Integration of Semantic Web and Knowledge Discovery for Enhanced Information Retrieval

S.Kalarani,
Ph.d Research scholar, Anna university,Chennai - 25.

G.V.Uma
Asst. Prof/CSE, Anna university, Chennai -25.

ABSTRACT

Knowledge management is a process which comprises knowledge discovery, knowledge collection, knowledge organization and knowledge process. Among these four process knowledge discovery is integrated with semantic web for enhanced information retrivel. Knowledge discovery is the process of automatically searching large volume of data for patterns that can be considered knowledge about the data. This is described as deriving knowledge from the input data. Knowledge discovery is defined as "the non-trivial extraction of implicit, unknown, and potentially useful information from the data". Knowledge discovery is one of the key component of performing knowledge discovery of existing data. Although search engine technology has improved in recent years, there are still many types of searches that return unsatisfactory results. This situation can be greatly improved if web pages use a semantic markup language to describe their content, this paper describe 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. This paper contains an 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. Identifying patterns as the process of knowledge discovery an University ontology is taken as case study.

1. INTRODUCTION

This paper concentrates only the input in the form of data base and web documents. Similarly the output formats are ontology and resource description framework (RDF). The effective knowledge discovery can convert what we already know into useful knowledge base and continually update the knowledge base such that it remains current, accessible, and usable. According to semantic web we need to do annotations while storing data. These annotations are done by SHOE. It is difficult for the users to locate relevant information with the standard web search engines because these tools match on individual words instead of their meanings. As such, they cannot take the relationship between words into account, map between the terminologies of different communities, or use any contextual information to differentiate between terms with many meanings.

In this paper we discuss the overview of SHOE in section 2 and explain how it is used for annotation, and in section 3 we propose a model which is used to discover knowledge from annotated web pages. In section 4 we propose the related works and finally in section 5 we conclude our problem.

knowledge management system. Today's World Wide Web has large volume of data – billions of document. So it is a time consuming process to discover effective knowledge from the input data. Here define knowledge discovery meta model (KDM) which defines an ontology for the software and their relationships for the purpose of

Keywords: Knowledge discovery, Knowledge discovery meta model, SHOE.

According to semantic web the following are considered as input data for knowledge discovery.

- Data bases
- Software Mining
- Text
- Graphs
- web

Output formats for discovered knowledge

- Data model
- Metadata
- Metamodels
- Ontology
- Knowledge representation
- Knowledge discovery Metamodel

2. SHOE Overview

The underlying philosophy of SHOE is that intelligent agents will be able to better perform tasks on the internet if the most useful information on web pages is provided in a structured manner. To this end, SHOE extends HTML with a set of knowledge oriented tags, unlike HTML tags, SHOE provide structure for knowledge acquisition as opposed to information presentation. In addition to providing explicit knowledge, SHOE sanctions the discovery of implicit knowledge through the use of taxonomies and inference rules available in reusable ontologies that are referenced by SHOE web pages. This allows information providers to encode only the necessary information on their web pages, and to use the level of detail that is appropriate to the context. Interoperability is promoted through the sharing and reuse of ontologies.

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.

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 HTML

- `<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>`

simple case study using shoe

Find the web pages of the prof James and Peter, in which James is a faculty in an university and Peter is a student and he is guided by James.

```
Find web pages for all  x, y, and  z
such that
        x is a person,
        y is a person,
        z is an university
        where
firstName(x,"James")and
firstName( y,"Peter") and
facultyOf( z, x)and      studentOf( z,
y)and       guideTo( x, y)and
involvedIn( z,"Ph.D research”)
```

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 ``peter" is the graduate student's first 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 ``first Name", classifications like ``Person", and relationships like ``student". Systems that gather claims about these attributes and relationships could use the resulting gathered knowledge to provide answers to sophisticated knowledge-based queries.

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. We can then use this knowledge to answer sophisticated queries about these documents and their relationships.

