INNOVATION AI

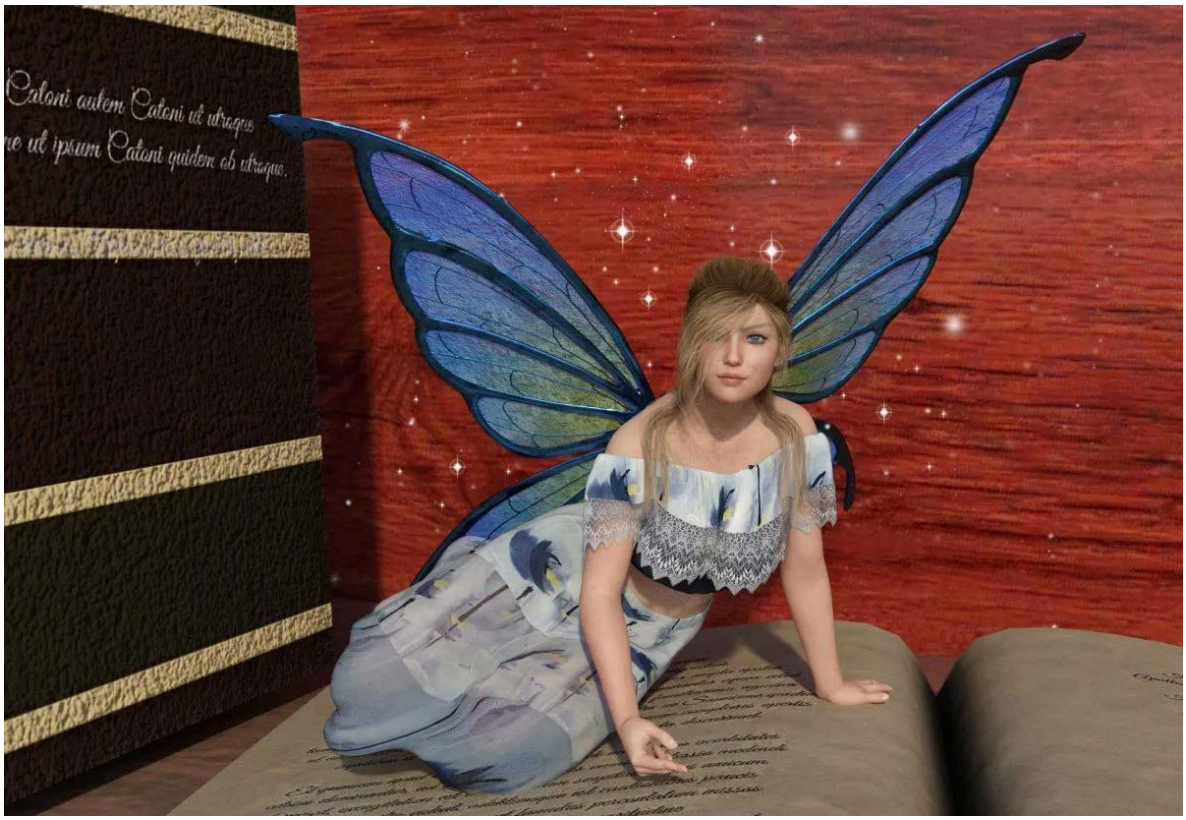
Taxonomies vs. Ontologies

Kurt Cagle Former Contributor
COGNITIVE WORLD Contributor Group

Mar 24, 2019, 07:26pm EDT

Updated Mar 24, 2019, 07:26pm EDT

This article is more than 5 years old.



Any classification taxonomy will face ambiguity: Cornish Pixie or English Fairy? KURT CAGLE 2019

As semantics and linked data become increasingly mainstream, one question that seems to be asked increasingly has to do with a comparatively recent term, ***ontology***, and how ontologies differ from ***taxonomies***. Understanding this distinction is important in making decisions about metadata management, and as such affects anyone who deals with enterprise-level data.

