

Ontologies and the Semantic Web

by Elin K. Jacob

Elin K. Jacob is associate professor, School of Library and Information Science, Indiana University-Bloomington and can be reached by e-mail at ejacob@indiana.edu

For those interested in the continuing evolution of the Web – and particularly for those actively engaged in development of the Semantic Web – ontologies are “sexy.” But even though ontologies are currently a very popular topic, there appears to be some confusion as to just what they are and the role that they will play on the Semantic Web. Ontologies have been variously construed as classification schemes, taxonomies, hierarchies, thesauri, controlled vocabularies, terminologies and even dictionaries. While they may display characteristics reminiscent of each of these systems, to equate ontologies with any one type of representational structure is to diminish both their function and their potential in the evolution of the Semantic Web.

Ontology (with an upper-case “O”) is the branch of philosophy that studies the nature of existence and the structure of reality. However, the definition provided by John Sowa (<http://users.bestweb.net/~sowa/ontology/index.htm>) is more appropriate for understanding the function of ontologies on the Semantic Web. Ontology, Sowa explains, investigates “the categories of things that exist or may exist” in a particular domain and produces a catalog that details the types of things – and the relations between those types – that are relevant for that domain. This catalog of types is an *ontology* (with a lower-case “o”).

The term *ontology* is frequently used to refer to the semantic understanding – the conceptual framework of knowledge – shared by individuals who participate in a given domain. A semantic ontology may exist as an informal conceptual structure with concept types and their relations named and defined, if at all, in natural language. Or it may be constructed as a formal semantic account of the domain with concept types and their relations systematically defined in a logical language and generally ordered by genus-species – or type-subtype – relationships. Within the environment of the Web, however, an ontology is not simply a conceptual

framework but a concrete, syntactic structure that models the semantics of a domain – the conceptual framework – in a machine-understandable language.

The most frequently quoted definition of an ontology is from Tom Gruber. In “Ontologies as a specification mechanism” (www-ksl.stanford.edu/kst/what-is-an-ontology.html), Gruber described an ontology as “an explicit specification of a conceptualization.” This definition is short and sweet but patently incomplete because it has been taken out of context. Gruber was careful to constrain his use of *conceptualization* by defining it as “an abstract, simplified view of the world that we wish to represent for some purpose” – a partial view of the world consisting only of those “objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them.” Following Gruber’s lead, an ontology can be defined as a partial, simplified conceptualization of the world as it is assumed to exist by a community of users – a conceptualization created for an explicit purpose and defined in a formal, machine-processable language.

Why Do We Need Ontologies?

Because the Web is currently structured to support humans, domain terms and HTML metadata tags that are patently transparent to human users are meaningless to computer systems, to applications and to agents. XML is gaining increasing acceptance and is rapidly replacing HTML as the language of the Web. But XML schemas deal primarily with the physical structure of Web documents; and XML tag names lack the explicit semantic modeling that would support computer interpretation. If the Semantic Web is to realize the goal of enabling systems and agents to “understand” the content of a Web resource and to integrate that understanding with the content of other resources, the system or agent must be able to interpret the semantics of each resource, not only to

Ontologies are not new to the Web. Any metadata schema is, in effect, an ontology specifying the set of physical and/or conceptual characteristics of resources that have been deemed relevant for a particular community of users.

accurately represent the content of those resources but also to draw inferences and even discover new knowledge. In the environment of the Semantic Web, then, an ontology is a partial conceptualization of a given knowledge domain, shared by a community of users, that has been defined in a formal, machine-processable language for the explicit purpose of sharing semantic information across automated systems.

An ontology offers a concise and systematic means for defining the semantics of Web resources. The ontology specifies relevant domain concepts, properties of those concepts – including, where appropriate, value ranges or explicit sets of values – and possible relationships among concepts and properties. Because an ontology defines relevant concepts – the types of things and their properties – and the semantic relationships that obtain between those concepts, it provides support for processing of resources based on meaningful interpretation of the content rather than the physical structure of a resource or syntactic features such as sequential ordering or the nesting of elements.

An Example of an Ontology

Ontologies are not new to the Web. Any metadata schema is, in effect, an ontology specifying the set of physical and/or conceptual characteristics of resources that have been deemed relevant for a particular community of users. Thus, for example, the set of elements and element refinements defined in the Dublin Core [DC] is itself an ontology. The most current version of the DC element set ([http://dublincore.org/usage/terms/dc/current elements/](http://dublincore.org/usage/terms/dc/current%20elements/)) consists of 16 attributes (element types) and 30 qualifiers (element refinements or subtypes) that are defined and maintained by the Dublin Core Metadata Initiative Usage Board. DC is intended to support consistency in the description and semantic interpretation of networked resources. To this end, declaration of the vocabulary of DC (the set of elements and element refinements) in the machine-processable language of RDF/RDFS (see below) is projected to be available in early 2003.

