Grass-roots Semantic Web Tools

Baoshi Yan, Robert MacGregor,
In-Young Ko, Juan Lopez
Distributed Scalable Systems Division
Information Sciences Institute
University of Southern California

{baoshi,macgregor,iko,juan}@isi.edu

ABSTRACT

One of the biggest challenges of the Semantic Web is to make its tools usable by ordinary users for grass-roots production and integration of semantic information. This paper introduces the ongoing research on this issue in our research group at the Information Sciences Institute.

1. RESEARCH OVERVIEW

Despite years of intense work and research on the Semantic Web, it has not become a reality. One of the biggest challenges is to make Semantic Web tools usable by ordinary users. Current tools for ontology creation, annotation, ontology alignment, and querying heterogeneous data sources are still too difficult for ordinary users. In this paper we'll discuss the various ongoing efforts in our research group aiming at creating Semantic Web tools that further lower the entrance barrier to Semantic Web for ordinary users.

1.1 Grass-roots Annotator

Metadata is the basis of the Semantic Web. There has been great effort on making metadata creation [1][6] easier for ordinary users. All these tools follow the same pattern: users are required to create an ontologies first, and then make annotations according to the created ontologies. However, ontology creation is an abstract activity, which is often difficult and unintuitive for ordinary users. As a result these tools are still difficult for ordinary users to use.

We are experimenting an extreme approach. Our Grass-roots Annotator (Figure 1) would allow users to create metadata first without creating any ontology. Users would be allowed to use whatever structures and terms they like to describe their data at hand without first defining these terms and structures. We would then try to induce ontologies from the metadata corpus.

The annotator is carefully designed so that some operations are indicative of possible ontologies. We are also developing techniques to mine the metadata corpus for patterns which indicates the existence of ontologies. Furthermore, our own experience with the tool shows that, with the metadata corpus growing, we tend to use same terminologies and structures to describe similar things in order to make it easier to manage the metadata. This indicates that it might be easier for users to generalize ontologies from the data they created than to create an ontology from scratch.
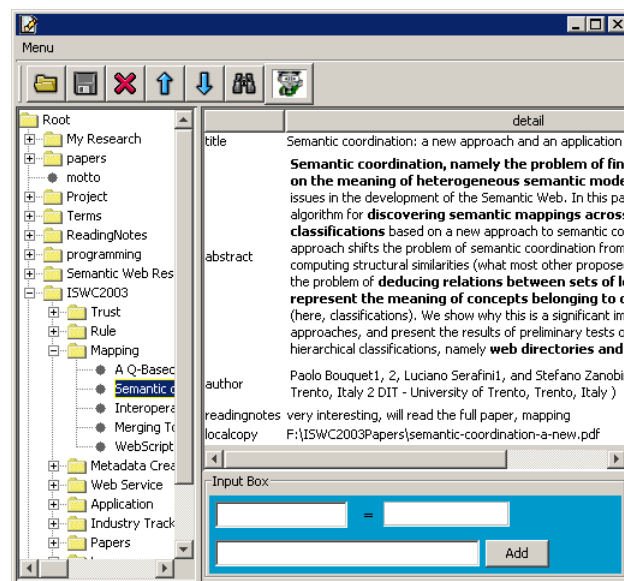
.

Figure 1: Grass-roots Annotator

1.2 WebScripter: Grass-roots Report Creation and Ontology Alignment

WebScripter[2](Figure 2) is a tool that enables ordinary users to easily and quickly assemble reports extracting and fusing information from multiple, heterogeneous Semantic Web sources. Different Semantic Web sources may use different ontologies. WebScripter addresses this problem by (a) making it easy for individual users to graphically align the attributes of two separate externally defined concepts, and (b) making it easy to reuse others' alignment work. At a high level, the WebScripter concept is that users extract content from heterogeneous sources and paste that content into what looks like an ordinary spreadsheet. What users implicitly do in WebScripter (without expending extra effort) is to build up an articulation ontology containing equivalency statements. We believe that in the long run, this articulation ontology will be more valuable than the data the users obtained when they constructed the original report. The equivalency information reduces the amount of work future WebScripter users have to perform.

The key difference we see between "traditional" ontology translation and WebScripter is that non-experts perform all of the translation - but potentially on a global scale, lever-
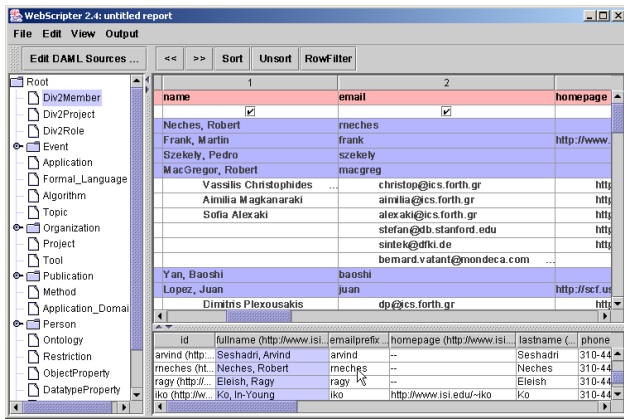
Figure 2: WebScripter Tool

Figure 3: Semantic Engineering Workbench

aging each other's work.

