A Tool for Organizing Web Information

Ontologies—models of concepts and their relationships—are a powerful way to organize query formulation and semantic reconciliation in large, distributed information environments. This case study involves a health care information system.

Kuhanandha Mahalingam and Michael N. Huhns
University of South Carolina

The physical and logical differences among information sources on the Internet complicate information retrieval in several ways.

- Data provided by the sources is no longer just simple text or tuples, but now includes objects and multimedia.
- Data also has varied and often arcane semantics.
- Sources have different policies, procedures, and conventions.
- Diverse platforms host the data.

Networks grow and evolve as new sources are incorporated or existing sources are revised, further complicating the picture. There are two basic approaches to working around these problems.

- *Client-server.* This approach has produced a plethora of search and query tools based mostly on keywords. Keywords are better for text than for the structured data found in most databases, but are completely unsuitable for sources that do not adhere to a uniform semantics.
- *Agents.* This approach achieves interoperation among sources, applications, and users as the agents serve as mediators, translators, and information brokers—the essence of a cooperative information systems architecture. The agents' major task is to reconcile the varied semantics of the mostly autonomous resources.

Ontology-based interoperation is especially good at dealing with inconsistent semantics in either approach. Ontologies can capture both the structure and semantics of information environments, so an ontology-based search engine can handle both simple keyword-based queries as well as complex queries on structured data.

ONTOLOGIES AND INTEROPERATION

To define an ontology, you start by defining a set of representational terms. Definitions associate the names of entities in a universe of discourse (such as classes, relations, functions, or other objects) with formal axioms that constrain the interpretation and well-formed use of these terms.[1] Ontologies capture these semantic relationships, whether they exist among keywords or among the tables and fields in a database.

The resulting ontology is a network structure that provides users with an abstract view of a domain-specific information space. Ontologies are well suited for knowledge sharing in a distributed environment, and researchers have implemented several ontology-based information systems.[2,3]

Value-mapping advantages

Ontologies have an advantage over unstructured text-based information spaces for mapping values to different units or formats, since query results do not typically contain information about the units of returned values. For example, when a query requests employee salaries, the results do not indicate whether the salaries are in dollars or pounds or both. In an information environment fraught with varying data representations, such problems abound.

Scalability advantages

Ontologies can grow and shrink as necessary based on the context in which they are used. In a different context, part of one ontology can be hidden or another made visible, generating a new view of the same information space to suit a certain audience.

JAVA ONTOLOGY EDITOR

Aside from their many advantages, ontologies have a major disadvantage: They are difficult to construct. We therefore developed the Java Ontology Editor (JOE) to help users build and browse ontologies. It also enables query formulation at several levels of abstraction, including a very abstract level comfortable for novice users. We've used this tool for a health care information system.[4]

A software tool written in Java, JOE provides a graphical user interface for creating or editing ontologies. It allows users to formulate queries by

0018-9162/97/$10.00 © 1997 IEEE

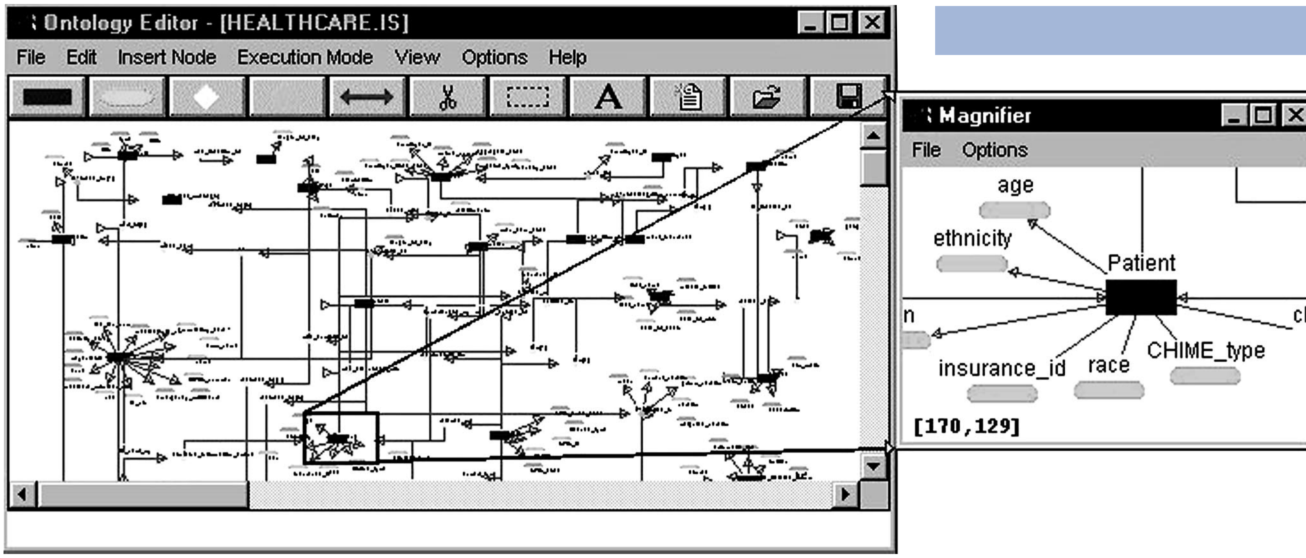

Figure 1. JOE displays entire ontology within the current window and a magnified view of the selected area on the right.

pointing and clicking on the information space it displays. We used Java for JOE because it provides advantages in distribution, security, and portability.