DC is a relatively simple representational structure applic-

able to a wide range of non-domain-specific resources. Nonetheless, it is an ontology, albeit a very general one, because it imposes a formally defined conceptual model that facilitates the automated processing necessary to support the sharing of knowledge across systems and thus the emergence of the Semantic Web. While an ontology typically defines a vocabulary of domain concepts in an is-a hierarchy that supports inheritance of defining features, properties and constraints, DC illustrates that hierarchical structure is not a defining feature of ontologies. The 16 elements currently defined by DC are independent of each other: none of the elements is required by the conceptual model and any one may be repeated as frequently as warranted for any given resource.

The Role of RDF/RDFS

Although hierarchy is not a defining characteristic of ontologies, it is an important component in the representational model prescribed by the Resource Description Framework (RDF) Model and Syntax Specification (www.w3.org/TR/REC-rdf-syntax/) and the RDF Vocabulary Description Language schema (RDFS) (www.w3.org/TR/rdf-schema). RDF and RDFS have been developed by the W3C and together comprise a general-purpose knowledge representation tool that provides a neutral method for describing a resource or defining an ontology or metadata schema. RDF/RDFS doesn't make assumptions about content; it doesn't incorporate semantics from any particular domain; and it doesn't depend on a set of predetermined values. However, it does support reuse of elements from any ontology or metadata schema that can be identified by a Uniform Resource Identifier (URI).

RDF defines a model and a set of elements for describing resources in terms of named properties and values. More importantly, however, it provides a syntax that allows any resource description community to create a domain-specific representational schema with its associated semantics. It also supports incorporation of elements from multiple metadata schemas. This model and syntax can be used for encoding information

in a machine-understandable format, for exchanging data between applications and for processing semantic information. RDFS complements and extends RDF by defining a declarative, machine-processable language – a “metaontology” or core vocabulary of elements – that can be used to formally describe an ontology or metadata schema as a set of classes (resource types) and their properties; to specify the semantics of these classes and properties; to establish relationships between classes, between properties and between classes and properties; and to specify constraints on properties. Together, RDF and RDFS provide a syntactic model and semantic structure for defining machine-processable ontologies and metadata schemas and for supporting interoperability of representational structures across heterogeneous resource communities.

RDFS

In order to understand more clearly both the nature and the function of ontologies, it is helpful to look more closely at the schema structure of RDFS. While an XML schema places specific constraints on the structure of an XML document, an RDFS schema provides the semantic information necessary for a computer system or agent to understand the statements expressed in the language of classes, properties and values

established by the schema. One of the more important mechanisms that RDFS relies on to support semantic inference and build a web of knowledge is the relationship structure that typifies the hierarchy and is so characteristic of traditional classification schemes. The creation of generic relationships through the nesting structure of genus-species (or type-subtype) capitalizes on the power of hierarchical inheritance whereby a subclass or subproperty inherits the definition, properties and constraints of its parent.

An RDFS ontology differs from taxonomies and traditional classification structures, however, in that the top two levels of the hierarchy – the superordinate class *resource* and its subordinate classes *class* and *property* – are not determined by the knowledge domain of the ontology but are prescribed by the RDFS schema. Every element in the ontology is either a type of *class* or a type of *property*. Furthermore, the relationships between classes or properties are potentially poly-hierarchical: thus, for example, a particular class may be a subclass of one, two, three or more superordinate classes.

A taxonomy or traditional classification scheme systematically organizes the knowledge of a domain by identifying the essential or defining characteristics of domain entities and creating a hierarchical ordering of mutually exclusive classes

The *BULLETIN OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE AND TECHNOLOGY* is a BIMONTHLY PUBLICATION that serves as the newsletter of the Society. It publishes short articles on a BROAD RANGE OF TOPICS of current concern to ASIST MEMBERS, focusing particularly on material of interest to practitioners. Readers are ENCOURAGED TO SUGGEST topics of interest or alert the Editor of suitable material that may have been presented at ASIST-sponsored events or elsewhere. In addition, authors are ENCOURAGED TO SUBMIT articles on topics such as CURRENT PRACTICE, PUBLIC POLICY, LEGISLATION, STANDARDS, PILOT PROJECTS, STATE-OF-THE ART REVIEWS or OVERVIEWS OF EVOLVING TECHNOLOGY AND ITS IMPACT. Articles informing the membership about various developments within ASIST are very welcome, as are articles reporting on ACTIVITIES OUTSIDE THE UNITED STATES. The *Bulletin* encourages original articles, but will consider TIMELY MATERIAL that has been presented or published elsewhere. Articles are posted in full on the ASIS Web Site at <http://www.asis.org/Bulletin/index.html>

Authors interested in developing material for a focused issue are urged to contact the Editor directly.

Authors are encouraged to discuss article ideas with the Editor if there are questions about suitability or relevance.

