Steven Staab
University of Karlsruhe
sst@aifb.uni-karlsruhe.de

Knowledge Representation with Ontologies: The Present and Future

Christopher Brewster, *University of Sheffield*
Kieron O'Hara, *University of Southampton*

Ontologies—specifications of what exists, or what we can say about the world—have been around at least since Aristotle. At various times, philosophers have wondered whether the present King of France is bald or whether existence is a predicate. Just as scientists have grappled with the reality of negative numbers, subatomic particles, or the vital force, so have theologians and mystics grappled with the reality of God and inner spiritual experiences. The nature of knowledge is an abiding question and has resulted in people's continuous attempts to find ways to express, word, or convey their own "knowledge." Physics and mathematics depend on specific symbolic languages, and many approaches to AI regard finding the problem's optimal representation as most of the solution.

Recently, we have seen an explosion of interest in ontologies as artifacts to represent human knowledge and as critical components in knowledge management, the Semantic Web, business-to-business applications, and several other application areas. Various research communities commonly assume that ontologies are the appropriate modeling structure for representing knowledge. However, little discussion has occurred regarding the actual range of knowledge an ontology can successfully represent.

How adequate a conception of knowledge is this? Clearly, we can't easily represent certain types of knowledge (for example, skills or distributed knowledge). We can't easily transform certain types of *representation* into ontology-appropriate formats (for example, diagrammatic knowledge). Other types of knowledge are extremely suited to ontological representation, such as taxonomic information. Most, but not all, definitions of *ontology* insist that an ontology specifically represents common, shared conceptual structures. Does this requirement for publicity help guarantee adequacy? And if so, can we talk of personal ontologies?[1]

In this installment of Trends and Controversies, we bring together several practitioners to debate these issues. We have tried to secure a range of perspectives, from the philosophical to the practical, because the question of ontology is so multilayered. Indeed, we hope these essays exhibit that very multifariousness.

Sociologist and epistemologist Steve Fuller kicks off our debate by distinguishing between two views of ontology, which he calls the *Newtonian* and the *Leibnizian*. The former refers to views of ontology as finding elegant simplifying principles; the latter hopes to do justice to the extreme complexity of experience. Clearly, in the context of the Semantic Web and other knowledge management contexts, the two approaches offer contrasting advantages. Elegant ontologies might be easier to manage, but scruffy ones might be easier to apply.

Lining up for the scruffy, Leibnizian team is Yorick Wilks, who combines philosophy and linguistics to argue that you can't take a lofty, unengaged view of what exists. Every ontological theory has a viewpoint and is associated with a set of interests, and through the terms used—just like any dictionary or thesaurus—involves covert ontological commitment. Newtonian abstraction will always be a chimera.

Computer scientist Enrico Franconi represents the elegant Newtonians. Franconi, while subscribing to a belief in ontologies with unambiguous semantics, argues that in the absence of sound and complete inference engines, ontologies' full formal semantics can't be exploited. In this case, ontologies revert to being mere data structures, and Franconi sees the development of inference engines as the Semantic Web's major challenge and therefore an essential research topic.

"Ontologies with unambiguous meanings" are clear Platonic descendants as opposed to Wilks' scruffy, relativistic, and task-dependent ontologies, which are more Aristotelean in their practicality. The question is whether the ontology used depends on what you want to do with it—in other words, the task for which it is developed. Aristotle's followers answer "yes," and the Platonists say "no."[2] Indeed, although beyond this introduction's scope, it's fascinating to muse about the relationship between Newton and Leibniz and Plato and Aristotle; are they the same distinction? And if not, how do they influence each other? Surely they can't be orthogonal. The articles we've selected make many suggestive points.

If ontologies are irrevocably task-relative (that is, we characterize the world differently depending on what we're trying to do), several heterogeneous ontologies should exist. But then will the modeling overhead be too high? Mark Musen, from his experience in medical informatics, reminds us of when the

Published by the IEEE Computer Society

primary focus of research in this area was not ontologies but problem-solving methods. This research taught us much about how information gets deployed to achieve tasks, and Musen is keen that we not forget this knowledge in the rush to address fashionable representational issues. Both knowledge types are valuable.

Jeremy Ellman, CTO of Wordmap, discusses a fascinating survey on how ontologies are actually used; his research shifts the focus away from the domain properties, and even the task requirements, and toward the necessity of integration into existing systems. Given current research assumptions, it is significant that less than 10 percent of the ontologies his company has dealt with involved inferential requirements. Simon Buckingham Shum, from his perspective of research into knowledge media, endorses this view, arguing that over-engineered systems simply won't be applied. An interesting theme that emerges with Buckingham Shum and Musen's work in particular is the tight link between ontologies, tasks, and tool building.

Returning to our earlier concern about knowledge that ontologies can't capture, the question arises: what is that knowledge? This question sounds philosophical, but it has massive practical implications for the evaluation of ontologies, the cost-benefit analysis of modeling and KA programs,[3] and the feasibility and suitability of obtaining that knowledge from texts.[4] This question might be impossible to answer, at least until we have a clear consensus about what an ontology actually is. (Ellman and Musen, for example, both argue that this consensus has yet to emerge.) Buckingham Shum points out that organizations' properties are often inimical to the consensus and knowledge maintenance required to keep ontologies relevant, although their integration into existing practices will mitigate the problem. Even more significantly, he questions whether a text-based infrastructure is suited to knowledge present in multiple modalities.

Wilks, Franconi, and Fuller, in their different ways, imply that how this final question is framed, and how ontologies are understood, will dramatically affect the answer. Even in this pragmatic context, no less than a balance between technical and philosophical analysis will approach an answer. For this inquiry, no discipline is dispensable.

References

1. J. Tennison, K. O'Hara, and N. Shadbolt, "APECKS: Using and Evaluating a Tool for Ontology Construction with Internal and External KA Support," *Int'l J. Human-Computer Studies*, vol. 56, no. 4, Apr. 2002, pp. 375–422.
2. K. O'Hara, *Plato and the Internet*, Icon Books, 2002.
3. N. Shadbolt, K. O'Hara, and L. Crow, "The Experimental Evaluation of Knowledge Acquisition Techniques and Methods: History, Problems, and New Directions," *Int'l J. Human-Computer Studies*, vol. 51, no. 4, Oct. 1999, pp. 729–755.
4. C. Brewster, F. Ciravegna, and Y. Wilks, "Text as Knowledge Maintenance: Background Knowledge in Ontology Learning," *Proc. Workshop Knowledge Management and the Semantic Web* (K-CAP 2003), 2003.

