

Ontologies on the Semantic Web

Catherine Legg
University of Waikato

Introduction

As an informational technology, the World Wide Web has enjoyed spectacular success. In just ten years it has transformed the way information is produced, stored, and shared in arenas as diverse as shopping, family photo albums, and high-level academic research. The “Semantic Web” is touted by its developers as equally revolutionary, although it has not yet achieved anything like the Web’s exponential uptake. It seeks to transcend a current limitation of the Web—that it largely requires indexing to be accomplished merely on specific character strings. Thus, a person searching for information about “turkey” (the bird) receives from current search engines many irrelevant pages about “Turkey” (the country) and nothing about the Spanish “pavo” even if he or she is a Spanish-speaker able to understand such pages. The Semantic Web vision is to develop technology to facilitate retrieval of information via *meanings*, not just *spellings*.

For this to be possible, most commentators believe, Semantic Web applications will have to draw on some kind of shared, structured, machine-readable *conceptual scheme*. Thus, there has been a convergence between the Semantic Web research community and an older tradition with roots in classical Artificial Intelligence (AI) research (sometimes referred to as “knowledge representation”) whose goal is to develop a *formal ontology*. A formal ontology is a machine-readable *theory* of the most fundamental concepts or “categories” required in order to understand information pertaining to any knowledge domain.

A review of the attempts that have been made to realize this goal provides an opportunity to reflect in interestingly concrete ways on various research questions such as the following:

- How explicit a machine-understandable theory of meaning is it possible or practical to construct?
- How universal a machine-understandable theory of meaning is it possible or practical to construct?

- How much (and what kind of) inference support is required to realize a machine-understandable theory of meaning?
- What is it for a theory of meaning to be machine-understandable anyway?

The World Wide Web

The World Wide Web's key idea arguably derives from Vannevar Bush's 1940s vision of a "Memex"—a fluidly organized workspace upon which a user or group of users could develop a customized "library" of text and pictorial resources, embellishing it with an ever-expanding network of annotated connections (Bush, 1945). However, it was only in 1989 that Tim Berners-Lee (then employed at the Organisation Européenne pour la Recherche Nucléaire [CERN]) spearheaded the development that became the current Web, whose startling success may be traced to the following design factors: hypertext markup language (HTML), universal resource identifiers (URIs), and hyperlinks.

Hypertext Markup Language

HTML provided formatting protocols for presenting information predictably across an enormous variety of application programs. This for the first time effectively bypassed the idiosyncracies of particular applications, enabling anyone with a simple "browser" to read any document marked up with HTML. These protocols were possessed of such "nearly embarrassing simplicity" (McCool, Fikes, & Guha, 2003, online) that they were quickly adopted by universal agreement. A particularly attractive feature of HTML was that formatting markup was cleanly separated from the Web resource itself in angle-bracketed "metatags." Although the term "metadata" became current at this time, the concept it represents—information about information—was of course by no means new, library catalog cards being a perfect example of pre-Web metadata.

Universal Resource Identifiers

HTML took advantage of the Internet's emerging system of unique names for every computer connected to the network to assign each Web resource a unique "location," the interpretation of which was, once again, governed by a simple and clear protocol. This rendered Web resources accessible from anywhere in the world in such a "low-tech" manner that, within just a few years, anyone with a personal computer could download and view any item on the Web and anyone able to borrow or buy space on a server was able to add to the Web resources that would then become instantly available to all Web users (for better or worse).

However, URIs embrace not only Uniform Resource Locators (URLs) but also “Unique Resource Names” (URNs). Whereas URLs *locate* an information resource—that is, they tell the client program where on the Web it is hosted—URNs are intended to serve purely as a *name* for a resource, which signals its uniqueness. Thus, a resource, copies of which exist at different places in the Web, might have two URLs and one URN. Conversely, two documents on the same Web page might have one URL and two URNs, reflecting different perspectives on the documents. Of course, unique naming systems for informational resources have been realized before (for example, the International Standard Book Number [ISBN], International Standard Serial Number [ISSN], the North American Industry Classification System [NAICS], and the United Nations Standard Products and Services Code [UNSPSC]). Unsurprisingly, Semantic Web designers sought to incorporate these older naming systems (Rozenfeld, 2001).

The generality (and hoped-for power) of the URN is such that it is not confined to Web pages but may be used to name any real-world object one wishes to identify uniquely (buildings, people, organizations, musical recordings). It is hoped that, with the Semantic Web, the ambitious generality of this concept will come into its own (Berners-Lee, Hendler, & Lassila, 2001). In short, the push to develop URNs is a vast, unprecedented exercise in *canonicalizing names* (Guha & McCool, 2003). At the same time, however, the baptism of objects with URNs is designed to be as decentralized as anything on the Web—anyone may perform such naming exercises and record them in “namespaces.” This tension between canonicalization and decentralization is one of the Semantic Web’s greatest challenges.

Hyperlinks

The World Wide Web also provides unique functionality for linking any Web resource to any other(s) at arbitrary points. Once again, the protocol enabling this is exceedingly simple and its consequences have been very “informationally democratic,” enabling users to link their resources to any other(s) no matter how official the resource so co-opted. (Thus, for example, a university department of mathematics Web site could be linked to by a page advocating the rounding down of π to 4 decimal places.) However, there is no guarantee that this new resource will be read by anyone.

The World Wide Web Consortium (W3C) (www.w3c.org) was founded in 1994 by Tim Berners-Lee and others to ensure that fundamental technologies on the rapidly evolving Web would be mutually compatible (“Web interoperability”). The consortium currently has over 350 member organizations, from academia, government, and private industry. Developers of new Web technologies submit them to W3C peer evaluation. If accepted, the reports are published as new Web standards. This is the process currently being pursued with the development of the Semantic Web.

The Semantic Web

The Semantic Web is the most ambitious project that the W3C has scaffolded so far. It was part of Berners-Lee's vision for the World Wide Web from the beginning (Berners-Lee et al., 2001, Berners-Lee, 2003) and the spectacular success of the first stage of his vision would seem to provide at least some *prima facie* argument for trying to realize its remainder.

Semantic Web Goals

The project's most fundamental goal may be simply (if somewhat enigmatically) stated as follows: to provide metadata not just concerning the *syntax* of a Web resource (i.e., its formatting, its character strings) but also its *semantics*, in order to, as Berners-Lee has put it, replace a "web of links" with a "web of meaning" (Heflin, Hendler, & Luke 2003, p. 29). As has been noted numerous times (e.g., McCool et al., 2003; McGuinness, 2003; Patel-Schneider & Fensel, 2002), there has traditionally been a significant difference between the way information is presented for *human* consumption (for instance, as printed media, pictures, and films) and the way it is presented for *machine* consumption (for instance, as relational databases). The Web has largely followed human rather than machine formats, resulting in what is essentially a large, hyperlinked book. The Semantic Web aims to bridge this gap between human and machine readability. Given the enormous and ever-increasing number of Web pages (as of August 2005, the Google search engine claims to have indexed over 8 billion), this has the potential to open unimaginable quantities of information to any applications able to traffic in machine-readable data and, thereby, to any human able to make use of such applications.

Planned applications of the Semantic Web range across a considerable spectrum of complexity and ambition. At the relatively straightforward end sits the task of disambiguating searches—for instance distinguishing "Turkey" the country from "turkey" the bird in the example cited earlier (Fensel, Angele, Decker, Erdmann, Schnurr, Studer, et al., 2000). The spectrum then moves through increasingly sophisticated information retrieval tasks such as finding "semantic joins" in databases (Halevy, Ives, Mork, & Tatarinov, 2003), indexing text and semantic markup together in order to improve retrieval performance across the Web (Guha & McCool, 2003; Shah, Finin, Joshi, Cost, & Mayfield, 2002), or even turning the entire Web into one enormous distributed database (Fensel et al., 2000; Guha & McCool, 2003; Maedche, Motik, & Stojanovic, 2003). The most ambitious goals for the Semantic Web involve the performance of autonomous informational integrations over an arbitrary range of sources by software agents (Cost, Finin, Joshi, Peng, Nicholas, Soboroff et al., 2002; Goble & de Roure, 2002; Hendler, 2001).

Not surprisingly, the vision of the Semantic Web's founders is located at the ambitious end of this spectrum. Berners-Lee et al. (2001) have envisaged an automated medical informatics-literate personal assistant that will book a sick mother's medical appointment while rescheduling her daughter's other commitments so that she can accompany her mother and also working out insurance provider issues. Semantic Web supporters see as a further goal the Web services view—to “provide access not only to documents that collect useful information, but also to services that describe or even provide useful behavior” (Klein & Bernstein, 2004, p. 30; McIlraith, Son, & Zeng, 2001). This dimension of the Semantic Web is, however, beyond the scope of this chapter.

It is worth emphasizing that the Semantic Web is expected not to replace but to *extend* the current Web. This aspect of the vision has been expressed by Berners-Lee (2000, online) in a famous “layer-cake diagram” (Figure 9.1).

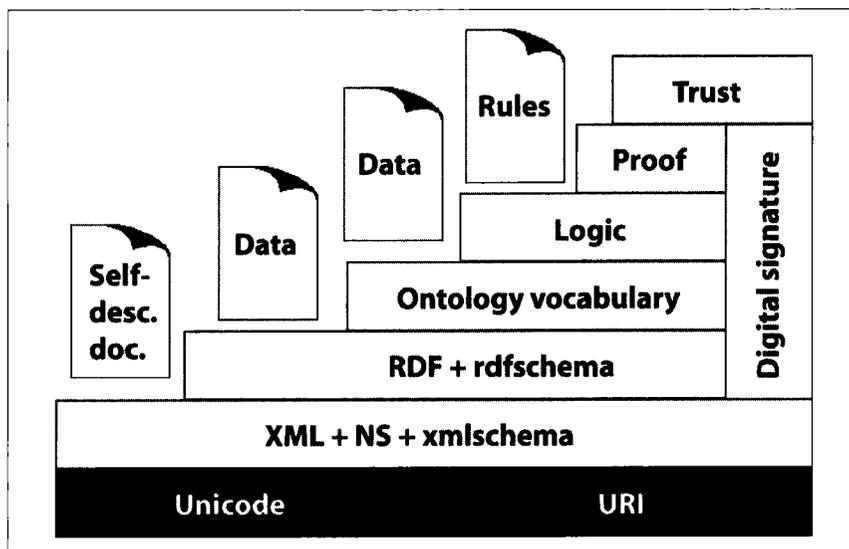


Figure 9.1 Berners-Lee's layer-cake diagram of the Semantic Web.

This diagram (Figure 9.1) represents a series of ever richer stages of informational interoperability. The first layer represents the pre-semantic Web. It provides universal character recognition and the URI system of referencing, which supports hyperlinking. At the second layer, applications exchange metatags but understand them purely qua character strings. At the third and fourth layers, with the introduction of Resource Description Framework (RDF) and ontology vocabularies, meaning is added to the tags, such that applications may be said to begin to understand the terms they are exchanging in metadata. The fifth and sixth layers add the ability to perform inferencing over shared knowledge. The

final layer provides mechanisms for the verification and authentication of knowledge (which will obviously be very important, although space does not permit discussion of it in this chapter).

In a related discussion, Decker, van Harmelen, Broekstra, Erdmann, Fensel, Horrocks, et al. (2000) have defined three levels of requirement that must be satisfied for knowledge to be shared on the Web in machine-understandable form. The first is *universal expressive power*, which consists in being able to express information from any knowledge domain. The second is *syntactic interoperability*, which means that any application can “read” any data by at least parsing them and individuating their words or symbols (p. 67). The third is *semantic interoperability*, the requirement that data be (machine-) “understandable”; the authors do not, however, offer a concrete or operational explanation of this very general definition (p. 67).

On the other hand, Euzenat (2001) has divided the interoperability problem into five separate layers. Layer 1 consists in both *encoding* interoperability, that is, “being able to segment the representation in characters,” and *lexical* interoperability, that is, “being able to segment the representation into words (or symbols)” (p. 20). Layer 2 pertains to *syntactic* interoperability, which is described as “being able to structure the representation in structured sentences (or formulas or assertions)” (p. 20). Layers 3, 4, and 5 are *semantic* in character, that is, “being able to construct the propositional meaning of the representation” (p. 20). Layer 6 is *pragmatic*, that is, “being able to construct the pragmatic meaning of the representation (or its meaning in context)” (p. 20). These last three categories draw on a traditional division in philosophical and linguistic analyses of meaning whereby syntax treats a language’s grammatical or formal-logical structure; semantics, the reference of its words to objects in the external world; and pragmatics, the ways in which the meaning of particular uses of language is influenced by their particular contexts of use (Cruse, 2006; Sowa, 2000).

Discussions of Berners-Lee’s layer-cake vision tend to agree that (1) syntax and semantics are separable, perhaps with further “pragmatic” issues concerning how language is used in context, and (2) some kind of inferencing capacity is vital to genuine representation of meaning on the Web. Beyond that, however, certain criticisms of the framework emerge. For instance, Patel-Schneider and Fensel (2002, pp. 20–21) have observed that exactly how and where “semantics start” has not been worked out and is by no means obvious technically. They have also suggested that analyzing inferencing as a separate layer might be a mistake on the grounds that inference is so tightly intertwined with meaning that it should be possible in the third and fourth layers. Moreover, they contend, “trust” does not belong on the layer cake because it is not a layer of *meaning* but a further issue that should be dealt with by separate applications.

Challenges

As has been noted, the Semantic Web has not yet enjoyed the rapid, seemingly inexorable uptake of the original Web. Four challenges have yet to be met.

