

Ontology

by [Tom Gruber](#)

to appear in the *Encyclopedia of Database Systems*, Ling Liu and M. Tamer Özsu (Eds.), Springer-Verlag, 2008.

SYNONYMS

computational ontology, semantic data model, ontological engineering

DEFINITION

In the context of computer and information sciences, an ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. The representational primitives are typically classes (or sets), attributes (or properties), and relationships (or relations among class members).

The definitions of the representational primitives include information about their meaning and constraints on their logically consistent application. In the context of database systems, ontology can be viewed as a level of abstraction of data models, analogous to hierarchical and relational models, but intended for modeling knowledge about individuals, their attributes, and their relationships to other individuals. Ontologies are typically specified in languages that allow abstraction away from data structures and implementation strategies; in practice, the languages of ontologies are closer in expressive power to first-order logic than languages used to model databases.

For this reason, ontologies are said to be at the "semantic" level, whereas database schema are models of data at the "logical" or "physical" level. Due to their independence from lower level data models, ontologies are used for integrating heterogeneous databases, enabling interoperability among disparate systems, and specifying interfaces to independent, knowledge-based services.

In the technology stack of the Semantic Web standards [1], ontologies are called out as an explicit layer. There are now standard languages and a variety of commercial and open source tools for creating and working with ontologies.

HISTORICAL BACKGROUND

The term "ontology" comes from the field of philosophy that is concerned with the study of being or existence. In philosophy, one can talk about an ontology as a theory of the nature of existence (e.g., Aristotle's ontology offers primitive categories, such as substance and quality, which were presumed to account for All That Is). In computer and information science, ontology is a technical term denoting an artifact that is *designed* for a purpose, which is to enable the modeling of knowledge about *some* domain, real or imagined.

The term had been adopted by early Artificial Intelligence (AI) researchers, who recognized the applicability of the work from mathematical logic [6] and argued that AI researchers could create new ontologies as computational models that enable certain kinds of automated reasoning [5]. In the 1980's the AI community came to use the term ontology to refer to both a theory of a modeled world (e.g., a Naïve Physics [5]) and a component of knowledge systems.

Some researchers, drawing inspiration from philosophical ontologies, viewed computational ontology as a kind of applied philosophy [10].

In the early 1990's, an effort to create interoperability standards identified a technology stack that called out the ontology layer as a standard component of knowledge systems [8]. A widely cited web page and paper [3] associated with that effort is credited with a deliberate definition of ontology as a technical term in computer science. The paper defines ontology as an "explicit specification of a conceptualization," which is, in turn, "the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold among them." While the terms specification and conceptualization have caused much debate, the essential points of this definition of ontology are

- An ontology defines (specifies) the concepts, relationships, and other distinctions that are relevant for modeling a domain.
- The specification takes the form of the definitions of representational vocabulary (classes, relations, and so forth), which provide meanings for the vocabulary and formal constraints on its coherent use.

One objection to this definition is that it is overly broad, allowing for a range of specifications from simple glossaries to logical theories couched in predicate calculus [9]. But this holds true for data models of any complexity; for example, a relational database of a single table and column is still an instance of the relational data model. Taking a more pragmatic view, one can say that ontology is a tool and product of engineering and thereby defined by its use.

From this perspective, what matters is the use of ontologies to provide the representational machinery with which to instantiate domain models in knowledge bases, make queries to knowledge-based services, and represent the results of calling such services. For example, an API to a search service might offer no more than a textual glossary of terms with which to formulate queries, and this would act as an ontology. On the other hand, today's W3C Semantic Web standard suggests a specific formalism for encoding ontologies (OWL), in several variants that vary in expressive power [7]. This reflects the intent that an ontology is a specification of an abstract data model (the domain conceptualization) that is independent of its particular form.

SCIENTIFIC FUNDAMENTALS

Ontology is discussed here in the applied context of software and database engineering, yet it has a theoretical grounding as well. An ontology specifies a vocabulary with which to make assertions, which may be inputs or outputs of knowledge agents (such as a software program). As an *interface specification*, the ontology provides a language for communicating with the agent.

An agent supporting this interface is not required to use the terms of the ontology as an *internal encoding* of its knowledge. Nonetheless, the definitions and formal constraints of the ontology do put restrictions on what can be *meaningfully* stated in this language. In essence, committing to an ontology (e.g. supporting an interface using the ontology's vocabulary) requires that statements that are asserted on inputs and outputs be *logically consistent* with the definitions and constraints of the ontology [3]. This is analogous to the requirement that rows of a database table (or insert statements in SQL) must be consistent with integrity constraints, which are

stated declaratively and independently of internal data formats.

