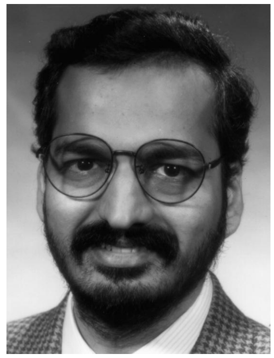

ONTOLOGIES FOR AGENTS

Michael N. Huhns • *University of South Carolina* • *huhns@sc.edu*
Munindar P. Singh • *North Carolina State University* • *singh@ncsu.edu*

When we need to find the cheapest airfare, we call our travel agent. We can communicate with our travel agent because we all speak the same language—say, English—and we all understand the basic elements of the subject under discussion—tickets, planes, destinations, departure times, fares, and so on. But suppose it is after hours and we are busy. Perhaps we'd rather our software agent contact our travel agent's software agent to arrange our flights. None of our agents understand English, so how can they communicate?

Ontologies may be the answer.

ONTOLOGIES: A DEFINITION

An ontology is a computational model of some portion of the world. It is often captured in some form of a semantic network—a graph whose nodes are concepts or individual objects and whose arcs represent relationships or associations among the concepts (see Figure 1). This network is augmented by properties and attributes, constraints, functions, and rules that govern the behavior of the concepts.

Formally, an ontology is an agreement about a shared conceptualization, which includes frameworks for modeling domain knowledge and agreements about the representation of particular domain theories. Definitions associate the names of entities in a universe of discourse (for example, classes, relations, functions, or other objects) with human-readable text describing what the names mean, and formal axioms that constrain the interpretation and well-formed use of these names.

For information systems, or for the Internet, ontologies can be used to organize keywords and database concepts by capturing the semantic relationships among the keywords or among the tables and fields in a database. The semantic relationships give users an abstract view of an information space for their domain of interest.

A SHARED VIRTUAL WORLD

How can such an ontology help our software agents? It can provide a shared virtual world in which each agent can ground its beliefs and actions. When we talk with our travel agent, we rely on the fact that we all live in the same physical world containing planes, trains, and automobiles. We know, for example, that a 777 is a type of airliner that can carry us to our destination.

When our agents talk, the only world they share is one consisting of bits and bytes—which doesn't allow for a very interesting discussion! An ontology gives the agents a richer and more useful domain of discourse.

We wrote in an earlier column (see "Conversational Agents," *IEEE Internet Computing*, Vol. 1, No. 2, pp. 73–75*) about communication protocols, such as KQML, by which agents can exchange messages. Such a protocol specifies the syntax but not the semantics of the messages. However, it also allows the agents to state which ontology they are presuming as the basis for their messages.

Suppose both our agents have access to an ontology for travel, with concepts such as airplanes and destinations, and suppose their agent tells our agent about a flight on a 777. Suppose further that the concept "777" is not part of the travel ontology. How could our agent understand? The travel agent's agent could explain that a 777 is a kind of airplane, which is a concept in the travel ontology. Our agent would then know the general characteristics of a 777. This communication is illustrated in Figure 2 (next page).

RELATIONSHIPS REPRESENTED

Most ontologies represent and support relationships among classes of meaning. Among the most important of these relationships are:

- *Generalization and inheritance*, which are powerful abstractions for sharing similarities among classes while preserving their differences. Generalization is the relationship between a class and one or more refined versions of it. Each subclass inherits the features of its superclass, adding other features of its own. Generalization

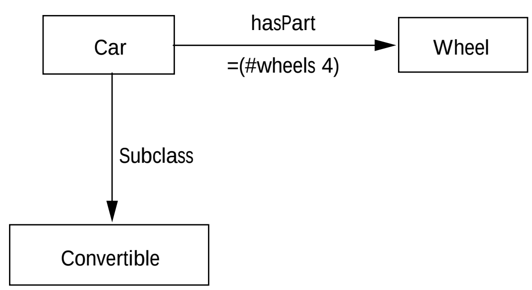

Figure 1. Structure of an ontology.

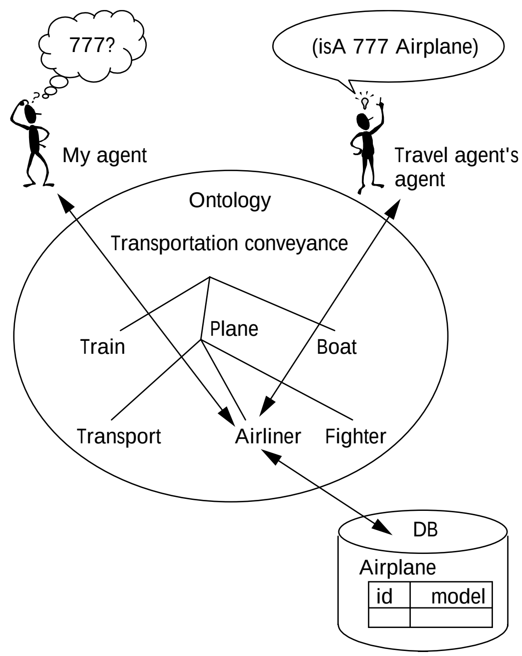

Figure 2. Communication between agents sharing a travel ontology.

and inheritance are transitive across an arbitrary number of levels. They are also antisymmetric.