The first major challenge is *inferential tractability*. The Semantic Web is not intended merely as a silo for storing and retrieving isolated data. Its realization also requires inferencing to assimilate the data and draw out their logical consequences. It is an understatement to claim that information on the Web is highly distributed. In the best-case scenario, one would like the capacity to answer queries by gathering information from an arbitrary number of unrelated sites and reasoning over the information retrieved by intelligently choosing whatever heuristics are most appropriate on a case-by-case basis. However, any potential implementation of such functionality faces enormous issues of scalability. An important concept here is that of a problem's being *decidable*, that is, formally capable of being evaluated as answerable within a finite time period (Rozenberg & Salomaa, 1994). If a Semantic Web application cannot finish processing a problem because it is not decidable, then that application will be of little use; obviously, Semantic Web applications will need some means of anticipating and bypassing such problems. Moreover, if an application is even merely very, very slow due to the vast quantities of data it is considering, it will also be of little use (Guha & McCool, 2003). An issue with enormous influence on inferential tractability is the logical expressivity of the languages used to represent and to query ontologies. This is discussed in the section on "The Logical Expressivity of Ontology Languages."

Many knowledge representation tools in classical AI made the so-called closed world assumption, which posits that the system knows all that there is to know in a given domain. Such an assumption, although often questionable in practice, makes inferencing considerably more powerful, because an application can assume that if it cannot prove a statement to be true, the statement is false. (A practical example would be allowing a human resources information system to infer that because it does not know that a particular person is employed by a company, he is not so employed.) It is impossible to make such an assumption concerning the Web (Heflin et al., 2003).

The second challenge concerns *logical consistency*. The vision of machine reasoning on the Semantic Web consists primarily in deductive (as opposed to inductive or abductive) inference. (For a defense of this claim, as well as a definition of these three different inferencing forms, see Sowa [2001, 2004].) This may be attributed at least in part to the fact that many projects have their roots in classical AI. However, it is a notorious fact that any proposition can be deduced from a logical contradiction; and, of course, on an information space of the unprecedented size and democratic character of the Web, one will find logically contradictory statements. These will include statements true at different times and circumstances (for instance, "New Orleans escaped major hurricane

damage,” which could be stated truly on August 30, 2005, and “New Orleans did not escape major hurricane damage,” which could be stated truly on August 31, 2005), deliberate misinformation, and apparent contradictions arising from differing interpretations of the same term (for instance, the statements “John Brown was born in 1945” and “John Brown was born in 1967” when the proper name “John Brown” refers to two different men).

A related issue is the *rapid changeability* of information on the Web. Even the *rate* of change of this information is unpredictable—both across pages (some change extremely quickly; others are entirely static) and across time (some sites, after long static periods, change suddenly without warning) (Heflin et al., 2003). Thus, a further dimension of the problem of obtaining information reliably from the Semantic Web, as noted by Guha and McCool (2003), is *predictability*: If a Semantic Web application delivers one answer to a question today and a different answer to the same question tomorrow because the dataset it is encountering is subtly different, this is a problem (unless the world has changed consonantly with the changed answers, in which case this outcome is desirable). This should give one pause, particularly with respect to blithe predictions by developers that the Semantic Web will not constitute a particular application so much as a ubiquitous infrastructure, like electricity (Berners-Lee et al., 2001), because this remit must perforce include so-called mission-critical applications. At least one necessary piece of the solution to this changeability problem would seem to be some kind of ontology versioning (Heflin & Pan, 2004; Noy & Klein, 2004; Noy & Musen, 2004).

The final major challenges facing the Semantic Web are not technical so much as *political*. First of all, who will mark up Web pages with the required semantic metadata? A response might be: Who marked up the World Wide Web with HTML tags? However, this is a different, much more complex task. HTML is learnable in an hour or two, but, as will be seen, to understand Semantic Web languages requires logical, and in many cases also considerable philosophical, *nous*. The obvious solution might seem to be to automate such markup and, indeed, several research programs have vigorously pursued this goal. Some examples include SHOE (which will be discussed in the section on “Ontologies with [Largely] DL Expressivity”), “MnM” (Vargas-Vera, Motta, Domingue, Lanzoni, Stutt, & Ciravegna, 2002), OntoAnnotate (Staab, Maedche, & Handschuh, 2001), and SemTag (Dill, Eiron, Gibson, Gruhl, Guha, Jhingran, et al., 2003) (see also Roesner, Kunze, & Kroetzsch, 2005; Witbrock, Panton, Reed, Schneider, Aldag, Reimers, et al., 2004). However, such automated markup might seem to require scaffolding by a working Semantic Web, creating an infinite regress problem.

Moreover, the ambitious directive to generalize URIs from mere address locators for downloading files to canonical names for real-world things is fraught with political implications. In stark contrast to the Web, where one genius of the system was that no preexisting relationship was

necessary between producers and consumers of information, canonicalized names make necessary “a very deep relationship,” namely that producers and consumers of data must agree on the references of all URIs (McCool et al., 2003, online). How is such agreement to be achieved?

It is widely appreciated that until the Web includes a significant quantity of semantic metadata, developers have little incentive to produce applications for the Semantic Web; but if few Semantic Web applications exist, there is little incentive for Web authors to mark up pages semantically. Ironically this same fact is taken by some as reason to be pessimistic about the future of the Semantic Web, while others take it as reason for optimism, on the grounds that bootstrapping will quickly accelerate development (Berners-Lee, 2001). Bray (2001) holds the Yahoo!’s hand-coded subject directories (done at considerable expense to the company) return very few hits as a consequence of the human mediation. Nonetheless, people like to use these directories, Bray contends, suggesting that the same will be true of information on the Semantic Web. Other commentators, such as Shirky (2003, online), warn of the “this will work because it would be good if it did” fallacy.

Semantic Web Technologies

The Semantic Web’s core technology is, as noted, a generalization of markup tags beyond indicating a Web resource’s intended *formatting*, to indicating its intended *meaning*. This has occurred primarily through the development of a series of new markup languages.

Extensible Markup Language

Extensible Markup Language (XML) (Bray, Paoli, Sperberg-McQueen, Maler, & Yergeau, 2000) was originally designed as a replacement for Standard Generalized Markup Language (SGML), which was used to share documents within government and the aerospace industry (Bailey, Bry, Furche, & Schaffert, 2005). XML was not developed specifically for the Semantic Web but has been embraced by its developers. It was initially conceived as a simple way to send documents across the Web (Swartz, 2002). In contrast to HTML’s decreed list of formatting tags, XML allows Web authors to define their own tags and, thus, their own document formats, subject to a syntax specified in the XML Recommendation. Originally, these tag definitions were registered in a separate document called a Document Type Definition (DTD). However, this system has now been superseded by XML Schema. When a document conforms to XML syntax, it is said to be “well-formed”; when, in addition, it conforms to a set of tag definitions, it is also said to be “valid.” Teasing apart these criteria might seem to enable the separation of a document’s syntax from its semantics, at least in principle (Sowa, 2000).

As an illustration, one might define the following tags:

```
<album> </album>
<artist> </artist>
<genre> </genre>
```

One may then—without using any kind of database software infrastructure, in other words, in a flat file—construct a short document describing a three-item music collection containing the following fields:

```
<album> I
  <artist> The Magnetic Fields </artist>
  <genre> Alternative </genre>
</album>
<album> Midnight Love
  <artist> Marvin Gaye </artist>
  <genre> R&B </genre>
</album>
<album> Hounds Of Love
  <artist> Kate Bush </artist>
  <genre> Rock </genre>
</album>
```

Such a document is said to have three main elements (`<album>`), each of which possesses two child elements (`<artist>`, `<genre>`). XML also allows attributes for tags to be defined, as in the following example:

```
<album genre='Rock'> Hounds Of Love
  <artist> Kate Bush </artist>
</album>
```

However, it is generally preferred to define new child elements rather than using attributes in this way. At any rate, each XML document essentially consists of a document tree, with a distinguished root element, containing elements, character data, and attributes (Bailey et al., 2005).

In these examples, data such as the character strings “Kate Bush” and “Marvin Gaye” are obviously receiving *some* kind of “meaning-tagging.” A human reader will probably infer that the `<artist>` tag applied to “Kate Bush” means that it names the artist who recorded the album listed immediately afterward. But what meaning do these tags have for a machine? As stated so far, absolutely nothing. So they need to be linked to some kind of definition. This function is served by XML namespaces. I can define a namespace (which I shall call “mc”) by inserting a tag in my document as follows:

```
<h:html xmlns:mc=HYPERLINK "http://www.musiccollections.com/
myCDs" http://www.musiccollections.com/myCDs"/>
```

I can then prefix all tags in my document with the name of my namespace as follows:

```
<mc:album> I
    <mc:artist> The Magnetic Fields </mc:artist>
    <mc:genre> Alternative </mc:genre>
</mc:album>
[...]
```

Although the term “namespace” might suggest some further document that includes definitions for the terms in tags, in practice a namespace is often just a URI, which need not point to any further location. Thus, namespaces are essentially just a way of indexing different tags (uniquely) via prefixes.

Of course, any number of XML namespaces can exist. Anyone can define one. So, how do these namespaces relate to each other, semantically speaking? Should it be stipulated that two tags from different namespaces have the same meaning if and only if they consist of the same character string? This would, of course, be a bad idea. The <artist> tag in my namespace “means” (i.e., is intended to apply only to) musicians, whereas an <artist> tag in another namespace might “mean” (i.e., be intended to apply only to) painters. Conversely, another person might document a music collection using the tag <musician> with the same meaning with which I have used my <artist> tag. Thus, to exactly the same degree that namespace prefixing allows for clear distinctions among tags that use the same character string, it drastically reduces tag shareability. Every namespace’s tags are now quite distinct. How may translation between them be effected?

In spite of XML’s new freedom to define tags at will, it still arguably provides syntactic interoperability at best (Swartz, 2002). We have seen that Decker et al. (2000) distinguish the interoperability requirements of universal expressive power, syntactic interoperability, and semantic interoperability. They argue that XML satisfies the first and the second, but not the third, requirement; moreover, they hold that the advantage of using it consists only in “the reusability of the parsing software components” (p. 68). In their estimation, it is “useful for data interchange between parties that both know what the data is, but not for situations where new communications partners are frequently added” (Decker et al., 2000, p. 68; see also Heflin et al., 2003).

This should not be too surprising because, as already noted, XML was not designed to share meaning so much as document format, and the latter is, in fact, a concept of enormous generality that embraces any kind of structure within data (including for instance, a document containing just four elements). In other words, XML allows for no principled distinction

between “content-specific” and “presentation-specific” tags (Heflin et al., 2003, p. 30).

XML Schemas

Attempts to remedy XML’s lack of semantic bite have been made by building a series of “schemas” that can be added to it. The most widely used is simply and somewhat confusingly entitled “XML Schema” because it was the first to achieve Recommendation status by the W3C, in May 2001 (Fallside, 2001); a second edition was released in October 2004 (Fallside & Walmsley, 2004). It was largely developed by Microsoft. An XML Schema instance is called an XML Schema Definition (XSD). Each application of XML Schema to a given XML document produces a further file that lists its vocabulary (i.e., element and attribute names), its content model (i.e., relationships and data structure), and its data types. This file is called the Post-Schema Validation Infoset (PSVI), and enables interoperability with many object-oriented programming tools. The following very simple XML document:

```
<h:html xmlns:mc=http://www.musiccollections.com/myCDs>
  <<mc:album> I
    <mc:artist> The Magnetic Fields </mc:artist>
    <mc:genre> Alternative </mc:genre>
  </mc:album>
```

may be described using the following XML Schema document:

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="album" type="xs:string"/>
  <xs:complexType>
    <xs:sequence>
      <xs:element name="artist"
        type="xs:string"/>
      <xs:element name="genre"
        type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

As may be evident, it is, unfortunately, often easier for a human reader to understand an XML document than its schema. Further XML

schema languages include the Japanese RELAX (Regular Language description for XML, www.xml.gr.jp/relax), TREX (Tree Regular Expressions for XML, <http://thaiopensource.com/trex>), and a later merging of these two called RELAX NG (Clark & Makoto, 2001). These languages are popular because they are considerably simpler to use than XML Schema itself; nevertheless, they are not easy to learn.

It can be seen that XML schemas categorize XML data in *some* sense. However, this is little more than the garden-variety data typing performed by relational databases (for example, “sequence” or “ComplexType”). What does such typing tell us about the *meaning* of data? Such reflections raise the interesting research question of how, in general, one distinguishes between the structure and the meaning of information in a principled way (not to mention in a machine-understandable way). In spite of the prima facie clarity of Berners-Lee’s layer-cake model, this has turned out to be a thorny issue in practice (see, for instance, Patel-Schneider & Fensel, 2002, discussed in greater detail in the section “DAML+OIL/OWL”). Research into such questions is rendered even thornier by the philosophical possibility some (albeit surprisingly few) commentators have raised—namely, that the structure of information might form part of its meaning. For instance, Sowa (2004, p. 13) has suggested that it will be necessary to find a way to represent the “structure or geometry” of data in order specifically to reproduce analogical reasoning, which he argues forms a considerable part of human inference and, thus, meaning (see also Sowa & Majumdar, 2003). Thus, Semantic Web developers have sought to develop ways of representing meaning in a purer and more explicit fashion than is possible in XML; this brings us to the topic of RDF.

Resource Description Framework

Work on Resource Description Framework (RDF) was initiated by R. V. Guha while at Apple Computer, and its first version attained W3C Recommendation status in 1999. A new version (Beckett, 2004) was published as a set of related specifications in 2004.

Strictly speaking, RDF is not a language but a data model for describing a machine-readable semantics for “Web resources” (in the very general sense of “resource” identified in the earlier discussion of URIs). Mathematically speaking, this model consists of a directed graph of nodes connected by labeled arcs. (For a textbook introduction to these concepts, see Bollobas [2002].) In a key advance on XML, RDF introduces propositional structure into its data. Each RDF “proposition” has three parts. These are referred to either as “subject,” “predicate,” and “object” (e.g., Beckett, 2004; Swartz, 2002) or as “object,” “attribute,” and “value” (e.g., Decker et al., 2000). The subjects/objects and the objects/values should be understood to lie on the nodes of RDF’s directed graphs and the predicates/attributes on the arcs. Semantically speaking, in each proposition the subject/object should be understood as the “thing” the proposition is about, the predicate/attribute as a property

that is ascribed to the subject/object, and the object/value as a value (or some other qualifier) assigned to that trait as it pertains specifically to that subject/object. As an example, take the proposition:

Kate Winslet's age is 27.