If Everything Always Is, Why Hasn't There Always Been Ontology?

Steve Fuller, *University of Warwick*

If ontology is indeed the first philosophy—the most fundamental examination of being "as such"—why did it acquire currency only in the 18th century, by the end of which Kant had already deemed it a pseudoscience? The answer to this question provides what philosophers call an *axiological backdrop* for contemporary discussions about ontology's role in knowledge management (KM). In other words, the attitude you have toward ontology reflects what you believe are the values inquiry exemplifies.

Two views of ontology

Leibniz's student, Christian Wolff, popularized ontology. Wolff was influenced by his master's concerns with Newtonian mechanics' theological implications. Newton had a rather distinctive way of interpreting Aristotle's economic definition of science as that which explains the most by the least: most of the most turns out to be not worth explaining at all. This, in a nutshell, captures the reductionist sentiment that let Newton encompass all physical motion—both in the heavens and on earth—in a neat set of three laws and one universal principle. It follows that most of what appears significant to us—notably, the sensory qualities of moving bodies—is irrelevant to the divine blueprint. Newton could accept this conclusion as an operationalization of the qualitative difference between the clarity of God's mind and the confusion of our own, which we can bridge only if God wills it (as Newton clearly thought, at least in his own case). Cognition consists of abstracting robust patterns from noisy data, such that the complex becomes simple.

However, Leibniz questioned Newtonian mechanics' piety because it seemed to suggest that God routinely generates waste. After all, His favorite creatures, humans, manage to register distinctions that are pointless except as deviations from an ideal type or variations on a mathematical theme (or simply "the value of a variable," to recall Quine's notorious definition of being). Why would God burden us with the distractions of sensory experience when they only serve to impede our ability to detect the simple patterns Newton postulated? But perhaps in raising this question, Newton's own system is dubious because—at least according to Leibniz and his followers—God acts in accordance with the "principle of sufficient reason." In other words, God wastes nothing. Nature's complexity is thus an invitation for us to provide direction for something that could develop in many different ways; hence, our need for free will. Cognition then is tantamount to construction or—for those who refuse the slide down the slippery slope from thought to action—simulated construction.

One thing is clear from this brief early history of ontology. Both Newton and Leibniz presuppose that access to reality requires effort. Ontology is the product of that effort. The question separating them is the nature of the effort: Is it a kind of subtraction from (Newton) or addition to (Leibniz) what is given in experience?

Does cognition replace our confused concepts with clear ones (Newton) or let us conceptualize open-ended situations (Leibniz)? Both metaphysical system-builders and KM system-builders can explore the ramifications of these alternatives.

Metaphysically speaking, the difference between Newton and Leibniz harks back to alternative conceptions of how the fabric of reality is woven. Newton focuses on *the one and the many*, Leibniz on *the part and the whole*. With the former, the issue boils down to determining the ideal type—"the one"—next to which "the many" are imperfect versions. With the latter, the issue is determining "the whole," whose properties somehow transcend those of its constituent "parts."[1]

Considerable social psychological evidence exists that this ancient distinction in metaphysical orientations corresponds to cross-cultural differences in default reasoning patterns.[2] For example, when presented with the test case, "What goes with a cow: a chicken or a bed of grass?" Westerners chose the chicken and East Asians the grass. The Westerners interpreted the task as a search for a common higher-order category, "the one" under which cow and chicken count as "the many." In contrast, the East Asians viewed the task as a search for a composite whole, in which the cow and a bed of grass are complementary parts. Unsurprisingly, Leibniz was the biggest Sinophile among major Western philosophers.

From philosophy to knowledge management

From the KM standpoint, we can find Newton's hand in specialist-driven search engines, or *expert systems*, that invite users to input vague data in return for an exact output they can use to determine a course of action. Here, the KM system is designed to anticipate the various confusions the user is likely to bring to the cognitive transaction, disentangle them (perhaps by disaggregating the likely causes of the effects the user perceives), and present a result that will focus the user's efforts, not compound his or her confusion. Phenomenology is thus the great enemy that ontology aims to overcome.[3]

In contrast, we can find Leibniz's hand in "self-programming" search engines that effectively learn to adapt to the user's needs through repeated use. In this incarnation, we can say that the KM system lets users become aware of the kind of world that their own inquiries presuppose. Whether users manage to learn as much as the search engine in the transaction is an open question: Does the self-programming system serve to deepen knowledge or merely entrench prejudice? More generally, does ontology turn out to be a realization, or merely a reification, of phenomenology?

Having followed in Leibniz's footsteps for most of his career, Kant's "critical" moment arrived when he converted to Newton. Thereafter, he ridiculed the pretensions of ontology as a discipline that tried to confer metaphysical significance on every aspect of human experience. Faced with the choice between a world view that constantly reminds humans that their experience normally falls short of grasping the structure of reality (Newton) and a world view that doesn't let humans distinguish between perfect correspondence and seamless self-deception (Leibniz), Kant argued that the former is the more intellectually responsible course of action, however daunting it renders the "interface" between us and whatever lies on the other side.

Translated into KM terms, a rich Leibnizian sense of ontology might provide a blueprint for cyborgs that incorporate the user in a harmonious knowledge system. However, the more austere Newtonian one continues to remind us that whatever the nature of the interaction between human and computer, it's not symmetrical.

References

1. S. Fuller, "Strategies of Knowledge Integration," *Our Fragile World: Challenges, Opportunities for Sustainable Development*, M.K. Tolba, ed., EOLSS Publishers, 2001, pp. 1215–1228.

2. R. Nisbett, *The Geography of Thought: How Asians and Westerners Think Differently—and Why*, Nicholas Brealey, 2003.

3. S. Fuller, *Knowledge Management Foundations*, Butterworth-Heinemann, 2002, pp. 116–195.

Are Ontologies Distinctive Enough for Computations over Knowledge?