Most people have an intuitive understanding of what a taxonomy is, even if they can't necessarily articulate it formally. The Linnaeus Taxonomy, for instance, should be familiar to any student of biology, as it shows (roughly) how different animals are related. The Dewey Decimal System and the Library of Congress classification system both provide ways to map topical content for books and related media to a continuum. Most governments maintain taxonomies of job titles and classifications, primarily for tax purposes.

Taxonomies classify

More directly, taxonomies provide the terms or categories that a given entity can be described by, and often also describes one or more orthogonal dimensions that provide narrower or broader classification. A cat is a carnivore is a mammal is a chordate is an animal. Dewey Decimal Number 398 describes folklore, 390 describes customs and etiquette, and 300 describes social science, and so forth. Each of these represents some form of class or clade (family).

The role of a taxonomist, in general, is to determine what particular class a given entity (such as an animal or book) most clearly falls into. If this sounds like the work of a librarian, it should: one of the primary roles that a librarian has is to classify new books into a taxonomy. Librarians are taxonomists.

However, before the librarian can do his or her job, there needs to be a taxonomy in the first place. Melvil Dewey was a librarian at Amherst Library when he first developed his library system in the 1870s and would go on to become one of the founders of the American Library Association. Dewey eventually came up with a classification system that mapped topics to a range between 0 and 1000 (mathematicians would say that this was a normalized system).

One of the central notions of any classification system is to minimize ambiguity, or, to put it another way, to minimize the number of buckets that a given resource can be put into, preferably to the point where

there are no overlaps. This is one reason that hierarchies are popular classification tools, and why most knowledge systems ultimately tend to utilize some form of hierarchy. Hierarchical trees provide a way of sub-classing, so that a book on Cornish pixies can readily be inserted into 398.4025 (where each decimal past the point can be thought of as a subtree with 10 branches).

The problem, though, with such knowledge systems is that it is possible for the same work to have more than one such categorization, and worse, that those categorizations have the potential to be in different branches. Are English Fairies the same as Cornish Pixies? Is this a book on folk tales or is it a book on mythology? Is a book on the Disney cartoon character Tinker Bell a book on fairies, a biography (920) (and yes, there are biographies of fictional characters) or a book on animation (773)? This faceting process is one of the challenges that librarians face, because in reality most things can be mutually described by different facets of categorization.

Dewey (and generations of librarians after him) built out a classification system, but knew, even fairly early on, that hierarchies have limitations, in great part because there are ambiguities about what constitutes broader knowledge, but also because such systems tend to be inherently ambiguous.

Ontologies Specify

Ontology as a philosophical concept goes back to the mid-first millennium BC, when the Greek philosopher Parmenides began questioning the nature of "things" - what does and does not constitute a thing. By the time of Plato and Aristotle, the nature of being and existence was one of the most hotly debated questions of philosophy (superseding the ever popular good vs. evil discussions that served as the medieval equivalent of clickbait, cf. Thomas Aquinas). What is a thing? What is a *name*? When does a collection of things become a thing in its own right? These are not easy questions to answer, of course, because reality (or, more properly, our perception of reality) tends to blur around the edges.

This issue of what does and does not constitute a thing kicked into high gear with the introduction of virtual reality. Every time a record is created in a database, it represents the trace of a physical event concerning a particular entity or set of entities. However, what if the thing being described has no physical representation? How do we define an entity?

In the 1990s, Tom Gruber (a professor at Stanford University who would go on to found Siri before that company was bought up by Apple) wrote a paper entitled "Toward Principles for the Design of Ontologies Used for Knowledge Sharing". Within that paper, he laid out the notion of an ontology as being "a specification of a conceptualization". He would later go on to clarify this on the web article What is an Ontology? , in which he argued that taxonomies are inadequate as knowledge representations because of the above complaints, but also because they do not have a formal mechanism for defining concepts at an axiomatic level.