Here the subject/object is Kate Winslet, the predicate/attribute is age, and the object/value is 27. Thus, one might argue that each RDF proposition is equivalent to one cell in a database table; the fervent hope exists one day to “slurp” the world’s databases onto the Semantic Web in RDF format (Hendler, Berners-Lee, & Miller, 2002).

When RDF graphs are made available on the Semantic Web, their nodes will consist of URIs describing Web resources; so-called “literals” (scalar data such as strings or numbers); or blank (unlabelled) nodes, which may be used to group or “aggregate” properties (Bailey et al., 2005). Thus, representing the proposition about Kate Winslet will consist in representing a ternary relationship among entities such as that illustrated in Figure 9.2.

An RDF statement can itself become the subject/object or object/value of a triple, a process known as “reification.” This is, of course, necessary if there is to be any reasoning about trust or levels of confidence in the propositions on the Semantic Web (Berners-Lee’s sixth semantic layer). Reification does, however, introduce considerable awkwardness into RDF implementation: When an RDF graph is traversed (i.e., read by an application) via statements regarding a given property, reified statements are traversed in a way different from that used for unreified ones (Garshol, 2005).

Because RDF is a data model rather than a language, it needs to be implemented (“serialized”) in a language. It is often serialized in XML, as it is in the W3C’s official specification (Beckett, 2004). However, this implementation is notoriously unwieldy with regard to syntax (de Bruijn, 2003). Somewhat less complex serializations include N3 and its even simpler subset N-Triples (Berners-Lee, 2001, 2005). Another language, developed at the University of Bristol is TURTLE (Terse RDF

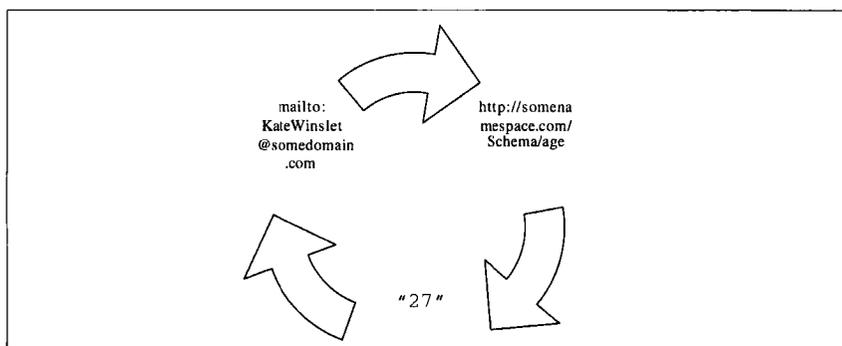


Figure 9.2 Relationships among entities representing the proposition about Kate Winslet's age.

Triple Language), which slightly extends N-Triples (Beckett, 2003); many other serializations have been proposed. In terms of the Semantic Web's goal of seamless semantic interoperability, applications that use RDF should perform serialization in independent fashion. However, this is not yet the case in practice (Bailey et al., 2005).

Bray (2001) has argued that RDF's three-part propositional structure renders it very scalable because arbitrary numbers of triples can be "stacked" in any order. By way of contrast, in XML, the order in which elements appear in a document is significant: This vastly complicates data interchange. The data structures definable in XML are also considerably more complex and syntactically unwieldy, for they can consist of arbitrary mixes of trees, graphs, and character strings. Haarslev and Moeller (2003) have also pointed out that RDF's triple-oriented style of data modeling offers means for providing graph-structured data over multiple documents; on the other hand, XML can only express graphs within a given document.

A standard RDF query language does yet exist; each language is typically tied to a particular implementation. A useful and thorough overview and comparison of query languages for XML, RDF, and Topic Maps (discussed in the section on "Ontologies with [Largely] DL Expressivity") is provided by Bailey et al. (2005), who work through a suite of test questions using each language in turn.

In spite of RDF's advantages over XML, it would be premature to conclude that RDF makes possible a "Semantic Web." In the sample proposition about Kate Winslet, a propositional structure exists with two of the three components assigned a URI. However, we have seen that URIs are just simple indices. To what are they indexed? RDF does not determine this. The problems of polysemy for metadata terms composed of identical character strings and hidden synonymy for terms composed of different character strings, which were noted with respect to XML, have not yet been resolved. Moreover, Heflin et al. (2003, p. 32) have observed that RDF provides "a very small set of semantic primitives" and, for this reason, it is logically quite inexpressive. Indeed, it does not even support an inheritance hierarchy. (This concept is explained in the section on "History of Ontology.") It has also been noted that RDF has relatively weak mechanisms for managing "schema evolution" (i.e., ontology versioning).

Something more is needed in order to determine meaning fully. Ontologies have been designated the semantic "silver bullet" (de Bruijn, 2003, p. ii). Therefore, in order to implement the Semantic Web's "ontology layer," developers have set themselves the task of building a number of further languages. These include RDF Schema and DAML-OIL/OWL, which are discussed later. First, however, an introduction to, and discussion of, ontologies in general is presented.

Introduction to Formal Ontology

History of Ontology

Ontology (which is generally, but not universally, treated as synonymous with “metaphysics”) originated in a branch of philosophy as old as Aristotle, which he called “first philosophy.” Etymologically, it derives from the Greek words *όντος*: (participle of *ειναι*: “to be,” i.e., “being”) and *λόγος*: (“word/speech”). Ontologists seek to build a theory, at the most general level, of all the different types of things that exist (including the relationships among the different types of things that exist). As Smith and Welty (2001, p. iii) have put it, “philosophical ontology is the science of what is, of the kinds and structures of objects, properties, events, processes and relations in every area of reality” (see also Smith, 2003b). The terms coined or adapted by ontologists to refer to these most basic “kinds and structures” are called *categories* (terminology which has carried over into ontology in information science). “Material Object,” “Person,” “Time-point,” and “Number” are just some of the basic categories proposed by traditional ontologists.

Aristotle’s work “Categories” arguably represents humanity’s first attempt to determine a systematic formal ontology. It is worth noting that this work was intertwined very much with Aristotle’s logic—thus, understanding the inferential rules by which knowledge falling under certain categories might be transformed into further, categorized knowledge was for Aristotle inseparable from understanding the categories themselves. Interestingly, although later philosophers sought to pursue ontology and formal logic as separate disciplines, formal ontology in information technology applications has been forced to reunite them. Aristotle mapped out the first systematic logical distinction between the subject and predicate of a proposition—a significant advance in Western thought. Building on this, he defined his most basic categories (“Substance,” “Quantity,” “Quality,” “Relation,” “Place,” “Time,” “Posture,” “State,” “Action,” and “Passion”) as groupings of predicates according to what kinds of entities they properly pertain. To apply a predicate from one category to entities from another (e.g., “The number 8 is red.”) produces a nonsensical so-called category error, something that will, of course, need to be discerned on the Semantic Web.

Famously, it was Aristotle who first formally defined the genus/species/differentia framework for categorizing groups of objects (whereby a species is defined by giving its genus—the kind under which the species falls—and its differentia—what distinguishes that species from others within that genus) (Smith, 2004). Thus were the sciences of classification born: For the first time, knowledge could be organized *taxonomically*—that is, into a hierarchy where knowledge “inherits” (is inferable downward, through an arbitrary number of genus–species relationships—a meaning that the term “inheritance” retains today in knowledge engineering). This feature alone greatly increased the organization and power of knowledge systems. Eventually, although not

until the 19th century, this logical framework was developed and mathematized into modern set theory, which, as we shall see, is the basis for much formal ontology today. Most ambitiously, Aristotle sought to organize *all* branches of knowledge taxonomically, discerning the basic principles that defined each and organizing them into an integrated system (Smith, 2004; Sowa, 2001).

Medieval philosophy was built very much on Aristotle's ontological foundation. Genus–species relationships were considerably elaborated and organized into a “tree of knowledge” (Sowa, 1999, book cover). However, in the early modern period, development in formal ontology as an intellectual discipline suffered a setback, for the Scientific Revolution's new understanding of (and consequent enthusiasm for) scientific experiment led to the spread of *empiricism*—the idea that genuine knowledge is gained only through real-world experience—and a corresponding loss of focus on formal inquiries. Very few real advances in logic occurred during this period.

Such empiricism was countered to some degree by rationalists such as Kant, who sought to demonstrate that learning from experience already presupposed a prior (possibly innate) conceptual scheme and devised his own system of categories (Kant, 1998; Sowa, 2004). The early empiricists were, on the whole, logical atomists who saw knowledge as decomposable into basic building blocks (namely individual sensory experiences); Kant, however, took a more holistic view. A physical object such as a chair can be split in half, Kant noted, but what, he asked, would constitute half a proposition? The meaning of a proposition (e.g., “Trees are green.”) is not reducible to the meanings of its subject (“trees”) and its predicate (“are green”) but constitutes a unity provided by the mind's understanding of their combination, he suggested.

Nevertheless, the spirit of the Scientific Revolution continued to pervade many currents of philosophical thought, culminating, in the early twentieth century, in logical positivism's wholesale rejection of speculative metaphysics (Schlick, 1936). During the formative years of mainstream analytic philosophy, the enormously influential Quine, partaking of this disdain for speculative metaphysics, presented his famous logical criterion of ontological commitment according to which the only mode of being is “to be the value of a [bound] variable [in our best scientific theory]” (Quine, 1953). Thus, the work of formally defining categories fell out of favor in mainstream philosophy, with the odd notable exception (e.g., Chisholm, 1996; Lowe, 1997, 1998; Weiss, 1958).

Formal logic, however, had received a new infusion of life toward the end of the 19th century with the invention of predicate logic by Frege (1970) and Peirce (1931–1958). This was the first major advance in logic since Aristotle, generalizing away from the Aristotelian framework of unary relations and two-premise arguments to relations of arbitrary complexity and arguments of arbitrary length, and it generated tremendous excitement. Set theory was invented, with the intent of using it to found all of mathematics on logic. Frege himself dreamed that this

would form the basis of a genuinely clear and objective theory of meaning that would “break the domination of the word over the human spirit by laying bare the misconceptions that through the use of language often almost unavoidably arise concerning the relations between concepts” (Frege, 1970, p. 20). Frege hoped that it would now be possible to build all knowledge into an integrated, *deductively complete*, taxonomic system (i.e., one in which everything true might be proven to be true). Alas, this dream was shattered when Russell uncovered a paradox at the heart of the new logical system that rendered it inconsistent—namely, the famous class of all classes that do not contain themselves, which must perforce both contain and not contain itself (Zalta, 2005). Nevertheless, as we shall have occasion to see, Frege’s dream has had a tendency to reemerge again in knowledge engineering circles.

Peirce, the other inventor of predicate logic, pursued a rather different vision. His philosophical pragmatism led him to see any attempt to formalize the entire meaning of a body of knowledge as impossible. In his view, an irreducible dimension of the meaning of any term (such as “hard” or “magnetic”) is constituted by the effects that an agent situated in the world would experience in relevant situations and the sum total of such effects can never be known in advance (or there would be no need for scientific inquiry) (Peirce, 1940). Moreover, one of Peirce’s major philosophical ambitions was to critique such aprioristic overreaching, which, for him, was epitomized by Descartes’s famous claim for the philosophical necessity of eliminating all doubt as a precondition for serious inquiry. He also rejected attempts to reduce all useful inference to deduction, claiming that induction—generalizing from past to future cases of a relevantly similar kind (a process that, in contrast to deduction, requires experience)—and “abduction”—the generation of possible explanations of phenomena—were equally important (Peirce, 1940). As has been mentioned, classical AI was excessively deductivist in its approach to knowledge; we will see that Semantic Web developments so far have not yet departed significantly from this stance.

An aspect of the Cartesian philosophical framework to which Peirce objected with especial vehemence was its model of *meaning*, whereby the meaning of a sign is determined by the intention of the person who uses it (for Descartes himself, this intention consisted in an idea in the user’s mind so private and inaccessible as to constitute a non-physical substance). Peirce sought to replace this with a new, publicly accessible, model of meaning. Whereas Descartes’s model was dualist, with meaning consisting fundamentally in a binary relationship between a user’s intention (a sign in the mind) and an object in the world, Peirce’s was triadic, with meaning consisting in a relationship between a sign, an object signified, and further uses of the same sign *by others* to “mean” the same object. (This crucial third term in his meaning model he called the sign’s *interpretant*.) An important difference between these models is that, whereas the Cartesian makes the comforting assumption that, by virtue

of our intentions, we determine and have the ultimate authority as to what we mean, Peirce's implies that the meanings of one's signs consist only in their continued use by others. Thus, the meaning a sign has for initial users might be very different from the meaning it is interpreted to have subsequently, scientific terms providing a very good example of this (Peirce, 1940; Legg, 2005). The issue of predictability versus unanticipated development in meaning is highly relevant to formal ontology's goal of defining meaning in machine-understandable form.

A final issue worth mentioning—and it is an understatement to say that it has bedeviled philosophical ontology from the beginning—is the question of the degree to which the categories devised by human ontologists should be thought of as universally applicable or objective, as opposed to artifacts of particular contexts (such as a culture, a time period, a species, or a set of perceptual capacities). To what extent should an ontologist's categories be regarded as valid for all time and shareable across all communities? Or must all ontological work be done “locally” and repeatedly? This is the question of *realism*; 2,000 years of continuing controversy have not discouraged philosophers from hotly debating it today. Snapshots of some recent skirmishes in this debate are provided by Rorty (1990), Harré and Krausz (1996), and Kirk (1999).