Yorick Wilks, *University of Sheffield*

Is there a problem with ontologies? Are they really distinct from nets, graphs, thesauri, lexicons, and taxonomies, or are people just confused about any real or imagined differences? Does the word "ontology" have any single, clear, meaning when AI and natural language processing researchers use it? If not, does that matter? Are those of us in AI and NLP just muddled computer people who need to have our thoughts firmed up, cleared up, sorted out, and so on by other, more philosophical, logical, or linguistic experts so we can better perform our jobs?

I address, if not answer these questions in this essay. The last is a recurrent question in AI to which I shall declare a practical, and negative, answer. Namely, decades of experience show that for effective, performing simulations of knowledge-based intelligence, enhanced representations—those meeting any criteria derived from logic—are rarely useful in advancing those simulations.

My background

Because the topic is metaphysical, I'll declare my wrinkled hand at the beginning as far as these matters are concerned. My PhD thesis[1] was a computational study of metaphysical arguments, as contained in classic historical texts. The claim (which the exiguous computing capacity of those days barely supported) was that such arguments proceed and succeed using methods quite different from the explicit, surface argument structure their authors proposed. Rather, the methods involve rhetorical shifts of our sense of key words, and authors might not even be aware of them. For example, Spinoza's whole philosophy, set out in the form of logical proofs that are all faulty, actually aims to shift our sense for the word "nature."[2]

My early investigation alerted me to the possibility that representational structures are not always necessary where deployed, and that we can't always be sure when representations are or are not adequately complex to express some important knowledge. I think of Roger Schvaneveldt's Pathfinder networks:[3] simple, associative networks derived from word use that seem able, contrary to most intuition, to express the kinds of skills fighter pilots have. I also recall the dispute Jerry Fodor originated—that connectionist networks could not express recursive grammatical structures,[4] an argument I believe he lost when Jordan Pollack produced his recursive auto-associative networks.[5]

My theme, then, is that man-made structural objects (namely, ontologies, lexicons, and thesauri) for classifying words and worlds contain more than they appear to, or more than their authors are aware of. This is why computational work continues to mine novelty from analyzing such objects as *Webster's 7th*, the *Longman Dictionary of Contemporary English*, Wordnet, or *Roget*. Margaret Masterman memorably claimed that Roget showed his unconscious, as well as his explicit, structuring—that of a 19th-century Anglican clergyman, an opposition between good and evil.[6]

If any of this is true, then what structural objects that contain knowledge need is not conceptual clearing up but investigation. Or, as Ezra Pound once put it: "After Leibniz, a philosopher was a guy too damn lazy to work in a laboratory."

Defining "ontology"

Those cursed with a memory of metaphysics are often irritated by modern AI and NLP, where the word "ontology" rarely means what it used to—namely, the study of what there is, of being in general. Recent exceptions to this are Nicola Guarino's[7] and Graeme Hirst's[8] discussions. However, almost all modern use refers to hierarchical knowledge structures whose authors never discuss what there is but assume they know it and just want to write down the relations between the parts/wholes and sets and individuals that undoubtedly exist.

To a large extent, I'll go along with this use, noting that as a Web search term, ontology locates two disjoint literatures with virtually no personnel in common: the world of formal ontology specification[9] and the world of ontologies for language-related AI tasks.[10] Rare overlaps include the CYC system,[11] which began as an attempt to record extensive world knowledge in predicate, but which its designer Douglas Lenat also claimed as a possible knowledge form for language processing.

We must begin with one of my earlier questions about the conflation of ontologies (construed as hierarchical classifications of things or entities) and thesauri or taxonomies (hierarchical classifications of words or lexical senses). A widespread belief exists that these are different constructs—as different (on another dimension) as encyclopedias and dictionaries—and should be shown as such. Others will admit that they are often mixed together. For example, Wordnet[12] is called an ontology, which it sometimes is, but this might not matter as regards its function as the most popular NLP resource, any more than it matters that dictionaries contain many world facts, such as "a chrysanthemum is a flower that grows at Alpine elevations."

Problems with formalization

A random paper I reviewed last month offered an ontological coding scheme, comprising what it called *universal words*, and whose first example item was

(Drink > liquor).

This was included to signify, through "universal words" that "drink is a type of liquor." At first, this seems the reverse of common sense—liquors (distilled alcoholic drinks) are a type of drink, and the symbols as written suggest that drink includes liquor, which is broadly true. However, if the text as written contains a misprint, and "liquid" is intended instead of "liquor," the quote is true, but the symbols are misleading.

We probably can't interpret the words in any straightforward way that will make the quotation true, but the situation is certainly more complex because "drink" has at least two relevant senses (potable versus alcoholic drink) and liquor has two as well (distillate versus potable distillate). This issue is always present in systems that claim to be ontologies, not systems using lexical concepts or items. As such, these systems claim to be using symbols that aren't words in a language (usually English), but rather are idealized or arbitrary items that only contingently look like English words.

It is not sufficient to say, as some such as Sergei Nirenburg consistently maintain, that ontological items simply seem like English words, and he and I have discussed this issue elsewhere.[10] I firmly believe that items in ontologies and taxonomies are and remain words in natural languages—the very ones they seem to be, in fact—and that this strongly constrains the degrees of formalization we can achieve using such structures. The word "drink" has many meanings (for example, the sea) and attempts to restrict it within structures by rule, constraint, or the domain used can only have limited success. Moreover, there is no way out using nonlinguistic symbols or numbers, for the reasons Drew McDermott explores.[13] Those who continue to maintain that "universal words" aren't the English words they look most like, must at least tell us which sense of that closest word is intended to bear under formalization.

When faced with the demand I just mentioned, a traditional move is to say that science doesn't require that kind of precision at all levels of a structure, but rather that "higher-level" abstract terms in a theory gain their meaning from the theory as a whole. Jerrold Katz adopted this view for the meaning of terms like "positron."[14] From a different position in the philosophy of science, writers such as Richard Braithwaite[15] argued that we should interpret scientific terms (such as "positron") at the most abstract level of scientific theory by a process of what he called *semantic ascent* from the interpretations of lower, more empirical terms.

This argument is ingenious but defective because a hierarchical ontology or lexicon isn't like a scientific theory (although both have the same top-bottom, abstract-concrete correlation). The latter isn't a classification of life but a sequential proof from axiomatic forms.

