

What Are Ontologies, and Why Do We Need Them?

B. Chandrasekaran and John R. Josephson, Ohio State University
 V. Richard Benjamins, University of Amsterdam

THEORIES IN AI FALL INTO TWO broad categories: *mechanism* theories and *content* theories. Ontologies are content theories about the sorts of objects, properties of objects, and relations between objects that are possible in a specified domain of knowledge. They provide potential terms for describing our knowledge about the domain.

In this article, we survey the recent development of the field of ontologies in AI. We point to the somewhat different roles ontologies play in information systems, natural-language understanding, and knowledge-based systems. Most research on ontologies focuses on what one might characterize as domain factual knowledge, because knowledge of that type is particularly useful in natural-language understanding. There is another class of ontologies that are important in KBS—one that helps in sharing knowledge about reasoning strategies or problem-solving methods. In a follow-up article, we will focus on method ontologies.

Ontology as vocabulary

In philosophy, ontology is the study of the kinds of things that exist. It is often said that ontologies “carve the world at its joints.” In AI, the term ontology has largely come to

mean one of two related things. First of all, ontology is a representation vocabulary, often specialized to some domain or subject matter. More precisely, it is not the vocabulary as such that qualifies as an ontology, but the conceptualizations that the terms in the vocabulary are intended to capture. Thus, translating the terms in an ontology from one language to another, for example from English to French, does not change the ontology conceptually. In engineering design, you might discuss the ontology of an electronic-devices domain, which might include vocabulary that describes conceptual elements—transistors, operational amplifiers, and voltages—and the relations between these elements—operational amplifiers are a *type-of* electronic device, and transistors are a *component-of* operational amplifiers. Identifying such vocabulary—and the underlying conceptualizations—generally

requires careful analysis of the kinds of objects and relations that can exist in the domain.

In its second sense, the term ontology is sometimes used to refer to a body of knowledge describing some domain, typically a commonsense knowledge domain, using a representation vocabulary. For example, CYC¹ often refers to its knowledge representation of some area of knowledge as its ontology.

In other words, the representation vocabulary provides a set of terms with which to describe the facts in some domain, while the body of knowledge using that vocabulary is a collection of facts about a domain. However, this distinction is not as clear as it might first appear. In the electronic-device example, that transistor is a *component-of* operational amplifier or that the latter is a *type-of* electronic device is just as much a fact about

THIS SURVEY PROVIDES A CONCEPTUAL INTRODUCTION TO ONTOLOGIES AND THEIR ROLE IN INFORMATION SYSTEMS AND AI. THE AUTHORS ALSO DISCUSS HOW ONTOLOGIES CLARIFY THE DOMAIN'S STRUCTURE OF KNOWLEDGE AND ENABLE KNOWLEDGE SHARING.

its domain as a CYC fact about some aspect of space, time, or numbers. The distinction is that the former emphasizes the use of ontology as a set of terms for representing specific facts in an instance of the domain, while the latter emphasizes the view of ontology as a general set of facts to be shared.

There continues to be inconsistencies in the usage of the term ontology. At times, theorists use the singular term to refer to a specific set of terms meant to describe the entity and relation-types in some domain. Thus, we might speak of *an ontology* for “liquids” or for “parts and wholes.” Here, the singular term stands for the entire set of concepts and terms needed to speak about phenomena involving liquids and parts and wholes.

When different theorists make different proposals for an ontology or when we speak about ontology proposals for different domains of knowledge, we would then use the plural term *ontologies* to refer to them collectively. In AI and information-systems literature, however, there seems to be inconsistency: sometimes we see references to “*ontology of domain*” and other times to “*ontologies of domain*,” both referring to the set of conceptualizations for the domain. The former is more consistent with the original (and current) usage in philosophy.

Ontology as content theory

The current interest in ontologies is the latest version of AI’s alternation of focus between content theories and mechanism theories. Sometimes, the AI community gets excited by some mechanism such as rule systems, frame languages, neural nets, fuzzy logic, constraint propagation, or unification. The mechanisms are proposed as the secret of making intelligent machines. At other times, we realize that, however wonderful the mechanism, it cannot do much without a good content theory of the domain on which it is to work. Moreover, we often recognize that once a good content theory is available, many different mechanisms might be used equally well to implement effective systems, all using essentially the same content.²

AI researchers have made several attempts to characterize the essence of what it means to have a content theory. McCarthy and Hayes’ theory (epistemic versus heuristic distinction),³ Marr’s three-level theory (information processing, strategy level, algorithms and data structures level, and physical mechanisms level),⁴ and Newell’s theory (Knowl-

edge Level versus Symbol Level)⁵ all grapple in their own ways with characterizing content. Ontologies are quintessentially content theories, because their main contribution is to identify specific classes of objects and relations that exist in some domain. Of course, content theories need a representation language. Thus far, predicate calculus-like formalisms, augmented with *type-of* relations (that can be used to induce class hierarchies), have been most often used to describe the ontologies themselves.