Formal Ontology in Information Technology

In the late 1950s, formal ontology began to be reinvented in computer science (a field itself born from the 19th-century advances in logic mentioned earlier). As databases became more sophisticated and attempts to integrate them more ambitious, it was noted (McCarthy, 1995; Smith, 2003b) that certain problems encountered by database designers were actually ontological in character. For example: What is a person? Is it an entity that exists wholly at a particular time (in which case one can conveniently assume that a person has only one address)? Or is it an entity that covers a whole “person-lifetime” (in which case databases will need somehow to deal with one “person” having multiple addresses)? Are organizations individuated by the physical locations of their buildings, in which case it is not possible to have two organizations in one location? Or are they individuated by some other means and, if so, how? (For example, what is the relationship between Microsoft Germany and Microsoft U.S.A.?) Some general standardized conceptual scheme is considered desirable, at least to prevent random, ad hoc solutions to ontological problems being “hacked” by computer programmers.

Perhaps less surprisingly, ontology became a topic of interest in the field of AI. Famously, after early inspiring successes on relatively simple problems, AI encountered difficulties that led to a humbling scaling back of goals and enthusiasm (and also funding) (Lenat, Guha, Pittman, Pratt, Guha, & Shepherd, 1990). In the late 1970s and early 1980s, concentrated research effort was put into so-called “expert systems,” which sought to represent knowledge of a domain in such a way that questions

about it could be answered at the level of a human expert. Examples included DENDRAL for organic chemistry (Lindsay, Buchanan, Feigenbaum, & Lederberg, 1980) and MYCIN for medical diagnosis (Buchanan & Shortliffe, 1984). These systems achieved success in the narrowly defined domains for which they were designed but became brittle when applied to new problems. A major stumbling block was delivering “artificial understanding” of natural languages such as English. It was realized that much natural language understanding draws on a background framework of general knowledge about the world (even at the level of simple sentence parsing, as evident in the differing interpretations we naturally give to the sentences “Aroha and Fiona are mothers” and “Aroha and Fiona are sisters”).

A few farsighted individuals began to look to philosophy for aid in constructing an artificial general knowledge framework (McCarthy, 1995; Sloman, 1995). Some of this new thinking crystallized in a series of formal ontology projects, the largest and most ambitious of which was the Cyc project, whose target was identified by Lenat as capturing “common sense”: everything a six-year-old knows that allows her to understand natural language and start learning independently (Lenat, 1995; Lenat & Guha, 1990). (The Cyc ontology is discussed in the section “Ontologies with First-Order Logic Expressivity [or Higher].”) These ontology projects drew on AI researchers’ attempts to represent knowledge in “declarative” languages explicitly modeled on formal logic (and subject to its inferencing rules). This embrace of formal logic proceeded in stages. The earliest knowledge representation systems were semantic networks (Quillian, 1967) and “frame systems,” the latter so called because they embodied a collection of named data structures (frames), each of which had a series of “slots” representing attributes of the entity represented by the frame, into which values were inserted (Minsky, 1975). It was soon realized, however, that the lack of formal semantics underpinning these systems rendered them unsatisfactory. Reasons for this included profound ambiguity: For instance, if the value “green” is inserted in the “color” slot within the “frog” frame, does this mean that frogs *must* be green or merely that frogs *can* be green? Genuine confusion existed here.

The first formal semantics specifically aimed at the field of knowledge representation was developed by Brachman (1978) and resulted in a language known as KL-ONE (Brachman & Schmolze, 1985). Around this time, the idea that it might be of benefit to restrict logical expressivity in favor of inferential tractability began to emerge; unfortunately, it was shown that reasoning even in KL-ONE was undecidable (Schmidt-Schauss, 1989). From this line of research developed a branch of formal logic known as Description Logic (DL). DL is a decidable fragment of first-order logic with a series of versions of greater and lesser expressivity. Brachman and Levesque (2004) give an overview of 20 years of development in this area. Detailed investigations into the formal features and consequences for decidability of different versions of DL include SHOQ

(D) (Horrocks & Sattler, 2001), SHIQ (Horrocks, Sattler, Tobies, 2000b), and ALL (Horrocks, Sattler, & Tobies, 2000a). Borgida (1996) provides a comparison of the expressiveness of DLs and traditional predicate calculus.

However, as the field of symbolic “knowledge representation” developed, it was found necessary to add more and more logical expressivity to representation languages in order to deal with the full range of typical human assertions, including negations (e.g., “Henry has no jazz CDs.”), disjunctions (e.g., “This CD is either jazz or pop.”), and assertions about assertions (e.g., “Ruth’s claim that Henry has no jazz CDs is not true.”). As a result, the field soon acquired languages with the expressivity of full first-order, and even higher-order, logic. Examples here include Stanford’s Knowledge Interchange Format (KIF) (Genesereth & Fikes, 1992), knowledge management (Clark & Porter, 2001), jointly developed at University of Texas at Austin and Boeing, and CycL, the language of the Cyc project. The difficulty of inferring over these languages soon became more than apparent and, as the field of AI pushed on into the 1990s and turned from the explicit symbolization of “Good Old-Fashioned AI” toward brute-force, number-crunching approaches epitomized by machine learning, these projects began to languish. However, the push to create the Semantic Web has reignited interest in them, although to what degree they can be adapted to present needs remains to be seen.

The majority of formal ontologists within information technology work with Gruber’s (1993, p. 199) simple definition of an ontology as “the specification of a conceptualization.” This definition takes a position on the realism question, by assuming that whatever is conceptualized as reality by a particular group is all the reality there is to represent (see also Gruber, 1995). Despite its “irrealist” approach, it does satisfy Quine’s criterion of ontological commitment (and one might, of course, ask how one might represent things one does not conceptualize). Even so, Gruber’s definition has critics in the formal ontology community, for instance Smith (2003b), who does, however, concede that it might be appropriate for certain domains where reality is arguably wholly human-created, such as administrative systems.

The Logical Expressivity of Ontology Languages

As noted, the scalability of applications that use ontologies is strongly influenced by the logical expressivity of ontology languages (both representation and query languages, if the latter differ from the former, as they frequently do). In Figure 9.3, McGuinness (2003) usefully taxonomizes ontologies across a spectrum of increasing expressivity.

The first and simplest category, *catalogs*, consists of finite lists of terms that are used as a controlled vocabulary, although no attempt is made to define the terms. The next category, *glossaries*, consists of lists of terms along with a meaning for each term stated in natural language.

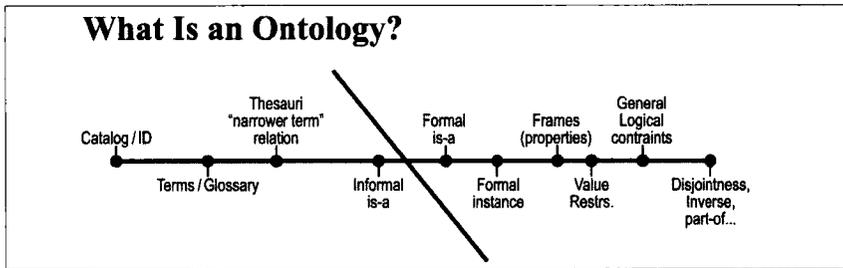


Figure 9.3 McGuinness's taxonomy of ontologies.

Thesauri add to glossaries some rudimentary semantic relationships, such as synonymy, between terms (thus, a thesaurus for the music domain might point out that “CD” and “compact disc” mean the same thing). Not infrequently, thesauri also enable informal “is-a” relationships (that is, membership of kinds or classes) to be discerned by a human user but not reasoned over in any machine-processable way.

To the right of the diagonal line, one begins to add to one's ontology properties and relations that can be reasoned over formally. The first addition is formal subclass and class membership relationships. We have now reached the complexity of frame systems. If the subclass relationship is rendered transitive, the ontology provides inheritance (sometimes also referred to as a “subsumption hierarchy”). The next logical sophistication generally added is so-called “value restrictions,” whereby assertions are made about the domain and range of the ontology's relations (for instance, “Every named artist on a CD is a person”). After this, it is customary to add the standard inter-set relationships of classical set theory. These include *union* (for instance, “The category of ‘80s music’ consists of releases from 1981, and from 1982, and from 1983, ... etc.”), *intersection* (for instance, “The category of ‘music recommended by dorm 10’ consists of those songs that are on the iPods of every resident of dorm 10”), and *disjointness* (“No jazz CDs are heavy metal CDs”). We have now reached the complexity of some versions of DL.

One then reaches ontology languages with the expressivity of full first-order logic. First-order logic is already undecidable; however, it is possible to proceed even further, into higher-order logic, by including such features as quantifying over properties, (“All songs in Cathy's collection have something in common.”), propositions about propositions (“Not all the statements Cathy has made about her CD collection are true.”), modal logic (“It is possible that Miles Davis might release a hip-hop CD.”), and context logic (“Whereas in the novel [X], all Kate Bush CDs were recorded by David Bowie, actually Kate Bush's CDs were recorded by Kate Bush.”). The problems of inferential tractability are so compounded here that very little research has been done on such languages (in comparison to the extensive work done on DLs). Nevertheless, some ontologies have ventured this far.

Although “real-world” ontology languages do not always fit cleanly into these boxes, this framework is used here to taxonomize ontologies on the Semantic Web into three broad categories that structure the next three sections—namely, “thesaurus ontologies,” ontologies with (roughly) the expressivity of DL, and ontologies with the expressivity of full first-order logic or higher. It is worth noting that the catch-all term “ontology” is somewhat misleading with respect to these three sections, as some of the developments discussed are merely ontology languages with no real knowledge represented in them (for instance, OWL); some comprise languages plus more or less extensive knowledge bases (for instance, SUMO); others (such as Cyc) constitute full knowledge representation systems (containing further tools such as inference engines, natural language interfaces, and ontology editors).

Thesaurus Ontologies

Dublin Core Metadata Element Set

The Dublin Core Metadata Initiative (DCMI) (<http://dublincore.org>) is a loosely affiliated group of researchers located in a variety of institutions worldwide, whose aim is to formulate interoperable metadata standards. To this end, they have dedicated themselves to developing metadata-sharing frameworks and technologies, constructing core specialized metadata vocabularies, and collecting and registering the metadata vocabularies of others. Heery and Wagner (2002, online) note that these registries trace a lineage back to “shared data dictionaries and the registration process encouraged by the ISO/IEC 11179 community.”

The inaugural Dublin Core Series Workshop took place in Dublin, Ohio, in 1995. The initiative’s first deliverable was the Dublin Core Metadata Element Set (DCMES), a vocabulary for describing 15 “core” information properties, such as “Description,” “Creator,” and “Date.” The DCMES functions like a library card catalog system for Web resources. “Simple Dublin Core” uses only the canonical elements defined in the Element Set, whereas “Qualified Dublin Core” allows the addition of further data-defining qualifiers. A number of people in the DCMI are active in the W3C; indeed, Dublin Core is the application on which RDF was initially prototyped. The logical expressivity of the DCMES is particularly simple, consisting as it does merely in attribute:value pairs (with, moreover, possible attributes numbering only 15 so that it lacks the “universal expressive power” of Decker et al., [2000, p. 67]); consequentially, one might hesitate to call it an ontology at all.

WordNet

WordNet has been under development since 1985 at the Cognitive Science Laboratory of Princeton University (under the direction of the psychologist George Miller). The work has received considerable funding from government agencies interested in machine translation. WordNet is

a lexicon for the English language containing approximately 150,000 words organized into more than 115,000 “synsets” (sets of synonymous words or phrases). Words typically participate in several synsets. WordNet distinguishes between nouns, verbs, adjectives, and adverbs. The meaning of the synsets is further clarified with short defining glosses. Every synset is connected to other synsets via a number of transitive relations. For nouns, the relations include synonyms, hypernyms (Y is a hypernym of X if every X is a kind of Y), “hyponyms” (Y is a hyponym of X if every Y is a kind of X), “holonyms” (Y is a holonym of X if X is a part of Y), and “meronyms” (Y is a meronym of X if Y is a part of X). The hypernym and hyponym relationships are also applied to verbs; thus, both nouns and verbs are organized into subsumption hierarchies.

The WordNet project, which is freely downloadable (<http://wordnet.princeton.edu>), is widely used and has spawned a number of derivative projects. Global WordNet attempts to coordinate the production and linking of wordnets for all languages (www.globalwordnet.org). The eXtended WordNet project at the University of Texas at Dallas aims to improve WordNet by semantically parsing the glosses, thus making them available for automatic knowledge processing systems (<http://xwn.hlt.utdallas.edu>). FrameNet, developed by Charles Fillmore at the University of California at Berkeley, adds some assertions ascribing properties to entities listed in WordNet, thus bringing it closer to DL expressivity (www.icsi.berkeley.edu/framenet). SENSUS, developed at the University of Southern California’s Information Sciences Institute, is an extension and reorganization of WordNet.

WordNet is very simple as regards logical expressivity, as it consists almost entirely of a lexicon of words with natural language definitions (although it does enable formal reasoning over subsumption hierarchies organized on the few relations previously mentioned). It is also simple in that it cannot, in principle, distinguish between words themselves and the concepts they express (although synsets do, in effect, create semantic clusters—thus, for example, the word “tree” participates in one synset qua biological organism and another qua mathematical structure). In spite of its simplicity, WordNet is often used as a formal ontology. Some researchers have suggested that this widespread use is precisely because of WordNet’s simplicity (e.g., Sowa, 2004). Developers of other, more sophisticated ontologies have put considerable effort into mapping WordNet onto their systems (for instance, SUMO and OpenCyc, which are discussed in subsections of “Ontologies with First-Order Logic Expressivity [or Higher]”).