However, what this analogy expresses is in the spirit of Quine's later views[16]—namely, not all levels of a theory measure up to the world in the same way, and no absolute distinction exists between high and low levels. If this is the case, a serious challenge remains for ontologies claiming any degree of formality: How can the designers or users control the sense and extension of the terms used and protect them from arbitrary change in use?

References

1. Y. Wilks, *Argument and Proof in Metaphysics*, doctoral thesis, Dept. of Philosophy, Cambridge Univ., 1968.
2. R. G. Bosanquet, "Remarks on Spinoza's Ethics," *Mind*, vol. 59, 1945, pp. 264–271.
3. R.W. Schvaneveldt, ed., *Pathfinder Associative Networks: Studies in Knowledge Organization*, Ablex, 1990.
4. J. Fodor and Z. Pylyshyn, "Connectionism and Cognitive Architecture," *Cognition*, vol. 28, nos. 1–2, Mar. 1988, pp. 3–71.
5. J.B. Pollack, "Recursive Auto-Associative Memory: Devising Compositional Distributed Representations," *Proc. 10th Ann. Conf. Cognitive Science Society*, Lawrence Earlbaum, 1988, pp. 33–39.
6. M. Masterman, *What is a Thesaurus?* memorandum ML-95, Cambridge Language Research Unit, 1959.

7. N. Guarino, "Ontological Principles for Designing Upper-Level Lexical Resources," *Proc. 1st Int'l Conf. Language Resources and Evaluation*, European Language Resources Assoc., 1998, pp. 527–534.

8. G. Hirst, "Existence Assumptions in Knowledge Representation," *Artificial Intelligence*, vol. 49, nos. 1–3, May, 1991, pp. 199–242.

9. D. Fensel et al., "OIL in a Nutshell," *Knowledge Acquisition, Modeling and Management, Proc. 12th Int'l Conf.* (EKAW 2000), LNCS 1937, Springer, 2000, pp. 1–16.

10. S. Nirenburg and Y. Wilks, "What's In a Symbol: Ontology, Representation, and Language," *J. Experimental and Theoretical Artificial Intelligence*, vol. 13, no. 1, Jan. 2001, pp. 9–23.

11. D.B. Lenat and R.V. Guha, *Building Large Knowledge-Based Systems,* Addison-Wesley, 1990.

12. C. Fellbaum, *Wordnet: An Electronic Lexical Database and Some of its Applications*, MIT Press, 1998.

13. D. McDermott, "Artificial Intelligence Meets Natural Stupidity," *ACM SIGART Newsletter*, vol. 57, Apr. 1976, pp. 4–9; reprinted in *Mind Design*, J. Haugeland, ed., MIT Press, 1981.

14. J.J. Katz, *Semantic Theory*, Harper and Row, 1972.

15. R.B. Braithwaite, *Scientific Explanation*, Cambridge Univ. Press, 1953.

16. W.V.O. Quine, "The Problem of Meaning in Linguistics," *From a Logical Point of View*, Harper and Row, 1963.

Using Ontologies

Enrico Franconi, *Free University of Bozen-Bolzano*

We now have ontologies all over the place—or we will have them soon. We have (or are headed toward) several standard languages in which to write them so that we can have a common understanding about their content. We need such standardization if ontologies are to automate information exchange by supporting the retrieval and understanding of data involved in transactions. The research community has spent considerable effort giving these ontology languages a formal semantics, making ontologies' meaning completely unambiguous—at least on paper. These logic-based ontology languages now form the foundation for the Semantic Web's layered architecture.

The first step

Having ontologies with unambiguous meanings is just the first step toward the Semantic Web vision or toward a fully automated, ontology-supported business-to-business scenario. Of course, in certain fields and applications, just agreeing on the meaning of the terms involved is an important step forward. Consider, for example, medical terminology or specialized terminologies in different businesses. In fact, applications in these fields could more easily interoperate if they were built with a common understanding of the information structure. This first step is limited when applications view terms in the ontologies as data structures, neglecting the ontology definitions' full semantic content.

Why we need ontology languages

If all we can do with ontologies is use them as basic abstract data types, the whole effort of giving a well-founded semantics to expressive ontology languages is useless. People tend to use ontologies only as frames to instantiate their applications, so a simple frame-based ontology language would suffice. Java or Corba IDL's class structure is definitely enough for these purposes. We don't need to put an entire legion of computational logicians to work for that. In fact, you can already find proposals for novel ontology languages with no formal semantics. These languages, according to their proposers, should supposedly overcome some of the big players' limitations. However, they can't compare with semantically based approaches, such as OWL. In fact, the underlying assumption that an ontology language should have an unambiguous, well-understood meaning doesn't hold anymore. Ontologies written in these languages can only play the role of shared data structures in interoperating applications.

If we accept that an ontology language should have an unambiguous meaning, then we probably also want to have the most expressive language possible in this framework. This rationale has been behind the discussions about the OWL Full language, which turned out to be very expressive—beyond first-order logic. This means that OWL Full can express most of the details we'd want described in an ontology, but other agents can still understand that ontology without any problems.

However, just because other agents can understand expressive ontologies, that doesn't mean applications can use them properly. Unless an application uses an ontology as a simple data structure, the application must properly consider the ontology definitions' full semantic content. This is the only way to guarantee that the application exploits all the information the ontology represents. The information's explicit structure might change considerably owing to the information contained in the ontology definitions. In an interoperability framework, you must exploit such implicit information because different applications might use, publish, or subscribe to data structured in different ways but still consistent with the shared ontology.

The trouble with incomplete inference engines

An ontology inference engine (such as iFaCT or Racer) can offer a reasoning service to applications willing to properly use an ontology. The inferential process's complexity depends strictly on the adopted ontology language's expressivity. We now have three layers of ontology languages (OWL Lite, OWL DL, and OWL Full, in order of expressivity) because the inference engine becomes increasingly complex as the ontology language becomes more expressive. In fact, theoreticians have proved that you can't build a complete inference engine for OWL Full, although it's possible to use existing description logic systems as inference engines for OWL Lite and OWL DL.

Designing and implementing complete inference engines for expressive ontology languages isn't easy. As a prerequisite, you must have formal proof that the algorithms are complete with respect to the ontology language's declared semantics. The description logics community has 20-plus years of experience to help provide theoretical results, algorithms, and efficient inference systems for all but the most expressive OWL languages. We can understand how important it is for an inference engine to be complete with the following example.