Why are ontologies important?

Ontological analysis clarifies the structure of knowledge. Given a domain, its ontology forms the heart of any system of knowledge representation for that domain. Without ontologies, or the conceptualizations that underlie knowledge, there cannot be a vocabulary for representing knowledge. Thus, the first step in devising an effective knowledge-representation system, and vocabulary, is to perform an effective ontological analysis of the field, or domain. Weak analyses lead to incoherent knowledge bases.

An example of why performing good analysis is necessary comes from the field of databases.⁶ Consider a domain having several classes of people (for example, students, professors, employees, females, and males). This study first examined the way this database would be commonly organized: students, employees, professors, males, and female would be represented as *types-of* the class *humans*. However, some of the problems that exist with this ontology are that students can also be employees at times and can also stop being students. Further analysis showed that the terms *students* and *employee* do not describe categories of humans, but are roles that humans can play, while terms such as *females* and *males* more appropriately represent subcategories of humans. Therefore, clarifying the terminology enables the ontology to work for coherent and cohesive reasoning purposes.

Second, ontologies enable knowledge sharing. Suppose we perform an analysis and arrive at a satisfactory set of conceptualizations, and their representative terms, for some area of knowledge—for example, the electronic-devices domain. The resulting ontology would likely include domain-specific

terms such as *transistors* and *diodes*; general terms such as *functions*, *causal processes*, and *modes*; and terms that describe behavior such as *voltage*. The ontology captures the intrinsic conceptual structure of the domain. In order to build a knowledge representation language based on the analysis, we need to associate terms with the concepts and relations in the ontology and devise a syntax for encoding knowledge in terms of the concepts and relations. We can share this knowledge representation language with others who have similar needs for knowledge representation in that domain, thereby eliminating the need for replicating the knowledge-analysis process. Shared ontologies can thus form the basis for domain-specific knowledge-representation languages. In contrast to the previous generation of knowledge-representation languages (such as KL-One), these languages are *content-rich*; they have a large number of terms that embody a complex content theory of the domain.

Shared ontologies let us build specific knowledge bases that describe specific situations. For example, different electronic-devices manufacturers can use a common vocabulary and syntax to build catalogs that describe their products. Then the manufacturers could share the catalogs and use them in automated design systems. This kind of sharing vastly increases the potential for knowledge reuse.

Describing the world

We can use the terms provided by the domain ontology to assert specific propositions about a domain or a situation in a domain. For example, in the electronic-device domain, we can represent a fact about a specific circuit: circuit 35 has transistor 22 as a component, where circuit 35 is an instance of the concept circuit and transistor 22 is an instance of the concept transistor. Once we have the basis for representing propositions, we can also represent knowledge involving propositional attitudes (such as *hypothesize*, *believe*, *expect*, *hope*, *desire*, and *fear*). Propositional attitude terms take propositions as arguments. Continuing with the electronic-device domain, we can assert, for example: the diagnostician *hypothesizes* or *believes* that part 2 is broken, or the designer *expects* or *desires* that the power plant has an output of 20 megawatts. Thus, an ontology can represent beliefs, goals,

On the one hand there are entities, such as processes and events, which have temporal parts... On the other hand there are entities, such as material objects, which are always present in their entirety at any time at which they exist at all. The categorical distinction between entities which do, and entities which do not have temporal parts is grounded in common sense. Yet various philosophers have been inclined to oppose it. Some ... have defended an ontology consisting exclusively of things with no temporal parts. Whiteheadians have favored ontologies including only temporally extended processes. Quine has endorsed a four-dimensional ontology in which the distinction between objects and processes vanishes and every entity comprises simply the content of some arbitrarily demarcated portion of space-time. One further option, embraced by philosophers such as David Lewis, accepts the opposition between objects and processes, while still finding a way to allow that all entities have both spatial and temporal parts.