Ontologies with (Largely) DL Expressivity

Topic Maps

Topic Maps (Pepper & Moore, 2001) derive from work in library sciences and knowledge indexing (Bailey et al., 2005) and are standardized in ISO/IEC 13250. They represent data concerning “topics” (which are

named and organized into a subsumption hierarchy of topic types) by means of “associations,” which are relations of arbitrary arity (i.e., numbers of places) between topics, and “occurrences,” which are information resources relevant to a topic. Topic Maps can be understood, like RDF data, as directed graphs with labeled nodes and edges. However, they are more logically expressive because of the arbitrary complexity of association relations (in contrast to RDF’s restriction to binary relations). Trials of large-scale Topic Maps knowledge bases have largely taken place within organizations that have voluminous in-house data, such as the U.S. Office of Naval Intelligence, the U.S. Internal Revenue Service, and the Dutch Tax and Customs Administration (Breebaart, 2004); a less centralized project, closer to the spirit of the Semantic Web, is Kulturnett 3.0 ([www/kulturnett.no](http://www.kulturnett.no)), a Norwegian government portal for cultural information.

Unfortunately, in spite of their similarity, RDF and Topic Maps have developed separate user communities. Initial efforts toward integrating them are described by Lacher and Decker (2001); Pepper, Vitali, Garshol, Gessa, and Presutti (2005); and Garshol (2005, p. 24), although the last notes that fully merging the two technologies does not appear “desirable or possible.”

RDF Schema

RDF Schema (RDFS) (Brickley & Guha, 2003) extends RDF with some basic frame modeling primitives (de Bruijn, 2003). It allows one to declare classes and properties, populate classes with instances, and organize them into a subsumption hierarchy. It also allows adding range and domain constraints to properties and ascribing properties to individuals. All of these features render it a fragment of DL in terms of expressivity (Horrocks, Patel-Schneider, & van Harmelen, 2003).

One might ask about the relationship between RDFS and XML Schema. Does the Semantic Web really need both? If it does not, which would be preferable? Or might they be merged somehow? Unfortunately, the exact logical relationship and the extent of possible interoperability between XML Schema and RDFS are complicated and very difficult to understand. This difficulty is compounded by the polarization of their user communities along cultural lines—XML Schema is embraced by those who see themselves as “quick and dirty” real-world developers, whereas RDFS’s users identify with a perceived elegance and formal correctness usually only supported by academic research funding. In 1999, the two communities met in an attempt to work out their differences; the result was a document known as the “Cambridge Communiqué” (Swick & Thompson, 1999). The participants agreed that XML Schema and RDFS should not be merged (due to their different data models), but that certain atomic data types should be shared. The communiqué also suggested that the XML Schema specification should provide an extension mechanism to (for example) allow elements from other namespaces

to be included in its schema documents, possibly even including the RDF data model itself. Further details are given by Melnik (1999). Another attempt to bridge the XML-RDF gap in a common formal semantics can be found in Patel-Schneider and Siméon (2002).

RDFS is still too logically simple to express a great deal of what is said on the Web. Within its framework, it is possible to declare new classes and populate them with instances, but one cannot say anything further about these classes and instances; moreover, one cannot state axioms or inference rules concerning them (Delteil, Faron-Zucker, & Dieng, 2001). Thus, the problem with *defining* (in a machine-understandable way) the meaning of terms used in one's ontology language remains—ultimately, RDFS terms are still indexed only via namespaces whose further meaning is opaque.

DAML+OIL/OWL

DAML+OIL was initially created by merging the results of two projects: the American Defense Advanced Research Projects Agency (DARPA) Agent Markup Language (DAML) project (www.daml.org) and the largely European “Ontology Inference Layer” (OIL) project (Fensel, Horrocks, van Harmelen, McGuinness, & Patel-Schneider, 2001; Horrocks, Patel-Schneider, & van Harmelen, 2002). It is a particularly expressive DL (de Bruijn, 2003). Although it supports all XML datatypes, it does not support explicit representation of them and the making of assertions about them.

DAML+OIL has recently been reworked and renamed OWL (Web Ontology Language) (McGuinness & van Harmelen, 2004), which became a W3C Recommendation in February 2004. OWL goes beyond RDF and RDFS by providing additional vocabulary and a formal semantics. The additional vocabulary includes the ability to define classes in terms of logical relationships between other classes (such as intersection, union, complement, or enumerations of specific entities), the ability to state class cardinality (e.g., “Jones has 6 Kate Bush CDs”), equality (for both classes and individuals), and characteristics of properties (such as symmetry, transitivity, functionality, and inverse relationships between properties). A major extension over RDFS, though, is the ability to provide restrictions on how properties behave that are local to a class (e.g., defining a class “Australian Person,” all of whose members have “Australia” as the value for the property “country of residence”). This, in particular, results in a much more complicated logic (Horrocks et al., 2003).

OWL has three sublanguages, each of which is an extension of the preceding one. OWL Lite, which is designed to support thesauri and other taxonomies, is limited to providing a classification hierarchy with some very simple constraints on values. OWL DL is designed for the greatest possible logical expressiveness while retaining decidability and computational completeness (all conclusions true in an OWL DL

knowledge base are computable). OWL Full sacrifices computational completeness for maximal logical expressiveness. In practice, however, OWL Lite has proven to be very close to the considerably more expressive OWL DL in terms of complexity of implementation; developers have therefore tended to favor the latter. OWL Full is so expressive that it is considered too difficult to implement and, consequently, is hardly used. As a result, most tools exist for OWL DL.

The exact formal relationship between OWL and RDF (how the former “extends,” in Berners-Lee’s sense, the latter) is a somewhat delicate matter. OWL Full can be viewed as an extension of RDF, whereas OWL Lite and OWL DL can only be viewed as extensions of a restricted view of RDF. In other words, every OWL document is an RDF document, and every RDF document is an OWL Full document, but only some RDF documents are legal OWL Lite or OWL DL documents. Thus, considerable care has to be taken in migrating RDF documents to OWL (McGuinness & van Harmelen, 2004). An interesting perspective on current issues in OWL that arise from its original roots in DL, as well as an overview of the conflicting requirements that led to the development of three separate languages, is to be found in Horrocks et al. (2003).

A penetrating paper (Patel-Schneider & Fensel, 2002) has raised a further issue with layering OWL over RDF and RDFS—namely, that it is in fact subject to Russell’s Paradox, which, as noted earlier defeated Frege’s original vision of a univocal, unified, and deductively complete system of knowledge. This is because OWL is sufficiently expressive to allow users to define classes at will from sets of resources, but the syntax of RDF lacks the resources to block the creation of the class of all classes that do not contain themselves. Patel-Schneider and Fensel concede that the response to this point might be “Who cares?” on the grounds that such a situation will arise extremely rarely, if at all, “in the wild,” and Semantic Web developers should stick to practical problems (see, for instance, Berners-Lee, 1998). However, they counter this, asking how practical it is to have “a complete collapse of the logical formalism” (Patel-Schneider & Fensel, 2002, p. 22). The dispute makes intriguingly vivid the divide within the Semantic Web community between application-oriented pragmatism and formal correctness.

Patel-Schneider and Fensel (2002) offer a detailed analysis of four possible solutions to this problem: (1) writing some explicit rules to limit inferences from statements in RDFS to statements in OWL (a measure that would cripple the expressiveness of the latter in bizarre and difficult-to-anticipate ways); (2) defining new syntactic constructs in OWL (which would seem to defeat the purpose of the layer-cake model according to which each layer extends the layers below it); (3) defining some new semantics for OWL that overrides certain RDFS semantics (a move that would also seem to defeat the purpose of the layer-cake model); and (4) defining a divergent syntax and semantics for OWL (which would seem the most inelegant solution of all).

In summary then, OWL, although currently the flagship ontology of the W3C group, is not problem-free; a case in point is its complex verbosity. For instance, here is Horrocks et al.'s (2003, p. 17) OWL translation of the statement "A student is a person who is enrolled in at least one class":

```
<owl:Class rdf:ID="Student">
  <owl:intersectionOf rdf:parsetype="Collection">
    <owl:Class rdfs:about="Person" />
    <owl:Restriction>
      <owl:onProperty
        rdf:resource="enrolledIn" />
      <owl:minCardinality
        rdfs:datatype="xsd:Integer">
        </owl:minCardinality>
    </owl:Restriction>
  </owl:intersectionOf>
</owl:Class>
```

Consider also that layering such a statement on top of RDF will entail encoding it into stackable triples which will have to be held together somehow if the statement's coherence is not to be lost (Horrocks et al., 2003).

SHOE

SHOE (Simple HTML Ontology Extensions) is an early Semantic Web ontology platform that combines characteristics of both frame systems and predicate logics (Heflin & Hendler, 2000; Heflin et al., 2003). It enables the annotation of Web pages with separate machine-readable metadata, which can be embedded directly in HTML documents or used in XML documents. The latter strategy is preferred because it exposes the metadata to a broader range of software tools (Heflin et al., 2003).

SHOE addresses the problem of the distributed nature of Web resources by providing a system for extending ontologies, using tags to specify the ID and version number of ontologies extended by a given ontology, and making use of prefixes to specify from which ontology a given term comes. It has a formal semantics. SHOE's developers claim to minimize the problem of logical contradiction on the Web by designing the language as far as possible to avoid the representation of logical contradictions. Thus, it does not permit logical negation, retractions of assertions already made, single-valued relations, or the specification of disjointness among classes. Of course, this restricts SHOE's expressive powers considerably (for instance, rendering impossible claims such as "No CDs are vinyl records").

The SHOE approach has been criticized by Fensel et al. (2000, p. 363) for its strategy of adding further data to pages rather than "mak[ing] explicit the semantics of already available data." Not only is the latter

approach more elegant, but it also avoids maintenance problems caused by the presence of two sets of tags for the same data. Fensel et al. also claimed that SHOE's inferencing is too weak, barely outstripping basic database capabilities. However, it should be noted that these researchers are the developers of a rival system, Ontobroker, which, in contrast to SHOE's basic database-style inferencing, is more logic-based and infers and records additional knowledge on top of what it is given. Such proactive deduction has traditionally been known in AI circles as "forward inference." However, given the exponential growth of forwardly inferred knowledge once just a few simple inferencing rules are in place, one might ask how scalable this is for the whole Semantic Web (Fensel et al., 2000).

That SHOE might have been something of a trial Semantic Web application, now superseded, is suggested by the fact that its official Web site is no longer actively maintained and that two of its key developers (Hendler and Heflin) are now working on OWL.

Ontologies with First-Order Logic Expressivity (or Higher)

Cyc

As has been noted, the Cyc project (www.cyc.com) is the most ambitious formal ontology project yet attempted, with roots firmly planted in classical AI. It is the most ambitious in terms of size (over 600,000 categories), depth of knowledge (over two million axioms), and time devoted to it (over 700 person-years) (Sowa, 2004). Cyc began in 1984 as a project within the Microelectronics and Computer Technology Corporation (MCC), spinning off as an independent company, Cycorp, in 1994. In 2002, Cycorp released the first version (0.7.0) of OpenCyc (www.open.cyc.org), an open source version of (about 40 percent of) its ontology. In February 2005, version 0.9.0 was released (about six times the size of 0.7.0). There also exists a slightly larger version for researchers, named ResearchCyc. The Cyc ontology has its own purpose-built inference engine, which is bundled with OpenCyc and ResearchCyc.

Even more ambitiously, the Cyc project was originally envisaged to move from initial hand-coding of its knowledge base by "ontological engineers" (a process that has turned out to be very expensive—a recent public trial on a high school chemistry textbook produced an estimate of \$10,000 a page [Sowa, 2004]) to automated acquisition of knowledge by the system itself through reading and asking questions of untrained humans (Matuszek, Witbrock, Kahlert, Cabral, Schneider, Shah, et al., 2005). Building a natural language interface is another major plank of the project: Unlike WordNet, there is a separation, and required mapping, between terms representing concepts in the Cyc ontology and words in English or any other language.

The Cyc project has boldly addressed itself to the full-blown ontological problem of not just indexing terms but also attempting to describe

their meanings in machine-understandable terms. For example, its representation of the concept of a tree (`#Tree-ThePlant`, which is distinguished from `#Tree-PathSystem`) comes with axiomatic assertions (e.g., “A tree is largely made of wood”) and rules (e.g., “If a tree is cut down, then it will be destroyed”). This work is done in the purpose-built CycL language, which, although largely consisting of first-order predicate logic and set theory, has the expressivity of higher-order logic—allowing assertions about assertions, context logic (Cyc contains 6,000 explicitly represented contexts, known as “microtheories”), and some modal statements (although only temporal modality currently has inference support). An added complication is that the language allows assertions to be accorded not two but five different truth values: non-defeasibly true, defeasibly true, unknown, defeasibly false, and non-defeasibly false.

Because this project has gone on for so long and at such considerable expense, it has been the object of some criticism. For instance, Copeland (1997) criticized particular ontological choices that have been made with respect to the representation of substance, causation, and time. In an early but influential article, Smith (1991, pp. 252–253) argued that methodologically, Cyc’s developers had “left out the middle part,” accusing them of

an assumption that you can move directly from broad intuition to detailed proposal, with essentially no need for intermediate conceptual results ... from the generality of human knowledge to the intricacies of slot inheritance; from the full flowering of intelligence to particular kinds of controlled search.

(Interestingly, critics of the Semantic Web have raised a similar complaint.) At any rate, one may fairly say that the research challenge of producing a general purpose ontology with a dedicated team of researchers and considerable cash has been much more difficult to carry through than was envisioned at the project’s outset (for instance, in Lenat & Feigenbaum, 1991). This criticism is justifiable insofar as the time at which the system is to begin learning on its own continues to be pushed forward by the Cycorp itself.

What potential use does this ontology have for the Semantic Web? On one hand, Cycorp has made strenuous efforts to map other ontologies and databases into Cyc. Examples include FIPS (Federal Information Processing Standards), the Central Intelligence Agency (CIA) WorldFactbook (Reed & Lenat, 2002), and WordNet. The company also recently published a paper on automated OWL annotation of text documents (Witbrock et al., 2004); one significant challenge the authors noted is that the translation of terms in the text to terms in OWL proceeds via the CycL language, not all of which is translatable into OWL.