An ontology, then, creates a formal framework that describes anything (not just a taxonomy) by establishing the classes, relationships and constraints that act on the concepts and entities within a given system. One early such ontology, SKOS, the Simple Knowledge Organization System, provides the relational terms and corresponding classes that identify how to create hierarchical (and quasi-hierarchical) taxonomies, in much the same way that the periodic table of elements provides the building blocks for both organic and inorganic chemistry. People who work with relational databases (or spreadsheets) would describe an ontology as the set of tables, columns and key lookups that form a relational schema, while someone familiar with XML would talk about an XML Schema Document (or DTD if they are old school).

In effect, an ontology is the system of classes and relationships that describe the structure of data, the rules, if you will, that prescribe how a new category or entity is created, how attributes are defined, and how constraints are established. An ontologist, correspondingly, is the person who creates the system. You can say in the above example that Melvil

Dewey was, in fact, an ontologist because he established the mechanism that made the Dewey Decimal System possible.

SKOS is an ontology for creating (or specifying) a taxonomy. Dublin Core (now largely superseded by Schema.org) is an ontology for describing intellectual works. A database schema is an ontology for creating records that satisfy the constraints of that database. *The ontology is not the data itself,* but rather the system that defines the columns and tables (classes, loosely) that each row and each primary key/foreign key relationship uses.

Because of the axiomatic nature of ontologies, it is also possible to use one ontology to build another. One of the most primitive ontologies, RDF, or the Resource Description Framework, provides a very minimal set of classes and constraints that are in turn used by other ontologies. Most notably among these is the OWL (web ontology language) specification, which includes a number of logical extensions for describing a comprehensive logical framework. These ontology frameworks are abstract and can be represented in a number of different ways including XML, JSON, Turtle (short for Terse RDF Language), functional or Manchester notation and so forth. More recently, a new framework, SHACL, has been defined that provides a more schematic approach to modeling, holding much the same notation to OWL as XSD has to DTDs in the XML realm.

While they are not quite synonymous, at a working level it's acceptable to replace the term *ontology* with *data model*. Most taxonomists got their start in library science, but most ontologiests come to it from older data modeling disciplines, such as UML, and increasingly, ontologies are becoming acceptable as machine-readable data models on par with other modeling efforts.

Where to Use What

Taxonomies are not dead, but increasingly taxonomies are being written using ontologies and ontological principles. This has several advantages. First, this can help differentiate between entities and categorizations,

which often tend to get munged together when pure hierarchical taxonomies are used. For instance, a car dealership has entities - cars and trucks - that it is selling, each of which has a one to one correspondence with a Vehicle Identification Number (VIN).

If I have a class of vehicles, this is clearly a list of entities. If I say in my ontology that a car has Arctic as an OEM color, those colors are mostly taxonomy terms in the class OEM Color, while another class (generic Color), indicates that Arctic is "essentially" white. If I have an ontology that indicates that something is at its core a term in a taxonomy, then I can extend that class so that I can do all the things that I can do with a term but also have a relationship that says that an OEM Color term has a generic Color term associated with it.

The use of Arctic White illustrates how taxonomies merge into ontologies. SUZUKI 2018