Figure 1. Call for papers for a special issue on temporal parts for *The Monist, An International Quarterly Journal of General Philosophical Inquiry*. This quote suggests that ontology has always been an issue of deep concern in philosophy and that the issues continue to occupy contemporary philosophers.

hypotheses, and predictions about a domain, in addition to simple facts. The ontology also plays a role in describing such things as plans and activities, because these also require specification of world objects and relations. Propositional attitude terms are also part of a larger ontology of the world, useful especially in describing the activities and properties of the special class of objects in the world called “intensional entities”—for example, agents such as humans who have mental states.

Constructing ontologies is an ongoing research enterprise. Ontologies range in abstraction, from very general terms that form the foundation for knowledge representation in all domains, to terms that are restricted to specific knowledge domains. For example, *space*, *time*, *parts*, and *subparts* are terms that apply to all domains; *malfunction* applies to engineering or biological domains; and *hepatitis* applies only to medicine.

Even in cases where a task might seem to be quite domain-specific, knowledge representation might call for an ontology that describes knowledge at higher levels of generality. For example, solving problems in the domain of turbines might require knowledge expressed using domain-general terms such as *flows* and *causality*. Such general-level descriptive terms are called the *upper ontology* or *top-level ontology*. There are many open research issues about the correct ways to analyze knowledge at the upper level. To provide some idea of the issues involved, Figure 1 excerpts a quote from a recent call for papers.

Today, ontology has grown beyond philosophy and now has many connections to information technology. Thus, research on ontology in AI and information systems has had to produce pragmatically useful proposals for top-level ontology. The organization of a top-level ontology contains a number of problems, similar to the problems that surround ontol-

ogy in philosophy. For example, many ontologies have *thing* or *entity* as their root class. However, Figure 2 illustrates that thing and entity start to diverge at the next level.

For example, CYC’s *thing* has the subcategories *individual object*, *intangible*, and *represented thing*; the Generalized Upper Model’s⁷ (GUM) *um-thing* has the subcategories *configuration*, *element*, and *sequence*; Wordnet’s⁸ *thing* has the subcategories *living thing* and *nonliving thing*, and Sowa’s root *T* has the subcategories *concrete*, *process*, *object*, and *abstract*. (Natalya Fridman Noy’s and Carol Hafner’s article discusses these differences more fully.⁹) Some of these differences arise because not all of these ontologies are intended to be general-purpose tools, or even explicitly to be ontologies. Another reason for the differences is that, in principle, there are many different taxonomies.

Although differences exist within ontologies, general agreement exists between ontologies on many issues:

- There are *objects* in the world.
- Objects have *properties* or *attributes* that can take *values*.
- Objects can exist in various *relations* with each other.
- Properties and relations can change over *time*.
- There are *events* that occur at different *time instants*.
- There are *processes* in which objects participate and that occur over time.
- The world and its objects can be in different *states*.
- Events can *cause* other events or states as *effects*.
- Objects can have *parts*.

The representational repertoire of objects, relations, states, events, and processes does not say anything about which classes of these

entities exist. The modeler of the domains makes these commitments. As we move from an ontology’s top to lower taxonomic levels, commitments specific to domains and phenomena appear. For modeling objects on earth, we can make certain commitments. For example, *animals*, *minerals*, and *plants* are subcategories of objects; *has-life(x)* and *contains-carbon(x)* are object properties; and *can-eat(x, y)* is a possible relation between any two objects. These commitments are specific to objects and phenomena in this domain. Further, the commitments are not arbitrary. For them to be useful, they should reflect some underlying reality.

There is no sharp division between domain-independent and domain-specific ontologies for representing knowledge. For example, the terms *object*, *physical object*, *device*, *engine*, and *diesel engine* all describe objects, but in an order of increasing domain specificity. Similarly, terms for relations between objects can span a range of specificity, such as *connected*, *electrically-connected*, and *soldered-to*.

Subtypes of concepts. Ontologies generally appear as a taxonomic tree of conceptualizations, from very general and domain-independent at the top levels to increasingly domain-specific further down in the hierarchy. We mentioned earlier that different ontologies propose different subtypes of even very general concepts. This is because, as a rule, different sets of subcategories will result from different criteria for categorization. Two, among many, alternate subcategorizations of the general concept *object* are *physical* and *abstract*, and *living* and *non-living*. In some cultures and languages, words for objects have gender, thus creating another top-level classification along the gender axis. We can easily think of additional subcategorizations based on other criteria. The existence of alternate categorizations only becomes more acute as we begin to model specific domains of knowledge. For example, we can subcategorize causal process into *continuous* and *discrete* causal processes along the dimension of how time is represented, and into *mechanical*, *chemical*, *biological*, *cognitive*, and *social* processes along the dimension of the kinds of objects and relations involved in the description.