Unfortunately, Semantic Web developers outside the company have so far made little use of this ontology (but see Sicilia, Lytras, Rodriguez,

& Garcia-Barriocanal, 2006). Its system of categories is extremely complex, requiring philosophical training to understand (the majority of Cycorp's in-house ontologists possess philosophy Ph.D.s, and even they require six months to learn the system properly), and for a long time was poorly documented. Inferential tractability must also be a particular worry given the extreme expressivity of the CycL language. Ramachandran, Regan, and Goolsbey (2005) have recently explored to what extent this issue might be mitigated by "first-orderizing" sections of the knowledge base. Furthermore, because Cyc comprises such an integrated suite of applications (where the inference engine and other tools are custom-built and as such difficult to "unbundle"), it seems destined to remain a "located" rather than a "distributed" technology and committed to canonicalization at the expense of the decentralization fostered by notions of the Semantic Web vision. It has been the company's strongly held belief, however, (echoing Kant's rationalist skepticism that meaning might be chopped into units) that it is necessary to solve the problem of understanding basic, commonsense knowledge in an integrated, coherent manner, via one general-purpose application as a first step in building any other knowledge-based application, including the Semantic Web.

SUMO

SUMO (Suggested Upper Merged Ontology) was developed by Teknowledge Corporation (<http://ontology.teknowledge.com>) and first released in December 2000. Adam Pease, the current Technical Editor of the standard, has authored a number of publications presenting its key features (Niles & Pease, 2001; Pease & Niles, 2002). It can be browsed (www.ontologyportal.org), downloaded, and used freely. However, unlike the Cyc system, it does not include an inference engine.

SUMO's ontology language is (the full first-order logic expressivity) KIF and, as is the case with Cyc, the basic structure of its ontology is set-theoretic, including both axioms and rules. Its upper ontology consists of 1,000 concepts, defined using 4,000 assertions. A complete mapping from WordNet synsets to SUMO classes has been defined. There is also a Mid-Level Ontology (MILO) (Niles & Allan, 2004). An attempt has been made to sell SUMO to Semantic Web developers (Pease, Niles, & Li, 2002). Unfortunately this ontology is subject to many of the same concerns as Cyc—indeed, it is merely a smaller version of the same basic design—and has seen just as little external uptake.

SUO

SUO (Standard Upper Ontology) is a voluntary effort spearheaded by researchers concerned that, given the potential worldwide reach and importance of any real-world solution to the ontological problem, a viable, free, open source alternative to proprietary solutions should be presented. It largely consists of an Institute of Electrical and Electronics

Engineers (IEEE) working group (<http://suo.ieee.org>) and an e-mail list. A motion was passed in June 2003 to define an ontology registry into which members might enter ontologies that will be “related in a generalization/specialization hierarchy” (IEEE, Standard Upper Ontology Working Group, 2003, online). However, given the voluntary nature of participation in this forum, contributions have largely been confined to parties proffering their own already-worked-out ontologies as “starting documents” (for instance, OpenCyc, SUMO). Sustained independent development within the forum has not occurred: This seems to offer further confirmation that the Semantic Web’s most difficult challenge may be political.

The list of ontologies discussed here is representative rather than comprehensive. Other general-purpose upper ontologies include Philippe Martin’s (2003) MSO (Multi-Source Ontology), which was built by integrating a series of others onto a cleaned-up WordNet, and Barry Smith’s (2003a) BFO (Basic Formal Ontology). BFO claims to distinguish itself by being based not on set theory, but on mereology (a theory of wholes and their parts as opposed to a theory of classes and their members); however, applications based on it seem still to make some use of a traditional subsumption hierarchy (Dos Santos, Dhaen, Fielding, & Ceusters, 2004).

To sum up the last three sections, then, ontologies were meant to be the “silver bullet” that delivered a machine-understandable theory of meaning; however, none of the applications examined here seems to have succeeded. They appear to fall across a spectrum whereby terms used by the simpler systems are defined at most in the form of URIs pointing to unique namespaces (Dublin Core) or in natural language glosses (WordNet). Some kind of machine-understandable definitions of the meanings of terms is needed if more robust results are to be achieved. And what is needed for that? It would appear axiomatic assertions and inferential rules concerning all the terms in one’s ontology, at least when working within a deductivist inferencing paradigm. (As has already been mentioned, it is admittedly a big assumption to posit reliance on deductive inferencing, but other forms of automated reasoning have been slow to appear on the Semantic Web.) To be able to formulate axioms and rules, though, one is forced to build considerable logical expressivity into the ontology language. At this point, one encounters a raft of new problems: the complexity of one’s language, the need to determine its formal semantics, and the inferential tractability, scalability, and brittleness of applications built using it. What noble attempts do exist suffer from obscurity and lack of uptake by others.

It might be argued, however, that the real problem actually lies with the attempt to build one all-embracing, general-purpose ontology and that local, distributed ontologies are the solution, a consideration that brings us to the next section.

Domain Ontologies

“Domain ontology” is the term of art for ontologies that pertain to specific, integrated areas of knowledge (such as individual sciences, for instance physics or chemistry, or real-world subject areas such as film or adventure sports). A bewildering variety of domain ontologies, in a wide range of formats, exists at present.

The knowledge domain in which ontology development is currently most advanced (in both size and sophistication) is bioinformatics. Notable examples here include the Gene Ontology (www.geneontology.org), which is downloadable in XML or OWL (Ashburner, Ball, Blake, Botstein, Butler, Cherry, et al., 2000), and SNOMED (the Systematized Nomenclature of Medicine), developed by the College of American Pathologists, which attempts the complete capture of healthcare terminology (www.snomed.org) (Spackman, Campbell, & Côté, 1997). Both of these are closer to glossaries or controlled vocabularies than to ontologies proper, however. Although they do provide subsumption hierarchies, they are very simple (containing very few roles). The fact that they have been constructed largely by domain experts ignorant of the tradition of formal knowledge representation has resulted in criticism by philosophically trained ontologists for various alleged metaphysical and logical blunders. For instance, Smith, Williams, and Schulze-Kremer (2003) have claimed that the Gene Ontology’s “is-a” relationship is deeply ambiguous. As SNOMED has been under development for twenty years and contains 150,000 “records,” it is particularly unruly, and Ceusters, Smith, Kumar, and Dhaen (2004) have recently alleged that it contains numerous logical errors. It should be noted that Smith himself is developing an alternative biomedical ontology based on BFO, which has been mentioned earlier. His Leipzig National Center for Biomedical Ontology (<http://bioontology.org>) includes a library of 50 stackable modular ontologies designed to be merged and used together.

Another ontology domain in which work is proceeding apace is geographical and spatio-temporal reasoning (a subject area currently very well funded because of its potential military applications). This is an extremely difficult domain to ontologize because spatial information is necessarily mathematical and mathematical relationships pose a significant challenge to the set-theoretic framework in which ontologies are traditionally built. (This is so because set-theoretic relations represent only a tiny subset of mathematical functions.) Nevertheless, the problem is currently being addressed by a large project at the University of Bremen (Bateman & Farrar, 2004), as well as by the nonprofit Open Geospatial Consortium (www.opengeospatial.org). The latter seems currently mainly to provide standards for Web services, but has sponsored the development of a Geographical Markup Language (GML) (<http://opengis.net/gml>).

Ontology Libraries and Related Resources

A number of ontology repositories or libraries exist. The original, and now relatively dated, site is the DAML Ontology Library (www.daml.org/ontologies). More recent offerings include SemWebCentral (www.semwebcentral.org) and SchemaWeb (www.schemaweb.info).

Resources to search across domain ontologies are, of course, a natural development, and are beginning to emerge. These include search engines, one notable example of which is Swoogle, a Google-like search engine for the Semantic Web developed at the University of Maryland, Baltimore County (UMBC) (<http://swoogle.umbc.edu>). This tool searches and indexes the world's metadata using both character string and URI references. It also seeks to provide "ontology rank," an equivalent to Google's Web page ranking service, although it is worth noting that the top-ranked ontologies as of 2004 were the W3C's own specification documents for RDF, RDFS, and DAML+OIL (Ding, Finin, Joshi, Pan, Cost, Peng, et al., 2004). Resources to explore Semantic Web ontologies also include browsers, of which an oft-cited example is Ontaria (www.w3c.org/2004/ontaria), which site is currently down for overhaul (as of December 2005).

Whether local, distributed ontologies might prove more useful than their general, all-purpose cousins for the Semantic Web it is arguably too early to tell. Even the most highly developed domains, bioinformatics and geographical and spatio-temporal reasoning, ontologies still require considerable development. Ontologies for most other domains are tiny, with numbers of categories in the tens rather than the hundreds or thousands. Burton-Jones, Storey, Sugumaran, and Ahluwalia (2005, p. 98) studied the DAML library and claimed that both the size and the quality of publicly available ontologies were currently questionable: "For the average ontology, 18 percent of its syntax is incorrect, 37 percent of its terms are uninterpretable, 22 percent of its terms are polysemous and 18 percent of its statements are irrelevant." A further question concerns how local distributed ontologies are going to interoperate, given that Semantic Web information integration operations are meant to be performable not merely within "domain islands," but across the Web as a whole. Will a further ontology be required to perform ontology integration? How will *its* terms be defined?

Conclusion

The Challenges Revisited

Throughout this chapter, a number of significant challenges have been identified with regard to the ontological dimension of the original vision of the Semantic Web. The first is the trade-off in ontology languages between logical expressivity and inferential tractability. Is there a way of resolving this tension? The various languages of DL expressivity might seem to provide a happy medium between, on the one hand,

extreme expressivity and inferential intractability and, on the other, extreme inferential tractability and inexpressivity by providing the maximum expressivity obtainable while retaining decidability. But still, in contrast to the original Semantic Web vision of rendering machine-readable the entire Web's human-readable knowledge, there is an enormous amount that cannot be said in these languages (to give just one example, "Bill Evans recorded exactly one album with Miles Davis").

It is sometimes claimed that ever more powerful computer hardware will solve this problem, rendering tractable tomorrow inferential problems that are intractable today. This is not correct, however, because inferential tractability decreases exponentially with such simple matters as the number of variables in a question, for which no principled upper limit has yet been set on the Semantic Web. In a position paper written in the early days of the Semantic Web, Berners-Lee (1998, online) blithely stated that the Semantic Web will not "require every application to use expressions of arbitrary complexity" but gave no further details regarding what might constitute a principled means of determining which applications would, and would not, use such expressions; whether the Semantic Web itself might be forced to divide this labor when serving up knowledge; and, if so, how. In the same paper, Berners-Lee also attempted to dodge the inferential tractability issue by claiming that the Semantic Web need not supply "proof generation," only "proof validation" (an example of such being the way access control is provided by Web sites). Once again, however, some details would be helpful—regarding, for example, how this might work for information integration over a number of independent sources (so as to answer, for instance, Berners-Lee et al.'s [2001] question: Which medical specialist would be able to see the ailing mother at a mutually convenient time?)

The second significant challenge is the trade-off between decentralization and canonicalization in the definitions of ontological terms. (This includes both terms for general concepts, such as "CD," and terms for individuals, such as "Kate Bush," which may well require differing solutions.) The invention of URIs qua unique character strings goes only the first, easy step toward solving the problem of uniquely identifying resources. The rest of the issue concerns how humans are to agree on a particular character string/resource alignment. (In an interesting twist, it has even been suggested that resources are most effectively picked out not by such pure names, but via uniquely identifying—"discriminant"—descriptions [Guha & McCool, 2003].) In an echo of Smith's complaint back in 1991 about Cyc's insouciance regarding the hard methodological middle ground, Shirky (2003, online) has observed that the modus operandi of Semantic Web development is to "take some well-known problem. Next, misconstrue it so that the hard part is made to seem trivial and the trivial part hard. Finally, congratulate yourself for solving the trivial part." The combined effect of these challenges has generated significant skepticism in some quarters regarding the Semantic Web (for

a particularly pungent critique that draws morals from library science, see Brooks, 2002).

Possible Alternatives to the Semantic Web

If the goals of the Semantic Web cannot be achieved by the current W3C development framework, might they be achieved by other means? And if not, what is the Web's future? Some recent Web developments are worth mentioning insofar as they seem to partake of the giddy momentum of the original technology. The first development is *tagging*. Tags are labels users voluntarily add to the Web. Ontologically speaking, the practice is entirely uncontrolled—no categories are prepared or agreed upon in advance. (Thus a given CD might be labeled “boring,” “amazing,” “Mike_likes_this,” “driving_music,” and “songs_about_fish.”) The practice began as a means of labeling Web pages with words or phrases meaningful to oneself so that one might rediscover them quickly; it has spread to embrace a number of other, much more public uses as a variety of Web sites has emerged to serve as tag clearinghouses. Examples of such sites include del.icio.us for the sharing of tagged bookmarks to Web sites (<http://del.icio.us>) and Flickr for tagged photographs (www.flickr.com). Unsurprisingly, developers have begun to develop search engines for this new slew of metadata (e.g., Technorati: www.technorati.com).

Tagging is said to produce not a *taxonomy* (in the sense of a mark-up according to a pre-given structure of categories) but the newly coined term *folksonomy* (Weinberger, 2005). In spite of the “feral” source of tags, it has been argued that, at the level of the entire Web, semantic patterns emerge among them, thus lessening the impact of individual idiosyncrasy, and that, “by forgoing formal classification, tags enable a huge amount of user-produced organizational value, at vanishingly small cost” (Shirky, 2005, online). This new development is interesting from the perspective of the Cartesian-versus-Peircean philosophical debate with respect to meaning. We saw that the Cartesian perspective holds that the meaning of a sign consists in the idea its user intends to convey by using it. In Semantic Web development terms, this model naturally generates the attempt to anticipate and define every possible aspect of a term's future meaning through explicit axioms and rules (as is epitomized, for instance, in the Cyc project). According to the Peircean perspective, on the other hand, the meaning of a sign consists merely in the way it continues to be used. Consider, for example, the popular pairing between the terms “blonde” and “joke.” From the Cartesian perspective, for the Semantic Web to work, these terms will need to be defined so that the meaning of their pairing might be anticipated, an obviously Herculean task. The Peircean perspective allows that the phrase might be a new development in the meaning of both terms—somehow consisting in the new uses of the phrase themselves.