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

- Provides a ``Person" classification
- Provides an ``university" classification
- Provides the ``guideTo" relationship between people
- Provides the ``firstName" and ``lastName" attributes for people
- Provides the ``faculty" relationship between university 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 will offer a unified structure for sharing knowledge on the World-Wide Web.

Let's assume there already exists an ontology called university-ontology version 2.1 which defines the classifications university and Thing, We'll *extend* the university-ontology ontology to include our other needed classifications and relationships. Namely, we'll borrow university 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 "universioty-
ontology" VERSION="2.1" PREFIX="uni">
<ONTDEF CATEGORY="Person"
ISA="uni.Thing">
<ONTDEF RELATION="lastName"
ARGS="Person STRING">
<ONTDEF RELATION="firstName"
ARGS="Person STRING">
<ONTDEF RELATION="guideTo"
ARGS="Person Person">
<ONTDEF RELATION="faculty"
ARGS="uni.university Person">
</ONTOLOGY>
```

This indicates that faculty is a Person, and persion is a subcategory of Thing as defined in the university-ontology ontology, that people have first and last names which are strings, that faculty can be guide to other people, and that people can be student of university.Furthermore, the place James work for, the University in Computer Science Department, has its home page.in URLlike : *http://www.cs.org.edu.* This declares James's web page to be a data entity with a unique key, and indicates that it will use the ontology university-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 James's home page, namely James's name, that James is a person, that he is guided to Peter, and that he works for the University of some X in Computer Science Department:

```
<CATEGORY "our.Person">
<RELATION "our.firstName" TO="James">
<RELATION "our.guideTo"
TO="http://www.cs.org.edu/~peter">
<RELATION "our.faculty"
FROM="http://www.cs.org.edu">
```

The category declaration indicates that James is a Person. The next relation declares that James is guide to Peter. The last relation declares the relationship faculty *from* university.

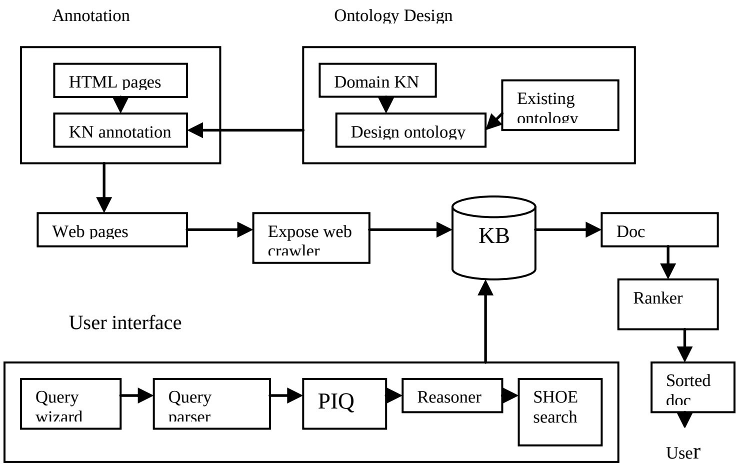
3. BUILDING THE SYSTEM

This section describes the technical aspects of the application. We also explain our design. The system architecture can be summarized as follows:

Website use a tool called the Knowledge Annotator to mark-up their pages with SHOE

The knowledge providers then place the pages on the Web. Exposé, the SHOE web crawler, is allowed to visit. Exposé crawls along the selected sites, searching for more SHOE annotated pages.
SHOE knowledge discovered by Exposé is loaded into a Parka knowledge base.

The following subsections describe how we created our ontology, how SHOE tags were added to web pages, how new SHOE information is discovered, and how users access information that is relevant to them. When SHOE pages are annotated and placed on the Web, they can be queried and indexed. collect the knowledge from the pages and store it in a repository. For this purpose, we have eveloped Exposé, a web-crawler that searches for web pages with SHOE mark-up and interns the knowledge. When Exposé discovers a new URL, it assigns it a cost and uses this cost to determine where it will be placed in a queue of URLs to be visited. When Exposé loads a web page, it parses it, and if the web page references an ontology that Exposé is unfamiliar with, it loads the ontology as well. In order to update its list of pages to visit, it identifies all of the hypertext links, category instances, and relation arguments within the page, and evaluates each new URL as above. Finally, the agent stores SHOE category and relation claims, as well as any new ontology information, in a knowledge base (KB). Currently, we store SHOE knowledge in a Parka KB. Parka has been shown to answer queries on KBs with millions of assertions in seconds, and when used on parallel machines, it provides even better performance.



3.1 ONTOLOGY DESIGN

The fundamental component of SHOE is the ontology. In SHOE, an ontology can extend one or more existing ontologies by adding its own category hierarchies, relations, and inference rules. In many cases, rules that identify the symmetric, inverse, and transitive relationships will provide sufficient inference.

3.2 ANNOTATION

Annotation is the process of adding SHOE semantic markup to a web page. A SHOE web page describes one or more instances, each representing an entity or concept. An instance is uniquely identified by a key, which is usually formed from the URL of the web page. The description of an instance consists of ontologies that it references, categories that classify it, and relations that describe it. Since manually annotating a page can be time consuming and prone to error, we have developed the Knowledge Annotator, a tool that makes it easy to add SHOE knowledge to web pages by making selections and filling in forms. The tool has an interface that displays instances, ontologies, and claims. Users can add, edit or remove any of these objects. When creating a new object, users are prompted for the necessary information. In the case of claims, a user can choose the source ontology from a list, and then choose categories or relations from a corresponding list. The available relations will automatically filter based upon whether the instances entered can fill the argument positions.

3.3 INFORMATION GATHERING

The vastness of the Internet and bandwidth limitations make it difficult for a system to perform direct queries on it efficiently. However, if the relevant data is already stored in a knowledge base, then it is possible to respond to queries very quickly. For this reason, we

have used Exposé, a softbot that searches for web pages with SHOE markup and interns the knowledge. However, since a web-crawler can only process information so quickly, there is a tradeoff between coverage of the Web and freshness of the data: if the system revisits pages frequently, then there is less time for discovering new pages. In order to use Exposé, we had to choose a knowledge base system for storing the information. We chose Parka as our knowledge base because evaluations have shown it to be very scalable, there is an n-ary version, and parallel processing can be used to improve query execution time. Since we were not interested in performing complex inferences on the data at the time, the fact that Parka's only inference mechanism is inheritance was of no consequence.

3.4 USER INTERFACES

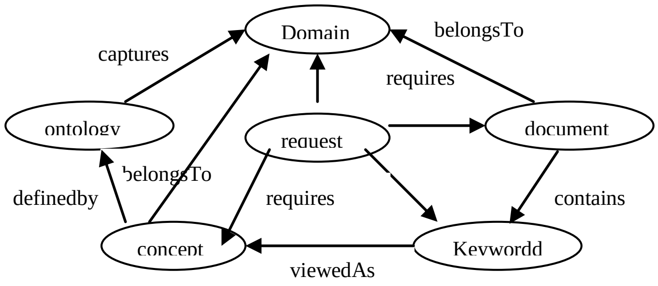
The most important aspect of the system is the ability to provide users with the information they need. User can use query wizard to create a complex query based on their need.