In principle, the number of classification criteria and distinct subtypes is unlimited, because the number of possible dimensions along which to develop subcategories can-

not be exhaustively specified. Often, this fact is not obvious in general-purpose ontologies, because the top levels of such ontologies commit to the most commonly useful subtypes. However, domain-specific ontologies can contain categorizations along dimensions that are usually outside the general ontology.

Task dependence of ontologies. How task-dependent are ontologies? Presumably, the kinds of things that actually exist do not depend on our goals. In that sense, ontologies are not task-dependent. On the other hand, what aspects of reality are chosen for encoding in an ontology does depend on the task. For example, in the domain of fruits, we would focus on particular aspects of reality if we were developing the ontology for the selection of pesticides; we would focus on other aspects of reality if we were developing an ontology to help chefs select fruits for cooking. In ontologies for engineering applications, categorizing causal processes into those that do, and that do not, produce dangerous side effects might be useful. Design engineers and safety analysts might find this a very useful categorization, though it is unlikely to be part of a general-purpose ontology's view of the causal process concept.

Practically speaking, an ontology is unlikely to cover all possible potential uses. In that sense, both an ontology for a domain and a knowledge base written using that ontology are likely to be more appropriate for certain uses than others and unlikely to be sharable across widely divergent tasks. This is, by now, a truism in KBS research and is the basic insight that led to the current focus on the relationship between tasks and knowledge types. Presuppositions or requirements can be associated with problem-solving methods for different tasks so that they can capture explicitly the way in which ontologies are task-dependent. For example, a method might have a presupposition (or assumption¹⁰) stating that it works correctly only if the ontology allows modeling causal processes discretely. Therefore, assumptions are a key factor in practical sharing of ontologies.

Technology for ontology sharing

There have been several recent attempts to create engineering frameworks for constructing ontologies. Michael R. Genesereth and

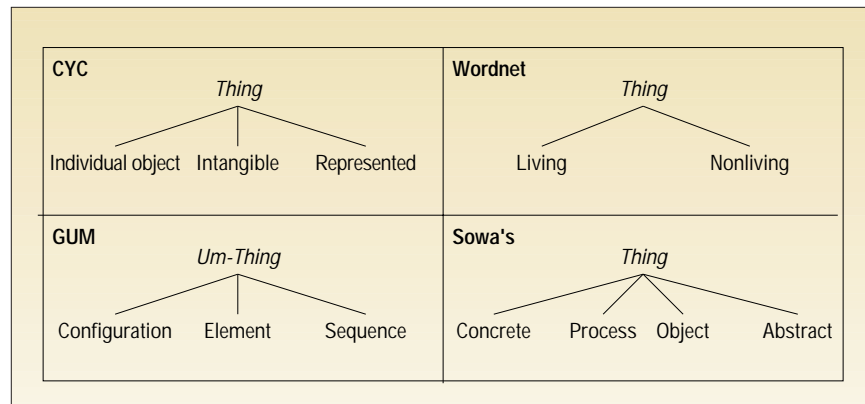


Figure 2. Illustration of how ontologies differ in their analyses of the most general concepts.

Richard E. Fikes describe KIF (Knowledge Interchange Format), an enabling technology that facilitates expressing domain factual knowledge using a formalism based on augmented *predicate calculus*.¹¹ Robert Neches and his colleagues describe a knowledge-sharing initiative,¹² while Thomas R. Gruber has proposed a language called *Ontolingua* to help construct portable ontologies.¹³ In Europe, the CommonKADS project has taken a similar approach to modeling domain knowledge.¹⁴

These languages use varieties of predicate calculus as the basic formalism. Predicate calculus facilitates the representation of objects, properties, and relations. Variations such as *situational calculus* introduce time so as to represent states, events, and processes. If we extend the idea of knowledge to include images and other sense modalities, we might need radically different kinds of representation. For now, predicate calculus provides a good starting point for ontology-sharing technologies.

Using a logical notation for writing and sharing ontologies does not imply any commitment to implementing a related knowledge system or a related logic. We are simply taking a knowledge-level⁵ stance in describing the knowledge system, whatever the means of implementation. In this view, we can ask of any intelligent system, even one implemented as a neural network, "What does the system know?"