A further runaway development in the recent Web is *RSS autodiscovery*. Via software of simplicity reminiscent of early HTML, this technology “syndicates” Web sites (frequently Weblogs) by providing summaries of their content, links to the full version, and other metadata in an XML file called an RSS (Really Simple Syndication/RDF Site Summary) feed. Content is filtered for individual receivers using keywords (the choice of which once again is wholly personal and idiosyncratic). There has been some speculation among Web developers that such advances might result in a “lowercase semantic web,” which will implement at least some of the original founders’ goals while bypassing (or, less charitably, precisely by bypassing) the formal standards they have labored so hard to create. (For an attempt to merge the two, see Cayzer [2004].)

This review makes evident that inferencing capability is inseparable from machine-understandable meaning; it is currently difficult to envisage how coherent inferencing rules might be built on such a turbulent, ambiguous, and amateur base as tags and RSS feeds. In direct contrast to formal ontologies, such “do-it-yourself” metadata seem to demonstrate an extreme version of the decentralization at the expense of the canonicalization plank of the Semantic Web vision. Still, these criticisms might equally well be leveled at Google’s deployment of its (in fact spectacularly successful) page-rank algorithm across the turbulent, ambiguous, and amateur World Wide Web. Shirky (2005) has suggested that, just as it was Google’s genius to realize how to leverage vast quantities of hyperlinks to create a meaning (namely, the significance of a given Web site) that the authors of hyperlinks never intended them to have, something analogous and as-yet-unanticipated, substituting order of magnitude of data for authorial intention, will surely happen to generate the requisite metadata.

An even more subversive question worth at least some thought is: “Is the Web really the future of the Internet?” Recent discussions on the popular U.K. information technology news and gossip site *The Register* have speculated that it might not be, suggesting that the Web is “much less important, compared to other Internet services, than the giant Web search engines like to think” and making reference to a so-called “Asian way of using Internet,” which focuses rather on games, Internet chat rooms, and Web radio and Television (Google can take the Web, yawn readers, 2005, online).

The Semantic Web vision is audacious and tempting. However, attempts to realize it may mean pursuing a “machine-understandable meaning at the end of the rainbow.” The more one attempts to build a formal system that explicitly and antecedently states what terms used on the Web will mean, the more it appears that the meaning of the terms in the formal system itself elude one’s grasp. It can easily appear that one needs an ontology behind one’s ontology in order to determine *its* meaning—an issue philosophers (the original ontologists) have been struggling with for centuries. The only alternative would seem to be to

explore with Peirce the surprising idea that machine-understandable data about the meaning of Web resources already exist precisely in the form of those resources themselves, if we can only find the means to leverage them.

Acknowledgments

For feedback and discussion that considerably improved this chapter, I am indebted to Ajeet Parhar, Lars Marius Garshol, Sally-Jo Cunningham, and the anonymous reviewers.

References

- Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., et al. (2000). Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25, 25–29.
- Bailey, J., Bry, F., Furche, T., & Schaffert, S. (2005). *Web and Semantic Web query languages: A survey*. Retrieved February 26, 2006, from www.pms.ifi.lmu.de/publikationen/PMS-FB/PMS-FB-2005-14.pdf
- Bateman, J., & Farrar, S. (2004). Towards a generic foundation for spatial ontology. *Proceedings of the Third International Conference on Formal Ontology in Information Systems*, 237–248.
- Beckett, D. (2003). *TURTLE: Terse RDF triple language*. Retrieved February 26, 2006, from <http://ilrt.org/discovery/2003/11/ntriplesplus>
- Beckett, D. (2004). *RDF/XML syntax specification (Revised)*. Retrieved February 26, 2006, from www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210
- Berners-Lee, T. (1998). *What the Semantic Web can represent*. Retrieved February 26, 2006, from www.w3.org/DesignIssues/RDFnot.html
- Berners-Lee, T. (2000, December). *Semantic Web*. Paper presented at XML 2000, Washington, DC. Retrieved March 22, 2006, from www.w3.org/2000/Talks/1206-xml2k-tbl/slide10-0.html
- Berners-Lee, T. (2001). *Notation 3*. Retrieved February 26, 2006, from www.w3.org/DesignIssues/Notation3
- Berners-Lee, T. (2003). Foreword. In D. Fensel (Ed.), *Spinning the Semantic Web: Bringing the World Wide Web to its full potential* (pp. xi–xxiii). Cambridge, MA: MIT Press.
- Berners-Lee, T. (2005). *Primer: Getting into RDF and Semantic Web using N3*. Retrieved February 26, 2006, from www.w3.org/2000/10/swap/Primer.html
- Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The Semantic Web. *Scientific American*, 284(5), 34–43.
- Bollobas, B. (2002). *Modern graph theory* (2nd ed.). Berlin: Springer.
- Borgida, A. (1996). On the relative expressive power of description logics and predicate logics. *Artificial Intelligence*, 82, 353–367.
- Brachman, R. (1978). *Structured inheritance networks* (Technical Report No. 3742). Cambridge, MA: Bolt Beranek & Newman.
- Brachman, R., & Levesque, H. (2004). *Knowledge representation and reasoning*. Boston: Morgan Kaufmann.

- Brachman, R., & Schmolze, J. (1985). An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9, 171–216.
- Bray, T. (2001). *What is RDF?* Retrieved February 26, 2006, from www.xml.com/pub/a/2001/01/24/rdf.html?page=2
- Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E., & Yergeau, F. (2000). *Extensible Markup Language (XML) 1.0* (Third Edition). Retrieved February 26, 2006, from www.w3.org/TR/REC-xml
- Breebaart, M. (2004). *Using topic maps and XML to implement a closed-loop search service for the Dutch Tax and Customs Administration website*. Retrieved February 26, 2006, from www.idealliance.org/papers/dx_xml04/papers/04-01-03/04-01-03.html
- Brickley, D., & Guha, R. V. (2003). *RDF vocabulary description language 1.0: RDF schema*. Retrieved February 26, 2006, from www.w3.org/TR/rdf-schema
- Brooks, T. A. (2002). The Semantic Web, universalist ambition and some lessons from librarianship. *Information Research*, 7(4). Retrieved March 19, 2006, from <http://informationr.net/ir/7-4/paper136.html>
- Buchanan, B. G., & Shortliffe, E. H. (1984). *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*. Reading, MA: Addison-Wesley.
- Burton-Jones, A., Storey, V. C., Sugumaran, V., & Ahluwalia, P. (2005). A semiotic metrics suite for assessing the quality of ontologies. *Data and Knowledge Engineering*, 55, 84–102.
- Bush, V. (1945, July). As we may think. *Atlantic Monthly*, 176(1), 101–108.
- Cayzer, S. (2004). Semantic blogging and decentralized knowledge management. *Communications of the ACM*, 47(12), 47–52.
- Ceusters, W., Smith, B., Kumar, A., & Dhaen, C. (2004). Ontology-based error detection in SNOMED-CT. *Proceedings of the 11th World Congress on Medical Informatics*, 482–486.
- Chisholm, R. (1996). *A realistic theory of categories: An essay on ontology*. Cambridge, UK: Cambridge University Press.
- Clark, J., & Makoto, M. (2001). *RELAX NG specification*. Retrieved December 3, 2001, from www.oasis-open.org/committees/relax-ng/spec-20011203.html
- Clark, P., & Porter, B. (2001). *KM-The Knowledge Machine: Users manual and situations manual*. Retrieved February 26, 2006, from www.cs.utexas.edu/users/mfkb/RKF/km.html
- Copeland, J. (1997). CYC: A case study in ontological engineering. *Electronic Journal of Philosophy*, 5(6). Retrieved March 19, 2006, from <http://ejap.louisiana.edu/EJAP/1997.spring/copeland976.2.html>
- Cost, R. S., Finin, T., Joshi, A., Peng, Y., Nicholas, C., Soboroff, I., et al. (2002). ITTalks: A case study in the Semantic Web and DAML+OIL. *IEEE Intelligent Systems*, 17(1), 40–47.
- Cruse, D. A. (2006). *Meaning in language: An introduction to semantics and pragmatics*. Oxford, UK: Oxford University Press.
- de Bruijn, J. (2003). *Using ontologies: Enabling knowledge sharing and reuse on the Semantic Web* (DERI Technical Report, DERI-2003-10-29). Innsbruck, Austria: Digital Enterprise Research Institute. Retrieved February 26, 2006, from www.deri.org/fileadmin/documents/DERI-TR-2003-10-29.pdf

- Decker, S., van Harmelen, F., Broekstra, J., Erdmann, M., Fensel, D., Horrocks, I., et al. (2000). The Semantic Web: On the respective roles of XML and RDF. *IEEE Internet Computing*, 4(5), 63–74.
- Delteil, A., Faron-Zucker, C., & Dieng, R. (2001). Extension of RDFS based on the CG formalisms. *Proceedings of the 9th International Conference on Conceptual Structures*, 275–289.
- Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., et al. (2003). SemTag and Seeker: Bootstrapping the Semantic Web via automated semantic annotation. *Proceedings of the 12th International World Wide Web Conference*. Retrieved February 26, 2006, from www.conf.ecs.soton.ac.uk/archive/00000508/01/p831-dill.html
- Ding, L., Finin, T., Joshi, A., Pan, R., Cost, R. S., Peng, Y., et al. (2004). Swoogle: A search and metadata engine for the Semantic Web. *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management*, 652–659.
- Dos Santos, M., Dhaen, C., Fielding, M., & Ceusters, W. (2004). Philosophical scrutiny for run-time support of application ontology development. *International Conference on Formal Ontology and Information Systems*, 342–352.
- Euzenat, J. (2001). Towards a Principled Approach to Semantic Interoperability. *Proceedings of the International Joint Conference on Artificial Intelligence Workshop on Ontologies and Information Sharing*, 19–25. Retrieved March 28, 2006, from <http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-47>
- Fallside, D. (2001). *XML schema part 0: Primer*. Retrieved February 26, 2006, from www.w3.org/TR/2001/REC-xmlschema-0-20010502
- Fallside, D., & Walmsley, P. (2004). *XML schema part 0: Primer* (2nd ed.). Retrieved February 26, 2006, from www.w3.org/TR/xmlschema-0
- Fensel, D., Angele, J., Decker, S., Erdmann, M., Schnurr, H., Studer, R., et al. (2000). Lessons learned from applying AI to the Web. *International Journal of Cooperative Information Systems*, 9(4), 361–382.
- Fensel, D., Horrocks, I., van Harmelen, F., McGuinness, D., & Patel-Schneider, P. (2001). OIL: An ontology infrastructure for the Semantic Web. *IEEE Intelligent Systems*, 16(2), 38–45.
- Frege, G. (1970). Begriffsschrift: A formula language, modeled upon that of arithmetic, for pure thought. In J. Van Heijenoort (Comp.), *Frege and Gödel: Two fundamental texts in mathematical logic* (pp. 1–82). Cambridge, MA: Harvard University Press. (Original work published 1879.)
- Garshol, L. M. (2005). *Living with topic maps and RDF*. Retrieved February 26, 2006, from www.idealliance.org/papers/dx_xml03/papers/02-03-06/02-03-06.pdf
- Genesereth, M., & Fikes, R. (1992). *KIF: Knowledge Interchange Format version 3.0 reference manual*. Stanford, CA: Stanford University, Knowledge Systems AI Laboratory.
- Goble, C., & de Roure, D. (2002). The Grid: An application of the Semantic Web. *SIGMOD Record*, 31(4), 65–70.
- Google can take the Web, yawn readers. (2005, May 18). *The Register*. Retrieved February 26, 2006, from www.theregister.co.uk/2005/05/18/google_domination_letters
- Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5, 199–220.
- Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge-sharing. *International Journal of Human-Computer Studies*, 43, 907–928.