Suppose a military agency asks you to write an ontology to recognize whether a particular individual description indicates some sort of "enemy" concept so that an application can take appropriate automatic action (such as shooting) given the inference engine's answer. If the inference engine is sound but incomplete, it will recognize most but not all enemies because it isn't a complete reasoner. Because it is sound, however, it won't confuse a friendly soldier with an enemy. So, the application will start the automatic shooting procedure only when the system recognizes without doubt that someone is an enemy. The application could fail to shoot an enemy, but field soldiers can take traditional backup (nonautomatic) action. Soundness is more important because you don't want to shoot your own soldiers. So far, so good.

The agency has another application strictly related to the first one. The task is now to recognize an individual description as an allied soldier to activate automatic procedures that will alert the soldier to the headquarters' secret position. Again, the system must have a sound inference engine because the agency doesn't want to disclose secret information to enemies. Moreover, incompleteness is not a major problem because the defense system can still be valid even if a soldier doesn't know where the headquarters is located.

The agency decides, of course, to use the same shared ontology for both applications. After all, the task in one case is to decide whether a soldier is an enemy and in the other case decide whether he or she isn't. So the second application can use the same ontology as the first, but it exploits the outcome in a dual way. Unfortunately, it turns out that the agency can't use the same ontology for both tasks if the ontology language's inference engine is sound but incomplete. If a sound but incomplete reasoning system exists for solving, say, the first problem (recognizing enemies), you can't use the same reasoning system as a sound (and possibly incomplete) procedure for solving the second problem (recognizing allies). In fact, using the same procedure for solving the second problem would be unsound—it will say an individual isn't an enemy when he or she actually is. Although this is harmless for the first problem, it is bad for the second, dual one. It would disclose valuable military secrets to enemies.

To solve this problem, you must have both a sound and complete inference engine for the ontology language. This rules out using OWL Full seriously for interoperating applications because having a complete inference engine with this language is impossible. This also rules out using inference engines with unknown completeness. And no formal property is known for most of the inference engine proposals for OWL family ontology languages implemented in the Semantic Web—both for ontology reasoning and for query answering with ontologies. Usually, developers perform a check against only a small class of benchmarks and use cases. Clearly, this is a bad practice, and I hope Semantic Web tool developers will consider the problem of properly using ontologies more seriously in the future.

Ontologies: Necessary—Indeed Essential—but Not Sufficient

Mark A. Musen, *Stanford University*

After more than a decade of discussion, the AI community still hasn't reached complete consensus on what, precisely, an ontology is. The arguments have become more polite and heads bob more in unison, but messiness still exists. Tom Gruber's definition, that an ontology is an explicit specification of a conceptualization,[1] still holds true, but it has some rough edges.

The main problem is that there are no agreed-upon borders concerning what is in a *specification*. Certainly, a taxonomic hierarchy of concepts is an appropriate specification, although taxonomies by themselves are rather impoverished in what they can represent. Add the notion of concepts' attributes, and you'll hear no arguments. Constraints on attributes' values seem reasonable, but with what degree of expressiveness? Do we allow only simple role restrictions, as in most description logics, or is there room for statements in more expressive logics, as in Ontolingua?[1] Do we allow instances as well as classes? If so, which kinds? For many developers of intelligent systems, UML comes to mind first when you say "ontology specification language"; for others, it's OWL (www.w3.org/TR/owl-features); and for still others, it's CycL (www.opencyc.org). When it comes time to commit a conceptualization to some specification, we can choose from a wide range of languages. These ontology languages differ greatly in their expressive power, their support for inference, their integration with software engineering environments, and their perspicuity and comprehensibility.

Questions such as, "How much knowledge can ontologies represent?" and "How adequate are ontologies for knowledge representation?" view ontologies as knowledge representations. Such questions are simply unanswerable, given that ontology-specification languages vary so greatly in their expressive power. It's much more helpful to concentrate on what's in a *conceptualization* in the first place, rather than dwell on the expressiveness of particular vehicles for encoding ontologies.

Although no simple predicate tells us unambiguously whether a particular specification is an ontology, we can still agree on certain things. We can agree that ontologies enumerate the salient concepts in an application area. We can agree that ontologies typically define concepts' properties and the relationships among concepts and often indicate constraints on those properties and relationships. An ontology provides a domain of discourse for discussing a given application area, but it does not—and cannot—represent all of an agent's knowledge.

Ontologies are radically incomplete knowledge representations

In his classic paper, Alan Newell[2] characterized knowledge as a behavioral phenomenon. He viewed knowledge in terms of an agent's goals, the actions of which the agent might be capable, and the means by which the agent selects actions to achieve its goals. A key observation was that knowledge lets an agent enact procedures to attain its goals, and that we attribute knowledge to an agent because we observe it behaving in the world in an apparently rational manner.

This view of knowledge goes well beyond the notion of a specification of a conceptualization, of an enumeration of concepts and relationships. From Newell's perspective, knowledge does more than account for what exists in the world; it directly links goals to actions. In that sense, knowledge has a strongly procedural element. When intelligent-system developers can model and discuss those procedures explicitly, they gain insight into how a system should use propositional knowledge to achieve its goals and what additional knowledge the system might require when reasoning fails.

Nearly 20 years ago, Bill Clancey[3] and B. Chandrasekaran[4] spotlighted the recurrent patterns of inference in various knowledge-based systems, emphasizing the importance of clarifying the procedural aspects of problem solving in addition to propositional knowledge about the domain. Their observations launched significant work in the knowledge-acquisition community to identify, model, and codify problem-solving methods (PSMs) that could form the basis for the procedural components of knowledge-based systems. These efforts led to libraries of planners, classifiers, constraint satisfaction engines, case-based reasoners, and other PSMs that define procedurally how systems can use domain knowledge to solve specific tasks. Methodologies for building intelligent systems such as CommonKADS and Protégé provided specific guidance for using PSMs to encode the procedural knowledge needed to solve domain tasks in a computationally expeditious manner.[5]

The notion of a PSM was different from that of a traditional "inference engine," such as backward chaining. Inference engines operate on *data structures*, as in the case of a backward chainer programmed to search a database of rules for one whose conclusion contains the same parameters as those on its left-hand side. PSMs, unlike inference engines, operate at the knowledge level.[4] They provide abstract procedures by which agents can use their knowledge to achieve particular goals. PSMs do not construe problem solving in terms of operations on data structures such as rules or frames. Instead, they construe problem solving in terms of operations on propositional knowledge (for example, knowledge specified as ontologies).