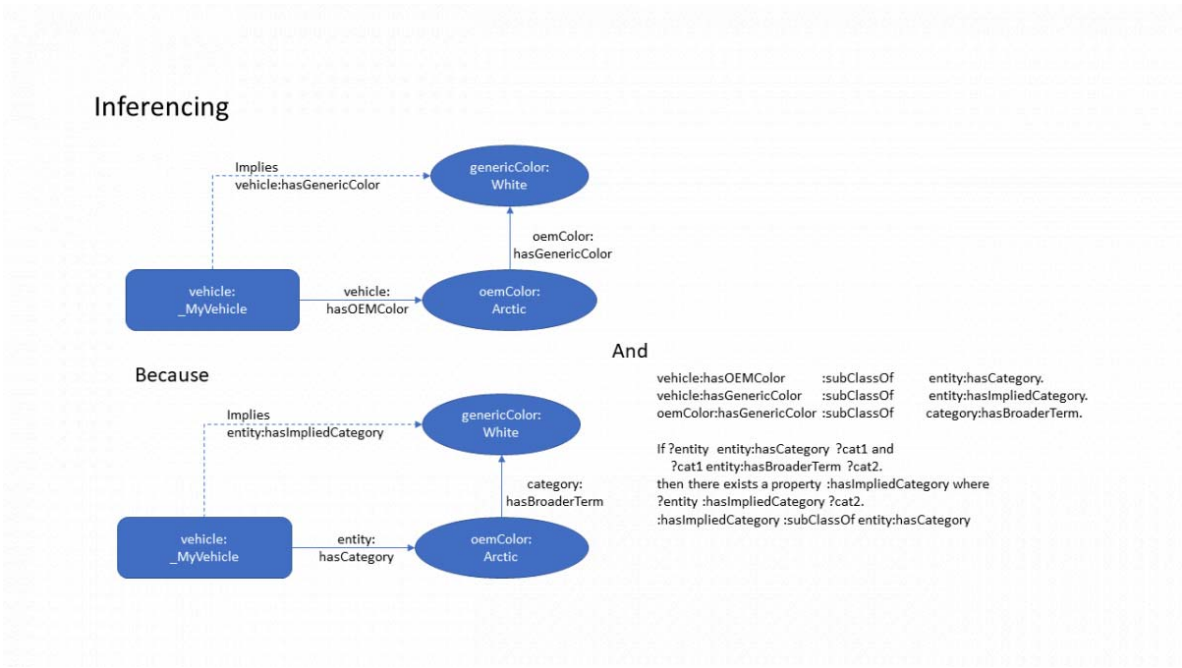
From the standpoint of someone managing an automobile company's IT operations, the need clearly exists for a taxonomist to add an entry to the list of approved OEM colors called "Arctic". It's a term in a taxonomy. However, it also has a particular color associated with it (perhaps a web hex value of #f4f3ef) and maybe an association with a Pantone color (Pantone 9061). A pure taxonomic system couldn't readily capture this, but it's trivial to add in an ontology. Similarly, the extended class OEM Color (from a base SKOS concept) would include the relationship that

connects OEMColor to Generic Color (such as *oemColor:_ArcticWhite oemColor:hasGenericColor genericColor:_White*).

Indeed, the relationship *oemColor:hasGenericColor* can formally be seen as a *skos:broaderTerm* relationship (what in RDF terms would be called a sub-property). This is one benefit of an axiomatic system.

if car with VIN 12591253135.. has an oemColor of Arctic, and oemColor/Arctic has a generic equivalent of genericColor/White, then VIN 12591253135.. has a generic color of white.

The above is an example of *inference:* the determination of new facts based upon existing ones in a logical fashion. It's worth noting that often times such inferences may seem obvious to human beings, but that a machine does not normally have the context necessary to say "Oh, the Arctic is filled with snow and the snow is white, so Arctic probably describes a kind of white."



Inferencing is used to surface potential facts. KURT CAGLE 2019

What that means moving forward is that ontology-enabled taxonomy tools make it possible to make such inferences. They also make possible the ability to uniquely identify a global key so that external data systems can all refer to the same concepts without having to pass around (and potentially duplicate or mangle) complex data structures.

Taxonomies classify. Ontologies Specify.

Ultimately, standalone taxonomy tools will likely be phased out in favor of semantic (i.e., ontologically oriented) systems. The need for librarians doesn't disappear. There is still information that needs to be curated, especially with regards to operational taxonomies, but the skill that such taxonomists bring can be oriented more towards enriching metadata through semantic tools than simply keeping track of terms and human (and frequently highly ambiguous) definitions. This metadata, being machine as well as human, readable, can also drive processes and reduce data redundancy, and can be used to surface new information through inferencing and rich querying/updates.

Follow me on LinkedIn.

Kurt Cagle

Kurt Cagle is a writer, data scientist and futurist focused on the intersection of computer technologies and society. He is the founder of Semantical, LLC, a smart data... **Read More**