Use of ontologies

In AI, knowledge in computer systems is thought of as something that is explicitly represented and operated on by inference processes. However, that is an overly narrow view. All information systems traffic in knowledge. Any software that does anything useful cannot be written without a commitment to a model of the relevant world—to entities, prop-

erties, and relations in that world. Data structures and procedures implicitly or explicitly make commitments to a domain ontology. It is common to ask whether a payroll system "knows" about the new tax law, or whether a database system "knows" about employee salaries. Information-retrieval systems, digital libraries, integration of heterogeneous information sources, and Internet search engines need domain ontologies to organize information and direct the search processes. For example, a search engine has categories and subcategories that help organize the search. The search-engine community commonly refers to these categories and subcategories as ontologies.

Object-oriented design of software systems similarly depends on an appropriate domain ontology. Objects, their attributes, and their procedures more or less mirror aspects of the domain that are relevant to the application. Object systems representing a useful analysis of a domain can often be reused for a different application program. Object systems and ontologies emphasize different aspects, but we anticipate that over time convergence between these technologies will increase. As information systems model large knowledge domains, domain ontologies will become as important in general software systems as in many areas of AI.

In AI, while knowledge representation pervades the entire field, two application areas in particular have depended on a rich body of knowledge. One of them is natural-language understanding. Ontologies are useful in NLU in two ways. First, domain knowledge often plays a crucial role in disambiguation. A well-designed domain ontology provides the basis for domain knowledge representation. In addition, ontology of a domain helps identify the semantic categories that are involved in understanding discourse in that domain. For this use, the ontology plays the role of a concept dictionary. In general, for NLU, we need

Related work

The field of ontology attracts an interdisciplinary mix of researchers, both from academia and industry. Here we give a selection of references that describe related ontology work. Because the literature is vast, a complete list is impossible. For an extensive collection of (alphabetically ordered) links to ontological work, including proceedings and events, see <http://www.cs.utexas.edu/users/mfkb/related.html>.

Special issues on ontology

- N. Guarino and R. Poli, "The Role of Ontology in the Information Technology," *Int'l J. Human-Computer Studies*, Vol. 43, Nos. 5/6, Nov.-Dec. 1995, pp. 623–965.
- G. Van Heijst, A.T. Schreiber, and B.J. Wielinga, "Using Explicit Ontologies in KBS Development," *Int'l J. Human-Computer Studies*, Vol. 46, Nos. 2/3, Feb.-Mar. 1997, pp. 183–292.
- M. Uschold and A. Tate, "Putting Ontologies to Use," *Knowledge Eng. Rev.*, Vol. 13, No. 1, Mar. 1998, pp. 1–3.

Ontology development

- J. Benjamin et al., "Ontology Construction for Technical Domains," *Proc. EKAW '96: European Knowledge Acquisition Workshop, Lecture Notes in Artificial Intelligence No. 1076*, Springer-Verlag, Berlin, 1996, pp. 98–114.
- W.N. Borst and J.M. Akkermans, "Engineering Ontologies," *Int'l J. Human-Computer Studies*, Vol. 46, Nos. 2/3, Feb.-Mar. 1997, pp. 365–406.
- A. Farquhar, R. Fikes, and J. Rice, "The Ontolingua Server: A Tool for Collaborative Ontology Construction," *Int'l J. Human-Computer Studies*, Vol. 46, No. 6, June 1997, pp. 707–728.
- A. Gomez-Perez, A. Fernandez, and M.D. Vicente, "Towards a Method to Conceptualize Domain Ontologies," *Working Notes 1996 European Conf. Artificial Intelligence (ECAI '96) Workshop on Ontological Eng.*, ECCAI, Budapest, Hungary, 1996, pp. 41–52.
- T.R. Gruber, "Towards Principles for the Design of Ontologies Used

for Knowledge Sharing," *Int'l J. Human-Computer Studies*, Vol. 43, Nos. 5/6, Nov.-Dec. 1995, pp. 907–928.

- R. Studer, V.R. Benjamins, and D. Fensel, "Knowledge Engineering, Principles, and Methods," *Data and Knowledge Eng.*, Vol. 25, Mar. 1998, pp. 161–197.
- M. Uschold and M. Gruninger, "Ontologies: Principles, Methods, and Applications," *Knowledge Eng. Rev.*, Vol. 11, No. 2, Mar. 1996, pp. 93–155.