1.3 Naive User Queries

Traditionally, users need to write queries conforming to the schema of a data source in order to retrieve information from it. On the Semantic Web, there will be numerous data schemas. Requiring people to write different queries for different schemas is a daunting task. Thus we propose that it's necessary to deal with another type of user queries: *naive user queries*–queries in users' own terms and own semantic structures. Without losing generality, we represent a naive user query as a list of triple patterns (s,p,o) (Although syntax doesn't affect our discussion, we use RDQL-alike [5] syntax for convenience). Semantic structures between terms are binary relations: p is the kind of relationship between s and o. Such type of user queries might not conform to the schemas of available data sources.

We propose an approach [8] that, given a naive user query, translates it into a list of queries conforming to different data source schemas. The approach is based on query-rewriting techniques. It utilizes partial alignment between different schemas, alignment between different naive user queries, similarities between term names, as well as other information as query rewriting rules. An early prototype showed that the result is promising.

1.4 Semantic Engineering Workbench (SEW)

ISI's n-Dimensional Information Management project is developing an integrated suite of tools, called the Semantic Engineering Workbench (SEW)[3][7](Figure 3), that provides an intelligent infrastructure for managing Semantic Web databases and developing Semantic Web applications. The SEW has been crafted by integrating key (open-source) software components into an integral whole. Retrieval capabilities and persistence is provided by combining Hewlett-Packard's Jena triple store with a relational database (we are currently using MySQL). Ontology editing is provided by Stanford's Protege Knowledge Acquisition tool. The SEW implements several layers of API's. The highest levels provide object-oriented representations of data objects, while lower-levels enable access to triples. The SEW transparently converts triples retrieved from Jena into Protege objects, using an on-demand strategy that imports data on an as-requested basis. The SEW is wholly implemented in Java, and currently runs on Windows PCs.
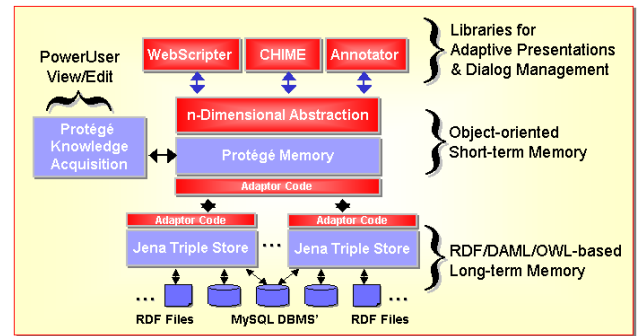
The design of the SEW was motivated by the need to provide high-level support to three Semantic Web applications: the Annotator and WebScripter tool as explained in Sections 1.1 and 1.2, and the CHIME tool [4] that allows users to view n-dimensional data.

2. CONCLUSION

We've briefly introduced several research projects in our group including Grass-roots Anntator–an extremely easy-to-use tool for metadata creation, WebScripter–a tool for grass-roots report generation and ontology alignment, Naive User Query Processing–a technique to enable queries in users' own terms and structures, and finally Semantic Engineering Workbench–the infrastructure underlying all our tools.

3. ACKNOWLEDGMENTS

4. ABOUT AUTHORS

Robert Macgregor is a senior project leader at University of Southern California's Information Sciences Institute. He is the project leader of the WebScripter and CHIME projects. Baoshi Yan and Juan Lopez are Ph.D. students participating in the WebScripter project. In-Young Ko is a post-doctoral research associate working on the WebScripter and CHIME projects.

5. REFERENCES

[1] http://protege.stanford.edu/.
[2] http://www.isi.edu/webscripter/.
[3] http://www.isi.edu/chime/sew.html.
[4] http://www.isi.edu/chime/.
[5] Rdql. http://www.hpl.hp.com/semweb/rdql.htm.
[6] S. Handschuh and S. Staab. Authoring and annotation of web pages in cream. In *The World Wide Web Conference*, 2002.
[7] R. Macgregor and I.-Y. Ko. Representing contextualized data using semantic web tools. In *1st International Workshop on Practical and Scalable Semantic Systems*, Octor 2003.
[8] B. Yan and R. Macgregor. Translating naive user queries on the semantic web. In *Semantic Integration Workshop*, Octor 2003.