Query formulation advantages

To get desired information in a large heterogeneous environment, users must efficiently formulate accurate queries. This is easy if users have experience with a database query language such as SQL, or if a system allowed natural-language queries such as "Get me the phone numbers of all the Johnsons in the Columbia area." JOE provides a middle ground between natural language and formal query languages. A few mouse clicks on the ontology displayed by our tool formulates this query as

Person
  phone-number = *<?.request.?>*
  last-name = *'Johnson'*
  lives-in
    City
      name = *'Columbia'*

This format is as readable as the English command and can easily be translated to SQL. Our tool allows users with SQL knowledge to refine these queries, which users can also edit and reuse later. This type of query formulation would be difficult without a graphical representation of the ontologies.

Group editing

We implemented JOE as a Java applet. Because applets can be downloaded anywhere and anytime, more than one user can simultaneously view and edit the same ontology. This group editing feature allows people with different expertise to cooperate in creating one global ontology. At the same time, each user can create unique versions for their individual use. Alternatively, users can merge different individual ontologies to create a large superontology.

Abstraction mechanisms

Graphical ontology editors typically do not work well with large numbers of nodes or links due to the limited viewing area of most computer monitors. In addition, navigating among a large number of nodes and links can be awkward. JOE has five abstraction mechanisms to overcome such problems.

Selective viewing. JOE allows users to view an ontology with complete details or with only selected types of nodes. A user can view only entities, entities and attributes, or entities and relations, greatly reducing the complexity of a large ontology.

Searching. In editor mode, JOE provides a window with an alphabetic listing of all available nodes in the ontology. The user can locate a node by just double clicking on its name in the list, and JOE will scroll the viewing window to that node. This option minimizes the tedium of searching through a large graph.

Zooming. JOE can display an entire ontology inside the current window by zooming out. Figure 1 shows this feature for a large health care ontology.

Magnification. When the entire ontology is displayed, JOE can also magnify a small portion of the ontology. This is necessary if the ontology is large enough so that detailed information is not displayed in the zoomed-out image of the ontology. JOE displays this magnified portion in a separate window, as shown on the right of Figure 1. Clicking anywhere inside the window sets the viewing mode to normal and centers the window at that point in the ontology.

Hierarchical information. JOE can display the hierarchical information of a given ontology in a treelike structure (similar to the MS Windows file manager format). As shown in Figure 2 on the next page, JOE will display only expanded nodes. With this feature, users can selectively view or work at a comfortable level of abstraction.

In addition to these abstraction mechanisms, JOE also supports most basic editing functionality, such as selecting, cutting, and moving.

TEST APPLICATION

Currently, JOE is being integrated with InfoSleuth—an agent-based information technology architecture—and applied to a health care information system. The health care industry provides many opportunities to use tools like JOE. Although health care institutions need to share information, they cannot request information directly from each other because there are no globally accepted semantics.

The idea behind our application is to simply represent the abstract view of the information fundamental to all health care industries with a global ontology. Then users can make queries based on this ontology in a standard manner. These queries would be further refined through intermediate "translating agents" within InfoSleuth before processing by individual health care providers. All health care providers can communicate freely with each other, while continuing to maintain their individual information source architectures. This is a feasible and economical solution; it would be expensive to reengineer each facility to adhere to a new standard.

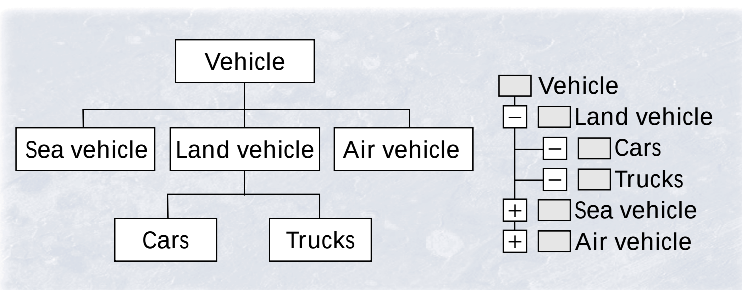

Figure 2. Vehicle ontology and its corresponding hierarchy tree.

Figure 3 shows JOE's editing mode. The graph displays part of the information space for a health care facility. It shows the patient table, all of its columns, and a few of its relations. Users can move to any node in the information space by selecting from a list of objects, attributes, and relations, shown in the left window.

When executing in query mode, JOE initially provides an option to select a "root entity" (table) from all available entities in the currently displayed ontology. After selecting the root entity, users can build a query by either

- clicking on attributes (shown as ovals), which displays a dialog box in which the user can set attribute constraints; or
- clicking on relations (diamonds) or associations (pentagons), which displays the entities related by this relation. Users can also expand these entities in turn.