Natural-language ontology

- J.A. Bateman, B. Magini, and F. Rinaldi, "The Generalized Upper Model," *Working Papers 1994 European Conf. Artificial Intelligence (ECAI '94) Workshop on Implemented Ontologies*, 1994, pp. 34–45; <http://www.darmstadt.gmd.de/publish/komet/papers/ecai94.ps>.
- K. Knight and S. Luk, "Building a Large-Scale Knowledge Base for Machine Translation," *Proc. AAAI '94*, AAAI Press, Menlo Park, Calif. 1994.
- G.A. Miller, "Wordnet: An Online Lexical Database," *Int'l J. Lexicography*, Vol. 3, No. 4, 1990, pp. 235–312.
- P.E. Van de Vet, P.H. Speel, and N.J.I. Mars, "The Plinius Ontology of Ceramic Materials," *Working Papers 1994 European Conf. Artificial Intelligence (ECAI '94) Workshop on Implemented Ontologies*, ECCAI, Amsterdam, 1994, pp. 187–206.

Ontologies and information sources

- Y. Arens et al., "Retrieving and Integrating Data from Multiple Information Sources," *Int'l J. Intelligent and Cooperative Information Systems*, Vol. 2, No. 2, 1993, pp. 127–158.
- S. Chawathe, H. Garcia-Molina, and J. Widom, "Flexible Constraint Management for Autonomous Distributed Databases," *IEEE Data Eng. Bulletin*, Vol. 17, No. 2, 1994, pp. 23–27.
- S. Decker et al., "Ontobroker: Ontology-Based Access to Distributed and Semi-Structured Information," *Semantic Issues in Multimedia Systems*, R. Meersman et al., eds., Kluwer Academic Publishers, Boston, 1999.

both a general-purpose upper ontology and a domain-specific ontology that focuses on the domain of discourse (such as military communications or business stories). CYC, Wordnet,⁸ and Sensus¹⁵ are examples of sharable ontologies that have been used for language understanding.

Knowledge-based problem solving is the second area in AI that is a big consumer of knowledge. KBPS systems solve a variety of problems—such as diagnosis, planning, and design—by using a rich body of knowledge. Currently, KBPS systems employ domain-specific knowledge, which is often sufficient for constructing knowledge systems that target specific application areas and tasks. However, even in specific application areas, knowledge systems can fail catastrophically when they are pushed to the edge of the capability of the domain-specific knowledge. In response to this particular shortcoming, researchers have proposed that problem-solving systems need commonsense knowl-

edge in addition to domain-specific knowledge. The initial motivation for CYC was to provide such a body of sharable commonsense knowledge for knowledge-based systems. There is a similar need for developing domain-specific knowledge. Thus, ontology-based knowledge-base development provides a double advantage. The ontologies themselves are sharable. With these ontologies, we can build knowledge bases using the structure of conceptualizations to encode specific pieces of knowledge. The knowledge bases that we develop using these ontologies can be shared more reliably, because the formal ontology that underlies them can help clarify the representation's semantics.

Information systems and NLU systems need factual knowledge about their domains of discourse. The inferences they make are usually simple. Problem-solving systems, in contrast, engage in complex sequences of inferences to achieve their goals. Such systems need to have reasoning strategies that

enable them to choose among alternative reasoning paths. Ontology specification in knowledge systems has two dimensions:

- *Domain factual knowledge* provides knowledge about the objective realities in the domain of interest (objects, relations, events, states, causal relations, and so forth).
- *Problem-solving knowledge* provides knowledge about how to achieve various goals. A piece of this knowledge might be in the form of a problem-solving method specifying—in a domain-independent manner—how to accomplish a class of goals.

Most early research in KBPS mixed factual and problem-solving knowledge into highly domain-specific rules, called *domain knowledge*. As research progressed, it became clear that there were systematic commonalities in reasoning strategies between

- S. Luke et al., "Ontology-Based Web Agents," *Proc. First Int'l Conf. Autonomous Agents*, ACM Press, New York, 1997, pp. 59–66; <http://www.cs.umd.edu/projects/plus/SHOE/> 1997.
- S.T. Polyak et al., "Applying the Process Interchange Format (PIF) to a Supply Chain Process Interoperability Scenario," *Proc. 1998 European Conf. Artificial Intelligence (ECAI '98) Workshop on Applications of Ontologies and Problem-Solving Methods*, ECCAI, Brighton, England, 1998, pp. 88–96.
- G. Wiederhold, "Intelligent Integration of Information," *J. Intelligent Information Systems*, Vol. 6, Nos. 2/3, 1996.
- G. Wiederhold and M. Genesereth, "The Conceptual Basis for Mediation Services," *IEEE Intelligent Systems*, Vol. 12, No. 5, Sept./Oct. 1997, pp. 38–47.