Similarly, while an ontology must be formulated in *some* representation language, it is intended to be a semantic level specification -- that is, it is independent of data modeling strategy or implementation. For instance, a conventional database model may represent the identity of individuals using a primary key that assigns a unique identifier to each individual. However, the primary key identifier is an artifact of the modeling process and does not denote something in the domain. Ontologies are typically formulated in languages which are closer in expressive power to logical formalisms such as the predicate calculus. This allows the ontology designer to be able to state semantic constraints without forcing a particular encoding strategy. For example, in typical ontology formalisms one would be able to say that an individual was a member of class or has some attribute value without referring to any implementation patterns such as the use of primary key identifiers. Similarly, in an ontology one might represent constraints that hold across relations in a simple declaration (A is a subclass of B), which might be encoded as a join on foreign keys in the relational model.

Ontology engineering is concerned with making representational choices that capture the relevant distinctions of a domain at the highest level of abstraction while still being as clear as possible about the meanings of terms. As in other forms of data modeling, there is knowledge and skill required. The heritage of computational ontology in philosophical ontology is a rich body of theory about how to make ontological distinctions in a systematic and coherent manner.

For example, many of the insights of "formal ontology" motivated by understanding "the real world" can be applied when building computational ontologies for worlds of data [4]. When ontologies are encoded in standard formalisms, it is also possible to reuse large, previously designed ontologies motivated by systematic accounts of human knowledge or language [11]. In this context, ontologies embody the results of academic research, and offer an operational method to put theory to practice in database systems.

KEY APPLICATIONS

Ontologies are part of the W3C standards stack for the Semantic Web, in which they are used to specify standard conceptual vocabularies in which to exchange data among systems, provide services for answering queries, publish reusable knowledge bases, and offer services to facilitate interoperability across multiple, heterogeneous systems and databases. The key role of ontologies with respect to database systems is to specify a data modeling representation at a level of abstraction above specific database designs (logical or physical), so that data can be exported, translated, queried, and unified across independently developed systems and services. Successful applications to date include database interoperability, cross database search, and the integration of web services.

CROSS REFERENCES

data model, data modeling, knowledge base, knowledge engineering

RECOMMENDED READING

[1] Berners-Lee, T., Hendler, J. and Lassila, O. [The Semantic Web](#), *Scientific*

American, May 2001. Also <http://www.w3.org/2001/sw/>

[2] Gruber, T. R., [A Translation Approach to Portable Ontology Specifications](#). *Knowledge Acquisition*, 5(2):199-220, 1993. See also [What is an Ontology?](#) <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>

[3] Gruber, T. R., [Toward Principles for the Design of Ontologies Used for Knowledge Sharing](#). *International Journal Human-Computer Studies*, 43(5-6):907-928, 1995.

[4] Guarino, N. [Formal Ontology, Conceptual Analysis and Knowledge Representation](#), *International Journal of Human-Computer Studies*, 43(5-6):625-640, 1995.

[5] Hayes, P. J. The Second Naive Physics Manifesto, in Hobbs and Moore (eds.), *Formal Theories of the Common-Sense World*, Norwood: Ablex, 1985.

[6] McCarthy, J. [Circumscription -- A Form of Non-Monotonic Reasoning](#), *Artificial Intelligence*, 5(13): 27-39, 1980.

[7] McGuinness, D. L. and van Harmelen, F. [OWL Web Ontology Language](#). W3C Recommendation 10 February 2004. <http://www.w3.org/TR/owl-features/>

[8] Neches, R., Fikes, R. E., Finin, T., Gruber, T. R., Patil, R., Senator, T., & Swartout, W. R. [Enabling technology for knowledge sharing](#). *AI Magazine*, 12(3):16-36, 1991.

[9] [Smith, B. and Welty, C. Ontology---towards a new synthesis](#). *Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS2001)*. ACM Press, 2001.

[10] Sowa, J. F. *Conceptual Structures. Information Processing in Mind and Machine*, Reading, MA: Addison Wesley, 1984.

[11] [Standard Upper Ontology Working Group](#) (SUO) IEEE P1600.1, <http://suo.ieee.org/>