- Guha, R., & McCool, R. (2003). TAP: A Semantic Web platform. *Journal of Network Computing*, 42, 557–577.
- Haarslev, V., & Moeller, R. (2003). Racer: A core inference engine for the Semantic Web. *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools*, 27–36. Retrieved February 26, 2006, from www.racer-systems.com/technology/contributions/2003/HaMo03e.pdf
- Halevy, A. Y., Ives, Z. G., Mork, P., & Tatarinov, I. (2003). Piazza: Data-management infrastructure for Semantic Web applications. *Proceedings of the 12th International World Wide Web Conference*, 556–567.
- Harré, R., & Krausz, M. (1996). *Varieties of relativism*. Oxford, UK: Blackwell.
- Heery, R., & Wagner, H. (2002). A metadata registry for the Semantic Web. *D-Lib Magazine*, 8(5). Retrieved March 19, 2006, from www.dlib.org/dlib/may02/wagner/05wagner.html
- Heflin, J., & Hendler, J. (2000). Dynamic ontologies on the Web. *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 443–449.
- Heflin, J., Hendler, J., & Luke, S. (2003). SHOE: A blueprint for the Semantic Web. In D. Fensel, J. Hendler, H. Lieberman, & W. Wahlster (Eds.), *Spinning the Semantic Web: Bringing the World Wide Web to its full potential* (pp. 29–63). Cambridge, MA: MIT Press.
- Heflin, J., & Pan, Z. (2004). A model theoretic semantics for ontology versioning. *Proceedings of the Third International Semantic Web Conference*, 62–76.
- Hendler, J. (2001). Agents and the Semantic Web. *IEEE Intelligent Systems*, 16(2), 30–37.
- Hendler, J., Berners-Lee, T., & Miller, E. (2002). Integrating applications on the Semantic Web. *Journal of the Institute of Electrical Engineers of Japan*, 122, 676–680.
- Horrocks, I., Patel-Schneider, P., & van Harmelen, F. (2002). Reviewing the design of DAML+OIL: An ontology language for the Semantic Web. *Proceedings of the 18th National Conference on Artificial Intelligence*, 792–797.
- Horrocks, I., Patel-Schneider, P., & van Harmelen, F. (2003). From SHIQ and RDF to OWL: The making of a Web ontology language. *Journal of Web Semantics*, 1, 7–26.
- Horrocks, I., & Sattler, U. (2001). Ontology reasoning in the SHOQ(D) description logic. *Proceedings of the International Joint Conference on Artificial Intelligence Workshop on Ontologies and Information Sharing*, 199–204.
- Horrocks, I., Sattler, U., & Tobies, S. (2000a). Practical reasoning for very expressive description logics. *Logic Journal of the IGPL*, 8(3), 239–264.
- Horrocks, I., Sattler, U., & Tobies, S. (2000b). Reasoning with individuals for the description logic SHIQ. *Proceedings of the 17th International Conference on Automated Deduction*, 482–496.
- IEEE. Standard Upper Ontology Working Group. (2003). *Cumulative resolutions*. Retrieved March 30, 2006, from <http://suo.ieee.org/SUO/resolutions.html>
- Kant, I. (1998). *Critique of pure reason* (P. Guyer & A. Wood, Trans.). Cambridge, UK: Cambridge University Press.
- Kirk, R. (1999). *Relativism and reality: A contemporary introduction*. London: Routledge.
- Klein, M., & Bernstein, A. (2004). Towards high-precision service retrieval. *IEEE Internet Computing*, 8, 30–36.
- Lacher, M., & Decker, S. (2001). On the integration of topic maps and RDF data. *Extreme Markup Languages 2001: Proceedings*. Retrieved February 26, 2006, from www.mulberrytech.com/Extreme/Proceedings/html/2001/Lacher01/EML2001Lacher01.html

- Legg, C. (2005). The meaning of meaning-fallibilism. *Axiomathes*, 15, 293–318.
- Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11), 32–59.
- Lenat, D. B., & Feigenbaum, E. (1991). On the thresholds of knowledge. *Artificial Intelligence*, 47, 185–250.
- Lenat, D. B., & Guha, R. V. (1990). *Building large knowledge-based systems: Representation and inference in the Cyc project*. Reading, MA: Addison Wesley.
- Lenat, D. B., Guha, R. V., Pittman, K., Pratt, D., & Shepherd, M. (1990). Cyc: Toward programs with common sense. *Communications of the ACM*, 33(8), 30–49.
- Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., & Lederberg, J. (1980). *Applications of artificial intelligence for organic chemistry: The DENDRAL project*. New York: McGraw-Hill.
- Lowe, E. J. (1997). Ontological categories and natural kinds. *Philosophical Papers*, 26, 29–46.
- Lowe, E. J. (1998). *The possibility of metaphysics: Substance, identity, and time*. Oxford, UK: Oxford University Press.
- Maedche, A., Motik, B., & Stojanovic, L. (2003). Managing multiple and distributed ontologies on the Semantic Web. *VLDB Journal*, 12, 286–302.
- Martin P. (2003). Correction and extension of WordNet 1.7. *Proceedings of the 11th International Conference on Conceptual Structures*, 160–173.
- Matuszek, C., Witbrock, M., Kahlert, R., Cabral, J., Schneider, D., Shah, P., et al. (2005). Searching for common sense: Populating Cyc from the Web. *Proceedings of the Twentieth National Conference on Artificial Intelligence*. Retrieved February 26, 2006, from www.cyc.com/doc/white_papers/AAA1051MatuszekC.pdf
- McCarthy, J. (1995). What has AI in common with philosophy? *Proceedings of the 14th International Joint Conference on AI*, 2041–2044. Retrieved March 19, 2006, from www-formal.stanford.edu/jmc/aiphil/aiphil.html
- McCool, R., Fikes, R., & Guha, R. (2003). *Semantic issues in Web-scale knowledge aggregation* (KSL 03-15). Stanford, CA: Stanford University, Knowledge Systems AI Laboratory. Retrieved February 26, 2006, from www-ksl.stanford.edu/KSL_Abstracts/KSL-03-15.html
- McGuinness, D. (2003). Ontologies come of age. In D. Fensel, J. Hendler, H. Lieberman, & W. Wahlster (Eds.), *Spinning the Semantic Web: Bringing the World Wide Web to its full potential* (pp. 171–194). Cambridge, MA: MIT Press.
- McGuinness, D., & van Harmelen, F. (2004). *OWL Web ontology language: Overview*. Retrieved February 26, 2006, from www.w3.org/TR/owl-features
- McIlraith, S., Son, T. C., & Zeng, H. (2001). Semantic Web services. *IEEE Intelligent Systems*, 16, 46–53.
- Melnik, S. (1999). *Bridging the gap between RDF and XML*. Retrieved February 26, 2006, from www-db.stanford.edu/~melnik/rdf/fusion.html
- Minsky, M. (1975). A framework for representing knowledge. In P. Winston (Ed.), *The Psychology of Computer Vision* (pp. 211–277). New York: McGraw-Hill.
- Niles, I., & Pease, A. (2001). Towards a standard upper ontology. *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems*. Retrieved March 19, 2006, from <http://projects.teknowledge.com/HPKB/Publications/FOIS.pdf>

- Niles, I., & Allan, T. (2004). The MILO: A general-purpose, mid-level ontology. *Proceedings of the International Conference on Information and Knowledge Engineering*, 5–19.
- Noy, N., & Klein, M. (2004). Ontology evolution: Not the same as schema evolution. *Knowledge and Information Systems*, 6, 428–440.
- Noy, N., & Musen, M. (2004). Ontology versioning in an ontology-management framework. *IEEE Intelligent Systems*, 19(4), 6–13.
- Patel-Schneider, P. F., & Fensel, D. (2002). Layering the Semantic Web: Problems and directions. *Proceedings of the First International Semantic Web Conference*, 16–29.
- Patel-Schneider, P. F., & Siméon, J. (2002). The yin/yang Web: XML syntax and RDF semantics. *Proceedings of the 11th international conference on World Wide Web*, 443–453.
- Pease, A., & Niles, I. (2002). IEEE standard upper ontology: A progress report. *Knowledge Engineering Review*, 17, 65–70.
- Pease, A., Niles, I., & Li, J. (2002). The suggested upper merged ontology: A large ontology for the Semantic Web and its applications. *Working Notes of the AAAI-2002 Workshop on Ontologies and the Semantic Web*. Retrieved March 19, 2006, from <http://home.earthlink.net/~adampease/professional/Pease.ps>
- Peirce, C. S. (1931–1958). *Collected papers* (C. Hartshorne, P. Weiss, & A. Burks, Eds.). Cambridge, MA: Harvard University Press.
- Peirce, C. S. (1940). *Philosophy of Peirce: Selected writings* (J. Buchler, Ed.). London: Routledge and Kegan Paul.
- Pepper, S., & Moore, G. (2001). *XML topic maps (XTM) 1.0*. Retrieved February 26, 2006, from www.topicmaps.org/xtm/1.0
- Pepper, S., Vitali, F., Garshol, L., Gessa, N., & Presutti, V. (2005, March). *A survey of RDF/topic maps interoperability proposals*. Retrieved February 26, 2006, from www.w3.org/TR/2005/WD-rdfm-survey-20050329
- Quillian, M. R. (1967). Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, 12, 410–430.
- Quine, W. V. (1953). On what there is. *From a logical point of view: 9 logico-philosophical essays* (pp. 1–19). Cambridge MA: Harvard University Press.
- Ramachandran, D., Reagan, P., & Goolsbey, K. (2005). First-orderized ResearchCyc: Expressivity and efficiency in a common-sense ontology. *Papers from the AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications*. Retrieved March 19, 2006, from www.cyc.com/doc/white_papers/folification.pdf
- Reed, S., & Lenat, D. (2002). Mapping ontologies into Cyc. *AAAI 2002 Conference Workshop on Ontologies for the Semantic Web*. Retrieved March 19, 2006, from www.cyc.com/doc/white_papers/mapping-ontologies-into-cyc_v31.pdf
- Roesner, D., Kunze, M., & Kroetzsch, S. (2005). *Transforming and enriching documents for the Semantic Web*. Retrieved February 26, 2006, from http://arxiv.org/PS_cache/cs/pdf/0501/0501096.pdf
- Rorty, R. (1990). *Objectivity, relativism, and truth: Philosophical papers, Vol. 1*. Cambridge, UK: Cambridge University Press.
- Rosenberg, G., & Salomaa, A. (1994). *Cornerstones of undecidability*. New York: Prentice Hall.
- Rozenfeld, S. (2001). *Using the ISSN (International Serial Standard Number) as URN (Uniform Resource Names) within an ISSN-URN namespace* (RFC 3044). Retrieved February 26, 2006, from <ftp://ftp.isi.edu/in-notes/rfc3044.txt>

- Schlick, M. (1936). Meaning and verification. *Philosophical Review*, 45, 339–369.
- Schmidt-Schauss, M. (1989). Subsumption in KL-ONE is undecidable. *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, 421–431.
- Shah, U., Finin, T., Joshi, A., Cost, R. S., & Mayfield, J. (2002). Information retrieval on the Semantic Web. *Proceedings of the 11th International Conference on Information and Knowledge Management*, 461–468.
- Shirky, C. (2003). *The Semantic Web, syllogism, and worldview*. Retrieved February 26, 2006, from www.shirky.com/writings/semantic_syllogism.html
- Shirky, C. (2005). *Ontology is over-rated: Categories, links, and tags*. Retrieved February 26, 2006, from www.shirky.com/writings/ontology_ouerrated.html
- Sicilia, M. A., Lytras, M., Rodriguez, E., & Garcia-Barriocanal, E. (2006). Integrating descriptions of knowledge management learning activities into large ontological structures: A case study. *Data and Knowledge Engineering*, 57, 111–121.
- Sloman, A. (1995). A philosophical encounter: An interactive presentation of some of the key philosophical problems in AI and AI problems in philosophy. *Proceedings of the 14th International Joint Conference on AI*. Retrieved March 19, 2006, from <http://www.cs.bham.ac.uk/research/cogaff/Sloman.ijcai95.pdf>
- Smith, B. (2003a). *BFO/MedO: Basic formal ontology and medical ontology. Draft 0.0006*. Retrieved February 26, 2006, from <http://ontology.buffalo.edu/bfo/BFO.htm>
- Smith, B. (2003b). Ontology. In L. Floridi (Ed.), *Blackwell guide to the philosophy of computing and information* (pp. 155–166). Oxford, UK: Blackwell.
- Smith, B., & Welty, C. (2001). Ontology: Towards a new synthesis. *Proceedings of the International Conference on Formal Ontology in Information Systems*, iii–x.
- Smith, B., Williams, J., & Schulze-Kremer, S. (2003). The ontology of the gene ontology. *Proceedings of AMIA Symposium*, 609–613.
- Smith, B. C. (1991). The owl and the electric encyclopedia. *Artificial Intelligence*, 47, 251–288.
- Smith, R. (2004). Aristotle's logic. *Stanford encyclopedia of philosophy*. Stanford, CA: Stanford University. Retrieved February 26, 2006, from <http://plato.stanford.edu/entries/aristotle-logic>
- Sowa, J. (1999). *Knowledge representation: Logical, philosophical, and computational foundations*. Pacific Grove, CA: Brooks Cole.
- Sowa, J. (2000). Ontology, metadata, and semiotics. In B. Ganter & G. W. Mineau (Eds.), *Conceptual structures: Logical, linguistic, and computational issues* (Lecture Notes in Computer Science, 1867; pp. 55–81). Berlin: Springer.
- Sowa, J. (2001). *Signs, processes, and language games*. Retrieved February 26, 2006, from www.jfsowa.com/pubs/signproc.htm
- Sowa, J. (2004). *The challenge of knowledge soup*. Retrieved February 26, 2006, from www.jfsowa.com/pubs/challenge.pdf
- Sowa, J., & Majumdar, A. (2003). Analogical reasoning. In A. de Moor, W. Lex, & B. Ganter (Eds.), *Conceptual structures for knowledge creation and communication* (Lecture Notes in Computer Science, 2746; pp. 16–36). Berlin: Springer.
- Spackman, K., Campbell, K., & Côté, R. (1997). SNOMED RT: A reference terminology for health care. *Proceedings American Medical Informatics Association, Annual Fall Symposium*, 640–644.

- Staab, S., Maedche, A., & Handschuh, S. (2001). An annotation framework for the Semantic Web. *Proceedings of the First Workshop on Multimedia Annotation*. Retrieved March 19, 2006, from www.aifb.uni-karlsruhe.de/WBS/ama/publications/mma01_staabetal.pdf
- Swartz, A. (2002). *The Semantic Web in breadth*. Retrieved February 26, 2006, from <http://logicerror.com/semantic-Web-long>
- Swick, R., & Thompson, H. (1999). *The Cambridge communiqué: W3C Note 7*. Retrieved February 26, 2006, from www.w3.org/TR/1999/NOTE-schema-arch-19991007
- Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., & Ciravegna, F. (2002). Ontology-driven semi-automatic and automatic support for semantic markup. *Proceedings of the 13th International Conference on Knowledge Engineering and Management*, 379–391.
- Weinberger, D. (2005). Taxonomies to tags: From trees to piles of leaves. *Release 1.0*, 23(2), 1–33.
- Weiss, P. (1958). *Modes of being*. Carbondale: Southern Illinois University Press.
- Witbrock, M., Panton, K., Reed, S. L., Schneider, D., Aldag, B., Reimers, M., et al. (2004). Automated OWL annotation assisted by a large knowledge base. *Workshop Notes of the 2004 Workshop on Knowledge Markup and Semantic Annotation at the 3rd International Semantic Web Conference*, 71–80.
- Zalta, E. (2005). Frege's logic, theorem and foundations for arithmetic. *Stanford Encyclopedia of Philosophy*. Stanford, CA: Stanford University. Retrieved February 26, 2006, from <http://plato.stanford.edu/entries/frege-logic>