Ontologies and knowledge management

- A. Abecker et al., "Toward a Technology for Organizational Memories," *IEEE Intelligent Systems*, Vol. 13, No. 3, May/June 1998, pp. 40–48.
- V.R. Benjamins and D. Fensel, "The Ontological Engineering Initiative (KA)2," *Formal Ontology in Information Systems*, N. Guarino, ed., IOS Press, Amsterdam, 1998, pp. 287–301.
- M.S. Fox, J. Chionglo, and F. Fadel, "A Common-Sense Model of the Enterprise," *Proc. Industrial Eng. Research Conf.*, Inst. for Industrial Engineers, Norcross, Ga., 1993, pp. 425–429.
- *Manual of the Toronto Virtual Enterprise*, tech. report, Enterprise Integration Laboratory, Dept. of Industrial Eng., Univ. of Toronto, Toronto, 1995.
- M. Uschold et al., "The Enterprise Ontology," *Knowledge Eng. Rev.*, Vol. 13, No. 1, Mar. 1998.

Task and method ontologies

- D. Fensel et al., "Using Ontologies for Defining Tasks, Problem-Solving Methods, and Their Mappings," *Knowledge Acquisition, Modeling, and Management*, E. Plaza and V.R. Benjamins, eds., Springer-Verlag, Berlin, 1997, pp. 113–128.
- J.H. Gennari et al., "Mapping Domains to Methods in Support of

- Reuse," *Int'l J. Human-Computer Studies*, Vol. 41, No. 3, Sept. 1994, pp. 399–424.
- A. Tate, "Roots of SPAR—Shared Planning and Activity Representation," *Knowledge Eng. Rev.*, Vol. 13, No. 1, Mar. 1998, pp. 121–128.
- Y.A. Tijerino and R. Mizoguchi, "Multis II: Enabling End-Users to Design Problem-Solving Engines via Two-Level Task Ontologies," *Proc. EKAW '93: Seventh European Workshop on Knowledge Acquisition for Knowledge-Based Systems*, Lecture Notes in Artificial Intelligence No. 723, Springer-Verlag, 1993, pp. 340–359.

Ontology workshops

- Applications of Ontologies and Problem-Solving Methods, ECAI '98 (European Conf. AI), <http://delicias.dia.fi.upm.es/WORKSHOP/ECAI98/index.html>
- Building, Maintaining, and Using Organizational Memories, ECAI '98, <http://www.aifb.uni-karlsruhe.de/WBS/ECAI98OM/>
- Formal Ontologies in Information Systems (FOIS '98), <http://krr.irst.itc.it:1024/fois98/program.html>
- Intelligent Information Integration, ECAI '98, <http://www.tzi.de/grp/i3/ws-ecai98/>
- Sharable and Reusable Components for Knowledge Systems, KAW '98 (Workshop on Knowledge Acquisition, Modeling, and Management), <http://ksi.cpsc.ucalgary.ca/KAW/KAW98/KAW98Proc.html>
- Ontological Engineering, AAAI Spring Symp. Series, Stanford, Calif., 1997, <http://www.aaai.org/Symposia/Spring/1997/sss-97.html>
- Problem-Solving Methods, IJCAI '97 (Int'l Joint Conf. AI), <http://www.aifb.uni-karlsruhe.de/WBS/dfe/PSM/main.html>
- Ontological Engineering, ECAI '96, <http://www.wis.cs.utwente.nl:8080/kbs/EcaiWorkshop/homepage.html>
- Practical Aspects of Ontology Development, AAAI '96
- Sharable and Reusable Ontologies, KAW '96, <http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96Proc.html>
- Sharable and Reusable Problem-Solving Methods, KAW '96, <http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96Proc.html>

goals of similar types. These reasoning strategies were also characterized by their need for specific types of domain factual knowledge. It soon became clear that strategic knowledge could be abstracted and reused.

With few exceptions,^{16,17} the domain factual knowledge dimension drives the focus of most of the AI investigations on ontologies. This is because applications to language understanding motivates much of the work on ontologies. Even CYC, which was originally motivated by the need for knowledge systems to have world knowledge, has been tested more in natural-language than in knowledge-systems applications.