A separate window to the right of the main window displays the current query. After completing the query, users can run it by choosing the submit option in the query menu. JOE displays results in a separate window. Our tool also provides an editor in which a user can directly modify or enter a new SQL statement for execution.

Although JOE's initial focus was a medical domain, there are no restrictions on its use in any similar domain. JOE can represent ontologies graphically irrespective of the domain. Given

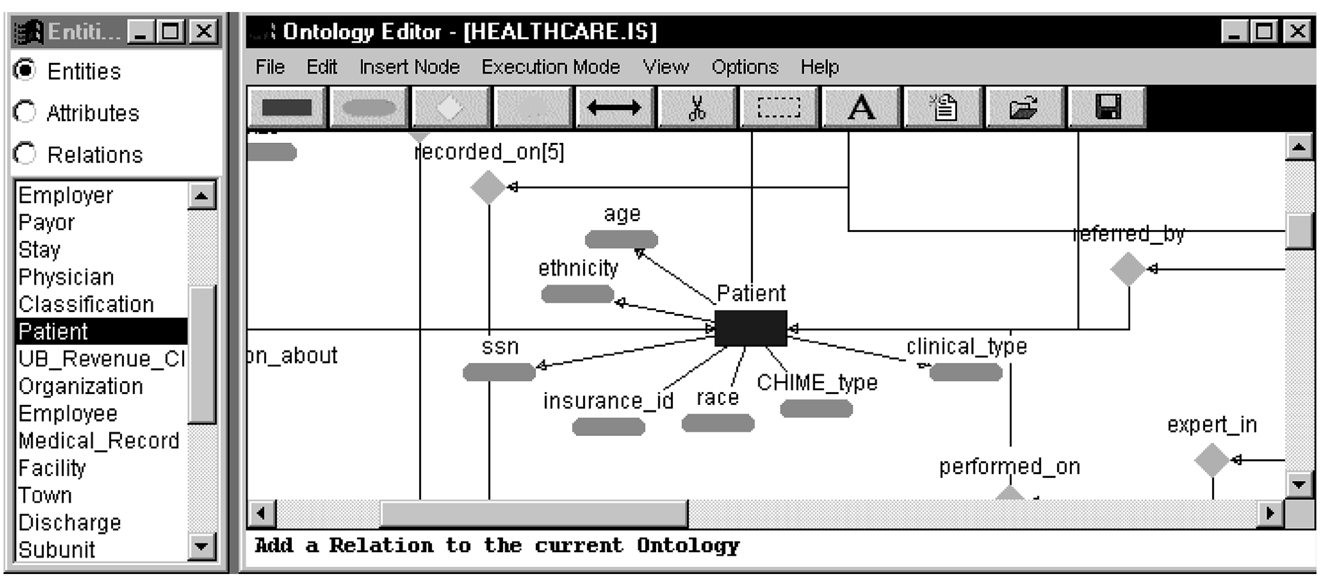

Figure 3. JOE's editing mode.

such an ontology, JOE simplifies the process of querying and returning results. In the future, JOE will allow additional editing features, such as comparing, merging, and splitting different ontologies by using semantic reconciliation and other measures. ❖

Acknowledgments

The Advanced Technology Program of the National Institute of Standards and Technology (HIIT contract number 70NANB5H1011) and the Healthcare Open Systems and Trials consortium supported this work under a cooperative agreement.

References

1. T. Gruber, "Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition, An International Journal of Knowledge Acquisition for Knowledge-Based Systems*, June 1993, pp. 199-220.
2. D. Lenat and R.V. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*, Addison-Wesley, Reading, Mass., 1990.
3. J. Hammer et al., "Information Translation, Mediation, and Mosaic-Based Browsing in the TSIMMIS System," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, ACM Press, New York, 1995, p. 483.
4. Healthcare Information Infrastructure Technology Program, http://host.scra.org/hiit.html.

Kuhanandha Mahalingam is a PhD student in the Electrical and Computer Engineering Department at the University of South Carolina. His research interests include ontology-based distributed information systems and software agent technology. Mahalingam received BS and MSEE degrees from North Carolina State University. He is a member of IEEE and the National Society for Professional Engineers.

Michael N. Huhns is a professor of electrical and computer engineering and director of the Center for Information Technology at the University of South Carolina. His research interests are multiagent systems and heterogeneous distributed databases. An associate editor for IEEE Internet Computing *and* IEEE Expert and Intelligent Systems, *he has written more than 100 technical papers. Huhns received a BSEE from the University of Michigan, Ann Arbor, and an MS and PhD in electrical engineering from the University of Southern California, Los Angeles. He is a member of the IEEE, ACM, AAAI, Sigma Xi, Tau Beta Pi, and Eta Kappa Nu.*

Contact the authors at the Center for Information Technology, Department of Electrical and Computer Engineering, University of South Carolina, Columbia, SC 29208; kuha@sc.edu.