In the 1990s, the idea of encapsulating procedural knowledge as PSMs caught on like wildfire in the academic community. Developers around the world began to apply well-known PSMs such as *heuristic classification* and *propose-and-revise* to various application tasks. Our laboratory[6] and many others[7] began to focus on experiments with PSMs.

True, with enough theorem-proving power, there is no need to extract out problem-solving knowledge if we can find a problem's solution somewhere within the deductive closure of a large set of axioms. However, the PSM approach has obvious advantages. Developers suddenly have insight concerning how a system can use its domain knowledge to achieve task goals. When the developers inspect the propositions in a knowledge base, the role that any entry plays in problem solving is immediately apparent by noting how the relevant PSM used the knowledge at runtime. PSMs identify the additional propositions a knowledge base should receive for maximum computational efficiency. Knowledge bases also achieve considerable parsimony: If an axiom is irrelevant to a given PSM's problem-solving requirements, there is no need to include the axiom in the knowledge base.

Where have all the problem solvers gone?

In recent years, the excitement over ontologies has eclipsed almost everything else in applied AI. Nearly every issue of *IEEE Intelligent Systems* has articles about ontologies. As of this writing, Google has indexed 1,620,000 Web pages that contain the string "ontology." The very word has gone from a technical term that people once uttered tentatively in academic circles to a buzz word that pervades current thought about knowledge-based systems.

> To build systems that solve real-world tasks, we must not only specify our conceptualizations, but also clarify how problem solving ideally will occur.

This new emphasis on ontologies isn't surprising. It reflects the important role that ontologies play in structuring our collections of propositional knowledge and in providing shared domain descriptions for various purposes. The notion of reusable ontologies has been a pivotal idea for AI, but ontologies aren't enough.

If we care about problem-solving tasks for which we have enough knowledge in advance to predict that certain solution strategies will be particularly well suited, then we need PSMs as a part of our basic set of knowledge-base building blocks. If we know that a particular problem-solving approach is appropriate for addressing a domain task that we wish to automate, then the corresponding PSM's knowledge requirements can help guide knowledge acquisition. The knowledge requirements can also ensure that our knowledge bases make the distinctions necessary to enable successful and efficient problem solving.

The fiery debates over the virtues of procedural versus declarative knowledge representations (which bogged down AI 30 years ago) might be casting a long shadow over our current work. We left the 1970s convinced that procedural knowledge representations had significant limitations, but many people in AI seem to have extended this condemnation to procedural knowledge itself. The ability to follow procedures is an inherent element of intelligence and of knowledge-based systems as well. When a system must solve a real-world problem such as designing an artifact or classifying an abnormal pattern, a strong representation of the necessary problem-solving procedure not only makes the runtime system more computationally efficient, but also enhances the software engineering of the knowledge base. The use of an explicit PSM links each knowledge-base entry to its role in problem solving and thus makes the system more understandable and traceable. Many intelligent systems are designed primarily to answer queries about large bodies of knowledge. In these cases, ontologies provide most of the representational muscle needed to build complete systems. To build systems that solve real-world tasks, however, we must not only specify our conceptualizations, but also clarify how problem solving ideally will occur. We must link our ontologies to the knowledge requirements of appropriate PSMs.

References

1. T.R. Gruber, "A Translation Approach to Portable Ontologies," *Knowledge Acquisition*, vol. 5, no. 2, Apr. 1993, pp. 199–220.
2. A. Newell, "The Knowledge Level," *Artificial Intelligence,* vol. 18, no. 1, Jan. 1982, pp. 87–127.
3. W.J. Clancey, "Heuristic Classification," *Artificial Intelligence*, vol. 27, nos. 1–2, Dec. 1985, pp. 289–350.
4. B. Chandrasekaran, "Generic Tasks in Knowledge-Based Reasoning: High-Level Building Blocks for Expert-System Design," *IEEE Expert*, vol. 1, Feb. 1986, pp. 20–23.

5. M.A. Musen and A.T. Schreiber, "Architectures for Intelligent Systems Based on Reusable Components," *Artificial Intelligence in Medicine,* vol. 7, no. 3, June 1995, pp. 189–199.

6. H. Eriksson et al., "Task Modeling with Reusable Problem-Solving Methods," *Artificial Intelligence*, vol. 79, no. 2, Dec. 1995, pp. 293–326.

7. R. Benjamins and D. Fensel, eds., Special Issue on Problem-Solving Methods, *Int'l J. Human-Computer Studies*, vol. 49, no. 4, Oct. 1998, pp. 305–649.

Corporate Ontologies as Information Interfaces

Jeremy Ellman, *Wordmap*

Internet access has made huge information resources available on corporate desktops. However, this has often led to confusion as intranet, content, and document management systems use different and competing access methods. Recognizing this, corporate information architects are embracing ontologies to unify and simplify navigation and search.

Defining corporate ontologies

"Ontology" in this sense is a generic, rarely defined, catch-all term. Some ontologies are strict hierarchies of category names and nothing more. Others are taxonomies or graphs where categories occur in more than one place and loops are allowed in the data structure. Still others support corporate vocabularies of preferred terms and their synonyms that might be multilingual. Others require complex metadata structures that you can use to prioritize retrieval, to indicate retention policy or legislative compliance, or for the technical support of linked applications. Consequently, there's a unity of purpose rather than technology.

Corporate ontologies, whatever their level of sophistication, aim to support the systematization of large volumes of information using abstraction. This is almost completely at odds with the AI ontologies of the mid 1970s and early 1980s, where the aim was often to represent a small domain in high detail.

Corporate ontologies then face several issues not generally considered in academic research. These include security and ownership, trust, the intended audience, and the media used to view the ontology.