- *Aggregation*, the part-whole or part-of relationship, in which classes representing the components of something are associated with the class representing the entire assembly. Aggregation is also transitive, as well as antisymmetric. Some of the properties of the assembly class propagate to the component classes.
- *Instantiation*, which is the relationship between a class and each of the individuals that constitute it.

Some of the other relationships that occur frequently in ontologies are *owns*, *causes*, and *contains*. Causes and contains are transitive and antisymmetric; owns propagates over aggregation, because when you own something, you also own all of its parts.

META CONTENT FORMAT

A recent development for the World Wide Web is Meta Content Format.* MCF is an open-format language for representing a wide range of ontological information about content. Targeted content includes Web pages, gopher and ftp files, desktop files, e-mail, and structured (that is, relational and object-oriented) databases. The corresponding metacontent includes indices such as Yahoo, gopher, and ftp directory structures, e-mail headers, data dictionaries, and so on. There are now sites on the Web that organize their information according to MCF, producing MCF information spaces (or Xspaces). There are also viewers available for your browser that let you "fly" through the Xspaces.

CLASSIFICATION SCHEMES AND STRUCTURES

Many efforts are under way to devise classification schemes and to use the schemes to build and populate classification structures. We list here four types of classification schemes of varying power that provide semantics for messages among agents. Each scheme has particular strengths and weaknesses, and provides the foundation upon which particular capabilities can be built.

- *Keywords*. Keywords are a quick way for agents to locate potentially useful information.
- *Thesauri*. Thesauri offer a more structured approach than keywords, arranging descriptive terms into broader, narrower, and related classification categories.
- *Taxonomies*. Taxonomies provide classification structures that add the power of inheritance of meaning from generalized taxa to specialized taxa.
- *Ontologies*. Ontologies permit a richer variety of structural and nonstructural relationships than taxonomies, which are limited just to generalization. Ontologies provide more complete and precise domain models as are needed, for example, by software applications that implement intelligent information services.

There are additional purposes for developing classifications for concepts. Among them are:

- helping users find a particular concept from among many;
- facilitating the administration of information systems;
- through inheritance, conveying semantic content that is often only incompletely specified by other attributes, such as names and definitions;
- deriving and formulating abstract and application concepts;
- ensuring appropriate attribute and attribute-value inheritance;
- deriving names from a controlled vocabulary;
- disambiguating communicated information;
- recognizing superordinate, coordinate, and subordinate concepts;
- recognizing relationships among concepts; and
- assisting in the development of modularly designed names and definitions.

SYSTEMS OF THE BIMONTH

To experiment with the creation of ontologies, try the Java Ontology Editor* from the University of South Carolina. JOE is a graphical user interface with two major parts: an ontology editor and a query formulation tool. The ontology editor provides a user interface for creating a new or editing an old ontology by adding new concepts (entities), attributes for the concepts, and relationships between two or more concepts. The query formulation tool is also a user interface. It allows a user to build queries on the information space displayed by the ontology editor.

An alternative way to construct an ontology is to use the editor developed as part of the Ontolingua Project at Stanford.* This site also has a number of ontologies contributed by other developers.

An example of a large ontology for a healthcare domain is the Object-

Oriented Healthcare Vocabulary Repository Project* at the New Jersey Institute of Technology. OOHVR has approximately 5,000 concepts organized in a semantic network and stored in an object-oriented database. It is accessible on the Web via any browser.

The largest and most comprehensive ontology is Cyc,* developed at MCC and Cycorp. Cyc has approximately 50,000 concepts, with more than four million constraints and relationships among the concepts. The upper level of the ontology is available at the Cycorp Web site.

URLs FOR THIS COLUMN

***Cyc Ontology Guide** • www.cyc.com/cyc-2-1/toc.html
***IEEE Internet Computing*, Vol. 1, No. 2** • www.computer.org/internet/ic1997/w2toc.htm
***JOE** • www.ece.sc.edu/Labs/HIIT/html/joe/
***MCF** • mcf.research.apple.com/
***Ontolingua Project** • www-ksl-svc.stanford.edu:5915/
***OOHVR** • object.njit.edu:2000/~newoohvr/JBI/INTERMED/InterTerms.html