrg(y,z) ⇒ WorksFor(x,z)

The first example shows that having a Ph.D. is a necessary condition of being a professor. This allows queries searching for people with doctoral degrees to also return pages that only happen to mention that the instance is a professor. The second example shows that the `Colleague` relation is symmetric. This allows agents to ignore the ordering of the arguments for that relation. The third example establishes that `Teaches` and `StudentOf` are inverse relationships. With this inference, the point of view of the Web author does not have to match that of the agent or query. Finally, the last sentence expresses the transitivity of the `WorksFor` relation through the `SubOrg` relation. In short, it says that if someone works for an organization and that organization is part of another organization, then the person also works for the second organization. This allows someone who works indirectly for an organization to be found by traversing the organization hierarchy, which may be expressed on other Web pages. As can be seen here, useful queries can be made against the predicates in the head clauses even if no page has made any declarations using those predicates. Similar types of inferences can be constructed for any domain.

To fully understand the power that SHOE provides, we will examine a more complex example than the ones above. Assume that you are looking for a list of experts on "Knowledge Representation". A simple keyword search will get you a lot of hits, many of them not what you were looking for. For example, there will be a lot of pages on conferences or indices concerning knowledge representation. Additionally, if someone mentions that they read a book or took a class on the subject, then they are as likely to be returned by a simple search engine as someone who taught a class or wrote a book on the subject. You could try adding the words "teach" or "wrote" to your query and this will improve your results, but you will still get false hits that include people who taught classes or wrote books on other topics. SHOE has none of these problems. Let's assume that the "computer-science" ontology has defined an expert on a subject as someone who has taught a class in or authored a publication on that subject. The following rules would express this:

- ∀p,c,x Teach(p,c) ∧ Subject(c,x) ⇒ Expert(p,x)
- ∀p,b,x AuthorOf(p,b) ∧ Subject(b,x) ⇒ Expert(p,x)

A SHOE query engine could use these rules to find people that match this definition of expert. Note that when we query SHOE documents, we must specify the context of the query by specifying the ontologies that we are using. Thus, other ontologies may define expert differently but the query will use the definition from the current context. This particular query may also use SHOE's categorization feature. For example, if the ontology defines "Semantic Network" as a sub-category of "Knowledge Representation", then the SHOE query engine will also be able to return people who are experts on a specific type of knowledge representation.

One of the most interesting potential Web applications for inference is in determining the validity of information. Since the Web is uncontrolled, information that is either subjective or factually incorrect is often published. Intelligent agents must not believe everything they read. Instead, they must treat knowledge found on the Web as a claim made by some party. A truly intelligent agent should be able to rank knowledge not only by how well it matches the criteria to perform an action, but also by how accurate the data is believed to be. This could be accomplished with SHOE by using inference and two special relations. The relation `Claims(x, p, y, z)` states that x claims y is related to z through predicate p. The relation `Believe(p, x, y, b)` states that we believe x is related to y through predicate p if b is true (where b is a boolean value). For example, we would be more willing to believe that someone is married to Madonna if she makes the same claim on her page. The inference rule might look like this:

```
∀x,y Claims(x, "MarriedTo", x, y) ∧
Claims(y, "MarriedTo", x, y) ⇒
Believe("MarriedTo", x, y, true)
```

Of course, this inference rule would work for any claims of marriage, not just marriage to Madonna. Alternatively, we might consider certain sources so unreliable that we never believe anything they say. For example:

```
∀p,x,y Claim("National Inquirer", p, x, y) ⇒
Believe(p, x, y, false)
```

It is important to note here that the above inference does not mean that anything claimed by the National Inquirer is necessarily false, it simply means that agents should ignore any claims by this party. It is likely that different users are willing to believe different things. Coupled with inference, the two relations above allow complex belief systems to be constructed and tailored to a specific user or user group.

4. Future Work

Although we feel our current specification provides much of the expressiveness needed for more advanced WWW agents, it still lacks important features found in sophisticated knowledge-representation systems. We are adding such features conservatively, seeking a compromise that provides much of the power of knowledge representation tools while keeping the system simple,

efficient and understandable to the lay community. We also plan to keep SHOE in step with all applicable standards. We are watching developments in XML and plan to keep SHOE compatible with it.

The main thrust of current research is building tools that demonstrate and test the capabilities of the language. In addition to the tools that have already been built, we are designing a number of agents, including:

- personal assistants that continuously scout the Web for information of interest to their masters
- on-line guides that help users browse by reading ahead of the user and suggesting the best links to follow
- on-line search engines that search the web in direct response to queries, intelligently using SHOE both to determine the validity and usefulness of information, as well as where to search next
- dynamic visualization tools that graphically display important relationships between objects and allow users to browse the Web at varying levels of abstraction

To implement these systems we must re-examine literature on databases as well as artificial intelligence, and apply the concepts to the unique situations of knowledge representation on the Web. Of particular interest are deductive databases and query planning in a distributed environment.