It seems obvious that the ontologies used in enterprises reflect the size of those enterprises. As such, they are likely to be large and partitioned according to the enterprise's interests. If the ontology is internal, it will likely reflect the divisions seen within the organization, such as marketing, R&D, human resources, and so on. If the ontology is for external use, it might represent product categories, support and sales divisions, and so on. In both cases, a separate part of the organization will "own" each section of the ontology and will be (or will want to be) responsible for its development and maintenance. So, security mechanisms are required so that only those authorized can edit or even view the ontology sections for which they're responsible.

Audiences are critical of corporate ontologies because they might have different information requirements. For example, an audience might be multilingual and prefer to view its ontology in its local language. However, audiences might have completely different interests and perspectives that require wholly (or seemingly) different representations of or interfaces to the same ontology.

Consider, for example, a corporate internal ontology: marketing's view of R&D is going to be quite different than R&D's view of itself because marketing doesn't need to be aware of technological intricacies. Similarly, R&D will have a simple view of marketing, although human resources might have a more balanced, but different, view of both.

The media you use to visualize an ontology also plays a role because it imposes different constraints. For example, organizations often have catalogs of products or services that they can distribute in print, on CD, or over the Web. Obviously, the information structure they use should be identical to minimize development and maintenance costs. Yet, size is a factor with paper because it rapidly becomes both heavy and expensive, while the branching factor is an issue with screen-based media because too many alternatives rapidly exhaust end users. We could even say that function and use drive the corporate ontology.

Quantifying ontology issues

So far, I have pointed out several factors that occur when you use ontologies as information interfaces. These factors don't often appear in the ontology literature, such as the intended audience and the interface's usage context, ownership issues, protection, and security. These issues are not equally prevalent, however. To quantify some of these arguments, Wordmap has briefly analyzed 22 ontologies or taxonomies that we have encountered over the past three years. From the knowledge representation viewpoint, all were hierarchical, although at least two were purposely designed to be no more than four-ply deep to simplify navigation. Nineteen ontologies required metadata to be associated with their categories, while 14 included synonyms for headwords. Seven ontologies included at least one language in addition to English headwords (these categories are not mutually exclusive). Interestingly, eight ontologies out of the 22 were based on existing resources such as thesauri or subject catalogs. This indicates the value in ontology content and the requirement to reuse existing work through data conversion. Note that developing a new ontology formalism is far easier than developing the content to populate it in sufficient depth to make it useful in a real application.

When we consider what their owners would use these ontologies for, 12 out of 22 were either subject or product catalogs that help provide navigable or searchable Web interfaces. Ten link to what we might loosely interpret as either content or document management systems, and six are used in text-classification systems. These might be either automatic or manual as part of the document metadata assignment carried out by people entering documents into retrieval systems.

Only two ontologies' representations vary depending on the media being used, and two more have forms that change depending on the viewing audience. Although these proportions are small, we must consider that supporting software for these aspirations is not available. Additionally, just two ontologies have clear inference requirements for which knowledge-based systems technology is necessary.

Finally, four ontologies had been derived from authoritative industry-standard vocabularies. This reflects the understandable desire to adhere to a common conceptual representation while avoiding the development costs associated with building content for new ontologies.

The UK government is promoting two further ontologies as standards for organizing public information in national and local government. This would make searching skills learned in the context of one official Web site transferable to others.

I doubt that anything in my discussion would surprise a corporate information architect. The challenges in representing knowledge with ontologies don't seem to lie in issues of representational adequacy or underlying formalism. Rather, we find them in the mechanics of integration with existing systems and the design or acquisition of content that's appropriate to the required function in the information interface.

Contentious, Dynamic, Multimodal Domains … and Ontologies?

Simon Buckingham Shum,
The Open University

As journalists and politicians know too well, sometimes simply asking a particular question is enough to make a point. Swinging the spotlight onto issues that people rarely discuss is a good mental hygiene exercise for the Semantic Web at this point in its young life.

It's about time to consider a six-month health-check for this increasingly active toddler, where we verify more rigorously than its doting parents can manage that the child is indeed seeing and hearing clearly.

The challenge

In the spirit of this collection of essays, I adopt a contentious stance to make my point. I focus on arguably the most controversial application proposed for the Semantic Web—namely, knowledge management in organizations. (The fact that it's not always seen as controversial demonstrates that all is not well). Now that KM has begun to mature, those who didn't see through the technocentric hype in the early days are rapidly realizing what others sought to emphasize above the roar of computing vendors and AI researchers revving their engines: the dominant metaphor in much real-world knowledge work is not the abstracted, indexed, textual *knowledge object*, but rather the situated, embodied *sensemaking process*.

Organizations need to make sense of rapidly changing environments where the questions (never mind the answers) might not be clear, where organizational and other politics make certain ideas untenable, where incomplete knowledge and backgrounds make understanding perspectives central, where expertise must be combined and reconfigured in the light of discussion, and where information must be interpreted in a timely manner. Contentious, dynamic domains, requiring good enough, timely, collective sensemaking on incomplete, multimedia information—quite a sobering reality check for any prospective knowledge infrastructure to confront. However, such considerations are well documented by leading thinkers such as Karl Weick[1] and John Seely Brown.[2]

Enter ontologies

What do ontologies require to operate?

- *Consensus*: An ontology is an agreed conceptualization of how the world is.
- *Hand-crafting*: We can't automatically construct nontrivial ontologies at present.
- *Maintenance*: Our world view changes, and so must our ontologies, or we're modeling a fiction.
- *Textual expression*: "If you can't type it, it doesn't mean anything" is not a promising precondition for a world where meaning is clothed in multiple modalities.

On the face of it, ontologies don't shape up as promising contenders for the knowledge infrastructure backbone.

May many Semantic Webs bloom

Clearly, ontology-based knowledge representation is relevant for stable, well-understood problems with well-known problem-solving methods. Organizations have a huge requirement for database integration and machine-machine interoperability. In such domains, we can even trust ontology-based agents to negotiate autonomously within the well-defined boundaries. The clear implication is that if deployed for KM, we're talking about innumerable Semantic Webs—islands of coherence whose members subscribe to that world view enough to publish and consume services with a degree of trust.

But away from these quiet backwaters, on the wilder rapids of organizational sensemaking, the brittle ontological canoe might snap. So, many conclude that this vessel is simply not the one to ride when shooting these rapids.