A query parser is used to parse the complex query which is created by query wizard into OWL-Ql query. The Java parka Interface for Queries (PIQ), graphical tool that can be used to query any Parka knowledge base. The reasoner and the SHOE search interface gives users a new way to browse the knowledge base by allowing them to submit complex queries and open documents by clicking on the URLs in the results. A user inputs a query by drawing frames and the relations between them. This specifies a conjunctive query in which the frames are either constants or variables and the relations can be a string matching function, a numerical comparison or a relation defined in an ontology. The answers to the query are displayed as a table of the possible variable bindings. If the user double-clicks on a binding that is a URL, then the corresponding web page will be opened in a new window of the user's web browser.

3.5 SHOE Search and Query Processing

The idea behind the SHOE Search tool is that if queries are issued within a context, the tool can prompt the user with context specific information and can more accurately locate the information desired by the user. SHOE Search is written in Java, and can be The user selects a context by choosing an ontology from a stored list. The list of available ontologies are those that are known by the underlying KB. The identifiers and the version numbers of each ontology are displayed, so that users may choose to issue their queries against earlier versions of ontologies.

The search process begins with the parsing of a user’s query. If a search request is in the form of keyword list, then these keywords would be treated as concepts in ontology, and documents that relates to these concepts will be retrieved based on ontology ranking. The following figure illustrates the relation between domain, ontology, concept, keyword, document and user request in a keyword based semantic search.



4. RELATED WORK

The World Wide Web is a repository of information that is structured for presentation to human readers and is thus mostly inaccessible to machines. There are numerous efforts to create semantic languages for the Web. The Ontobroker project (Fensel et al. 1998) uses a language to describe data that is embedded in HTML, but relies on a centralized broker for ontology definitions. The Ontology Markup Language (OML) and Conceptual Knowledge Markup Language (CKML) (Kent 1999) are used together for semantic markup that is based on the theories of formal concept analysis and information flow. The W3C has developed the Resource Description Framework , which uses XML to specify semantic networks for describing web resources. RDF has only a weak notion of ontologies, and although the RDF Schema proposal. This situation will be somewhat alleviated by the Extensible Markup Language (XML), which allows content to be separated from presentation. However, although XML Document Type Declarations (DTDs) can specify the grammar of markup languages, there are no facilities for formalizing the meaning of these languages. To create a web language with semantics, one must extend XML with features of knowledge representation (KR) languages.

5. CONCLUSION

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. The biggest barrier to the SHOE solution is the knowledge acquisition problem. However, adding SHOE annotations to web pages is only moderately more time consuming than converting them to standard XML. We feel that if users can be convinced of the benefits of semantic markup, then they would be more willing to take the time to do it. Nevertheless, automatic and semiautomatic solutions will be necessary to achieve a critical mass. Therefore we are examining approaches to extract SHOE from semi-structured web pages, to translate documents that use common XML DTDs to SHOE, and to translate other semantic web languages such as RDF to SHOE. Knowledge representation tools for the Web must be geared toward the average user, who often does not have the time or desire to learn first-order logic. The suite of SHOE tools, particularly SHOE Search, are a step in this direction, but there is much room for improvement.

6. REFERENCES

[1] Guarino, N. and Giaretta, P. 1995. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In N. Mars (ed.) Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing, 25-32. Amsterdam: IOS Press.

[2] Heflin, J.; Hendler, J.; and Luke, S. 1999. SHOE: A Knowledge Representation Language for Internet Applications, Technical Report, CS-TR-4078 (UMIACS TR-99-71), Dept. of Computer Science, University of Maryland

[3] Noy, N. and Hafner, C. 1997. The State of the Art in Ontology Design. AI Magazine18(3):53-74.

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

[5] Heflin, J., and Hendler, J. 2000. Dynamic Ontologies on the Web. In Proceedings of American Association for Artificial Intelligence Conference (AAAI-2000). Menlo Park, Calif.: AAAI Press.

[6] Heflin, J., Hendler, J., and Luke, S. 1999. Applying Ontology to the Web: A Case Study. In: J. Mira, J. Sanchez-Andres (Eds.), International Work-Conference on Artificial and Natural Neural Networks, IWANN'99.Proceedings, Volume II. 715-724. Berlin: Springer.

[7] Stoffel K., Taylor, M.,and Hendler, J. 1997. Efficient Management of Very Large Ontologies. In Proceedings of American Association for Artificial Intelligence

[8] ConfEvett, 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.

[9] 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/*