Irene L. Travis, Editor
Bulletin of the American Society for Information Science and Technology
1320 Fenwick Lane,
Silver Spring, MD 20910
(301) 495-0900
Bulletin@asis.org

BULLETIN
of the American Society for Information Science and Technology

to which the entities themselves are then assigned. In contrast, an RDFS ontology does not create classes into which domain resources are slotted. Rather, the ontology defines a set of elements (or slots) to which values may be assigned, as appropriate, in order to represent the physical and conceptual features of a resource. And, unlike a classification scheme, the ontology may also incorporate a set of inference rules that allows the system or agent to make inferences about the represented knowledge, to identify connections across resources or to discover new knowledge.

In an RDFS ontology, relationships between classes and properties are created by specifying the *domain* of a property, thereby constraining the class or set of classes to which a given property may be applied. In this respect, the structure of an RDFS schema is reminiscent of a faceted representational language or thesaurus. However, unlike a thesaurus, which authorizes a controlled vocabulary of terms (values) that can be assigned to represent the content of a resource, the structure of an RDFS ontology consists of a system of elements or slots whose possible range of values may or may not be established by the ontology. RDFS does provide for establishment of a controlled vocabulary (or vocabularies) within the structure of the ontology: specifying the *range* of a property stipulates that any value of that property must be an instance of a particular class of resources (e.g., the class *Literal*). An RDFS ontology is further distinguished from a

traditional thesaurus in that it does not incorporate a lead-in vocabulary. And, while it is possible to map natural language synonyms to the appropriate classes or properties in the ontology, this must be accomplished through a domain lexicon that is external to the ontology itself.

The argument that an ontology constitutes a controlled vocabulary is only valid if the standard concept of a controlled vocabulary is redefined. A controlled vocabulary is generally understood to consist of a set of terms (values) that have been authorized to represent the content of a resource. In contrast, an ontology consists of a catalog of types and properties – a catalog of controlled and well-defined element slots – that are meaningless when applied to a resource unless they are paired with an appropriate value. And, although an ontology defines a catalog of types, it is not a dictionary. A dictionary is a list of terms and associated definitions arranged in a meaningful order; but, because that order is generally alphabetical, it does not establish the meaningful relationships among terms (elements) that are characteristic of an ontology.

An ontology is not a taxonomy, a classification scheme or a dictionary. It is, in fact, a unique representational system that integrates within a single structure the characteristics of more traditional approaches such as nested hierarchies, faceted thesauri and controlled vocabularies. An ontology provides the semantic basis for metadata schemes and facilitates communication among systems and agents by enforcing a standardized conceptual model for a community of users. In so doing, ontologies provide the meaningful conceptual foundation without which the goal of the Semantic Web would be impossible.

Recommended Reading

- Guarino, N. (1998). Formal ontology and information systems. In N. Guarino (Ed.), *Formal ontology in information systems: Proceedings of FOIS '98* (pp. 3-15). Amsterdam: IOS Press. Available at www.ladseb.pd.cnr.it/infor/ontology/PUBL15.html
- Guarino, N., & Giarretta, P. (1995). Ontologies and knowledge bases: Towards a terminological clarification. In N. Mars (Ed.), *Towards very large knowledge bases: Knowledge building and knowledge sharing* (pp. 25-32). Amsterdam: IOS Press. Available at www.ladseb.pd.cnr.it/infor/Ontology/Papers/KBKS95.pdf
- Holsapple, C.W., & Joshi, K.D. (2002). A collaborative approach to ontology design. *Communications of the ACM*, 45(2), 42-47.
- Kim, H. (2002). Predicting how ontologies for the Semantic Web will evolve. *Communications of the ACM*, 45(2), 48-54.
- Noy, N. F., & McGuinness, D. L. (2001). *Ontology development 101: a guide to creating your first ontology*. Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880. Stanford Knowledge Systems Laboratory. Available at www.ksl.stanford.edu/people/dlm/papers/ontology-tutorial-noy-mcguinness-abstract.html
- Uschold, M., & Grüninger, M. (1996). Ontologies: principles, methods and applications. *Knowledge Engineering Review*, 11(2), 93-155. Available at <http://citeseer.nj.nec.com/uschold96ontology.html>

Future Directions

Much still must be done to extend the capabilities and effectiveness of current ontological models. While there is ongoing work to refine the RDF/RDFS model and schema, other efforts such as the DAML+OIL Web Ontology Language (www.w3.org/TR/daml+oil-reference) and the Web Ontology Language [OWL] (www.w3.org/TR/2002/WD-owl-ref-20021112/) seek to build on the foundation established by RDF and RDFS.

Conclusion

It is simply not true that there is nothing new under the sun. This is aptly underscored not only by the history of the Web itself but also by ongoing efforts to realize the potential of the Semantic Web. Limiting responses to these new challenges by adhering to traditional representational structures will ultimately undermine efforts to address the unique needs of these new environments. As recent developments with ontologies illustrate, the knowledge accrued across generations of practical experience must not be discarded; but there must be the conscious effort to step outside the box – to rethink traditional approaches to representation in light of the changing requirements occasioned by the constantly evolving environment of the Web.