Making ontologies less brittle

Semantic Web adherents, however, keep the faith and are demonstrating how we can make the canoe more flexible. To adopt a less confrontational stance, I'm more than happy to recognize that this is where we find some of the most interesting work at present. The emphasis by people such as Jim Hendler on "scruffy" reasoning is absolutely right, as exemplified in the Advanced Knowledge Technologies project (www.aktors.org), some of whose work I'll turn to next.

Simple reasoning over multiple databases might prove an interesting strategy. The ontologies assist with the data capture, data integration, and the reasoning service definitions, resulting in added value through productive combinations of previously disparate data sources (see www.aktors.org/technologies/csaktivespace). *Language technologies* hold at least one key for addressing the capture bottleneck that plagues any formal representation. Agents can harvest text from the Web or other live sources and interpret it ontologically to keep ontologies populated with up-to-date instances.[3] This doesn't, however, revise the ontological structure itself, only instances. *Services* that we can rapidly define, publish, and configure within a corporate Semantic Web intranet can exploit distributed expertise.[4] In principle, service providers with completely different ontologies could still interoperate, but we know how hard making this work really is, and in business practice, the case is not yet proven.

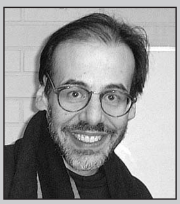
Mixing formality and informality to support collaborative knowledge work

My own work deals with collaborative analysis and sensemaking. A large "collaboration technologies graveyard" exists of over-engineered systems that didn't recognize the target end-user community's work practices, and so were dumped.

My strategy combines semiformal and formal semantics with the informality inherent in collaborative work. For example, the e-Science CoAKTinG project (www.aktors.org/coakting) combines free-text instant messaging, visual online presence cues, and "dialog mapping" using a semiformal concept mapping tool with audio and video records of virtual meetings. There's also a project ontology of the people, events, technologies, and organizations to provide integration with other

Christopher Brewster is a PhD student in the Department of Computer Science at the University of Sheffield, UK. Contact him at c.brewster@dcs.shef.ac.uk.

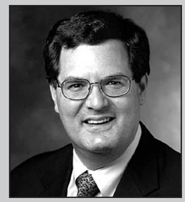
Kieron O'Hara is a senior research fellow in the Intelligence, Agents, Multimedia Group, Dept. of Electronics and Computer Science at the University of Southampton. Contact him at kmo@ecs.soton.ac.uk.



Steve Fuller is a professor of sociology at the University of Warwick. Contact him at s.w.Fuller@warwick.ac.uk; www.warwick.ac.uk/~sysdt/index.html.

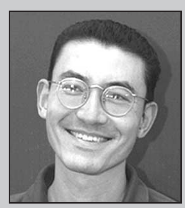
Yorick Wilks is a professor of computer science at the University of Sheffield and the director of the Institute of Language, Speech, and Hearing. Contact him at y.wilks@dcs.shef.ac.uk.

Enrico Franconi is an associate professor at the Faculty of Computer Science of the Free University of Bozen-Bolzano, Italy. Contact him at franconi@inf.unibz.it.



Mark A. Musen is a professor of medical informatics and computer science at Stanford University and heads the Stanford Medical Informatics laboratory. Contact him at musen@smi.stanford.edu.

Jeremy Ellman is a senior lecturer in informatics at the University of Northumbria and was Chief Technology Officer at Wordmap. Contact him jeremy.ellman@northumbria.ac.uk.



Simon Buckingham Shum is a senior lecturer at the UK Open University's Knowledge Media Institute. Contact him at sbs@acm.org.

ontology-based resources and services.[5] How people want to communicate leads the requirements that the tools deliver. If the tools don't help the work you have to do, then you just don't use them.

Supporting conflicting interpretations and perspectives

A second example is the Scholarly Ontologies project. What can the Semantic Web offer in domains where there is little consensus, no master view, and conflicting perspectives? In the Scholarly Ontologies project (www.kmi.open.ac.uk/projects/scholonto), we're developing a semantic digital library server that provides services for researchers whose business is, of course, constructing and debating world views. Our tools provide a discourse ontology for making, extending, and challenging *claims*. Although we still want to deliver useful knowledge services, we must relax many of our normal knowledge engineering assumptions for nonengineers who want to construct distributed, collaborative knowledge bases.[6] Although currently being applied to research literatures, the underlying approach applies to any domain where it's as important to capture principled disagreement as it is to capture consensus.

Diagnosis

So, health-check over; is the infant okay, doctor? It's probably too early to tell. Some early heart murmurs might cause concern, but they could pass with time. The key is not to smother the child in cotton wool. The recommended regime is lots of exercise out in the dirt with other children to make sure that the child is properly socialized and develops the right immunities in the rough and tumble competitive world. If in the end, no one will play with him, you'll only have yourself to blame, Mrs. Ann O'Tate. Next, please.

Acknowledgments

The views in this piece are my own and are not necessarily shared by my colleagues in the cited projects and publications. I gratefully acknowledge the support of the UK EPSRC, who fund the ScholOnto, AKT, and CoAKTinG projects.

References

1. K.E. Weick, *Sensemaking in Organizations*, Sage Publications, 1995.
2. J.S. Brown and P. Duguid, *The Social Life of Information*, Harvard Business School Press, 2000.
3. F. Ciravegna, "Challenges in Information Extraction from Text for Knowledge Management," *IEEE Intelligent Systems*, vol. 16, no. 6, Nov./Dec. 2000, pp. 84–86.
4. E. Motta et al., "IRS-II: A Framework and Infrastructure for Semantic Web Services," *Proc. 2nd Int'l Semantic Web Conf.* (ISWC 2003), LNCS 2870, Springer-Verlag, 2003, pp. 306–318.
5. S. Buckingham Shum et al., "CoAKTinG: Collaborative Advanced Knowledge Technologies in the Grid," *2nd Workshop Advanced Collaborative Environments,* held in conjunction with *11th IEEE Int'l Symp. High-Performance Distributed Computing*, 2002; http://eprints.aktors.org/archive/00000156.
6. V. Uren et al., "Interfaces for Capturing Interpretations of Research Literature," *Workshop Distributed and Collaborative Knowledge Capture* (DC-KCAP 03), 2003; http://kmi.open.ac.uk/publications/papers/kmi-tr-130.pdf.