We are currently working with the Joint Institute for Food Safety and Applied Nutrition (JIFSAN) to fully annotate a comprehensive web site on Transmissible Spongiform Encephalopathy (TSE) Risk. JIFSAN, a cooperative agreement between the Food and Drug Administration (FDA) and the University of Maryland, is part of the President's Food Safety Initiative. This project allows us to apply SHOE to a large, real-world domain. The developmental ontology already includes elements from biology, medicine, and manufacturing. This work will help us to determine if SHOE has the correct level of expressiveness for Web documents.

Finally, we are investigating the possibilities of claim validation. We believe this is an important issue to the success of intelligent agents acting on the Web. However, the idea suggested in this paper produces its own set of problems. How do we determine what are the "ground terms" in our beliefs? Some things must be believed without a corresponding Believe() relation. Otherwise, we would never believe anything. How does one subscribe to a belief system and what complications will arise from subscribing to more than one belief system? Would establishing a degree of belief be significantly more useful than the simple boolean value suggested in this paper, and if so, would it be worth the additional complexity?

5. 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, Web users are finding it increasingly more difficult to gather information relevant to their interests without considerable and often fruitless searching. Since Web page authors design their documents with human readers in mind, they often assume a certain amount of common sense and a base level of knowledge about the subject. Truly intelligent agents must have access to knowledge of this sort. With SHOE ontologies, a user community can build a common set of categorizations and rules that provide agents with a context and initial understanding of their document content. Additionally, this packaging of information reduces the potential search space of algorithms that perform inheritance or inference. The value of each piece of information is increased because it can be used alone or in conjunction with other information to create additional knowledge. Without the taxonomies and inferences provided by SHOE ontologies, either Web page authors will need to explicitly state every fact that they deem important, or those querying the Web must think of all the possible alternate forms of their query. Certainly, neither of these approaches is very appealing.

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. Perhaps the biggest obstacle facing SHOE is getting Web pages annotated. We believe that the primary goal of web authors is to get their information to people who are interested in reading it. As SHOE demonstrates an increase in query efficiency and as we develop tools that make it even easier to annotate pages, these authors will begin to use SHOE. Additionally, many pages are being generated by databases, and the programs that do this could easily be modified to generate SHOE tags as well. If used widely, SHOE could greatly expand the speed and usefulness of intelligent agents by removing the single most significant barrier to their effectiveness: a need to comprehend text and graphical presentation as people do.

Acknowledgments

This work is supported in part by grants from ONR (N00014-J-91-1451), ARPA (N00014-94-1090, DABT-95-C0037, F30602-93-C-0039) and the ARL (DAAH049610297).

References

Arocena, G., A. Mendelzon and G. Mihaila. 1997. Applications of a Web Query Language. In *Proceedings of ACM PODS Conference*. Tuscon, Arizona.

Ashish, N. and C.A. Knoblock. 1997. Semi-automatic Wrapper Generation for Internet Information Sources. In *Proceedings of the Second IFCIS Conference on Cooperative Information Systems (CoopIS)*. Charleston, South Carolina.

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

Bray, T., J. Paoli and C.M. Sperberg-McQueen. 1998. Extensible Markup Language (XML). *W3C (World-Wide Web Consortium).* (At http://www.w3.org/TR/1998/REC-xml-19980210.html)

Dobson, S.A. and V.A. Burrill. 1995. Lightweight Databases. In *Proceedings of the Third International World-Wide Web Conference (special issue of Computer and ISDN Systems)*. v. 27-6. Amsterdam: Elsevier Science. (At http://www.igd.fhg.de/www95/papers/54/darm.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. (At http://www.cs.umd.edu/projects/plus/Parka/parka-kanal.ps)

ISO (International Organization for Standardization). 1986. *ISO 8879:1986(E). Information processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML)*. First edition -- 1986-10-15. [Geneva]: International Organization for Standardization.

Kent, R.E. and C. Neuss. 1995. Creating a Web Analysis and Visualization Environment. *Computer Networks and ISDN Systems*, 28.

Konopnicki, D. and O. Shemueli. 1995. W3QS: A Query System for the World Wide Web. In *Proceedings of the 21st International Conference on Very Large Databases*. Zurich, Switzerland.

Lassila, O. and R.R. Swick. 1998. Resource Description Framework (RDF) Model and Syntax. *W3C (World-Wide Web Consortium).* (At http://www.w3.org/TR/WD-rdf-syntax-19980216.html.)

Luke, S. and J. Heflin. 1997. *SHOE 1.0, Proposed Specification*. At http://www.cs.umd.edu/projects/plus/SHOE/spec.html

Papakonstantinou, Y., et al. 1995. A Query Translation Scheme for Rapid Implementation of Wrappers. *Procedeedings of the Conference on Deductive and Object-Oriented Databases(DOOD).* Singapore.

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

Roth, M.T., and P. Schwarz. 1997. Don't Scrap It, Wrap It! A Wrapper Architecture for Legacy Data Sources. In *Proceedings of 23rd International Conference on Very Large Data Bases*.

Stoffel, K., M. Taylor and J. Hendler. 1997. Efficient Management of Very Large Ontologies. In *Proceedings of American Association for Artificial Intelligence Conference (AAAI-97).* AAAI/MIT Press.

Wiederhold, G. 1992. Mediators in the Architecture of Future Information Systems. *IEEE Computer*, 25(3).