KBPS RESEARCHERS REALIZED that, in addition to factual knowledge, there is knowledge about how to achieve problem-solving goals. In fact, this emphasis on meth-

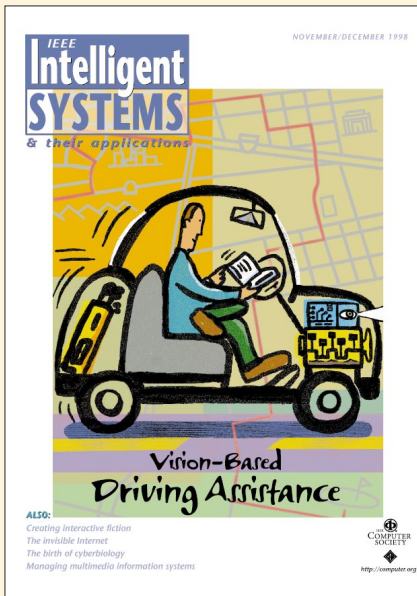
ods appropriate for different types of problems fueled second-generation research in knowledge systems.¹⁸ Most of the KBPS community's work on knowledge representation is not well-known to the general knowledge-representation community. In the coming years, we expect an increased focus on method ontologies as a sharable knowledge resource. ■

Acknowledgments

This article is based on work supported by the Office of Naval Research under Grant N00014-96-1-0701. We gratefully acknowledge the support of ONR and the DARPA RaDEO program. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of ONR. Netherlands Computer Science Research Foundation supported Richard Benjamins with financial support from the Netherlands Organization for Scientific Research (NWO).

References

1. D.B. Lenat and R.V. Guha, *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project*, Addison-Wesley, Reading, Mass., 1990.
2. B. Chandrasekaran, "AI, Knowledge, and the Quest for Smart Systems," *IEEE Expert*, Vol. 9, No. 6, Dec. 1994, pp. 2–6.
3. J. McCarthy and P.J. Hayes, "Some Philosophical Problems from the Standpoint of Artificial Intelligence," *Machine Intelligence Vol. 4*, B. Meltzer and D. Michie, eds., Edinburgh University Press, Edinburgh, 1969, pp. 463–502.
4. D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*, W.H. Freeman, San Francisco, 1982.
5. A. Newell, "The Knowledge Level," *Artificial Intelligence*, Vol. 18, 1982, pp. 87–127.
6. R. Wieringa and W. de Jonge, "Object Ident-



March/April Issue

Intelligent Agents

Guest Editor Jim Hendler of DARPA will present articles discussing development techniques for and the practical application of intelligent agents, which might be the solution to handling the data and information explosion brought about by the Internet. Scheduled topics include

- Agents for the masses: Is it possible to develop sophisticated agents simple enough to be practical?
- Extempo Systems' interactive characters, who engage, assist, and entertain people on the Web
- AgentSoft's efforts at commercializing intelligent agent technology

Intelligent Systems will also continue its coverage of the ontologies track, started in this issue, and the track on vision-based vehicle guidance, which began in the November/December 1998 issue.

IEEE Intelligent Systems covers the full range of intelligent system developments for the AI practitioner, researcher, educator, and user.

IEEE Intelligent Systems

tifiers, Keys and Surrogates: Object Identifiers Revisited," *Theory and Practice of Object Systems (TAPOS)*, Vol. 1, No. 2, 1995, pp. 101-114.

- J.A. Bateman, B. Magini, and F. Rinaldi, "The Generalized Upper Model," *Working Papers 1994 European Conf. Artificial Intelligence (ECAI '94) Workshop on Implemented Ontologies*, 1994, pp. 34-45; <http://www.darmstadt.gmd.de/publish/komet/papers/ecai94.ps>.
- G.A. Miller, "Wordnet: An Online Lexical Database," *Int'l J. Lexicography*, Vol. 3, No. 4, 1990, pp. 235-312.
- N. Fridman Noy and C.D. Hafner, "The State of the Art in Ontology Design," *AI Magazine*, Vol. 18, No. 3, 1997, pp. 53-74.
- D. Fensel and V.R. Benjamins, "The Role of Assumptions in Knowledge Engineering," *Int'l J. Intelligent Systems*, Vol. 13, No. 7, 1998, pp. 715-747.
- M.R. Genesereth and R.E. Fikes, *Knowledge Interchange Format, Version 0.3*, Knowledge Systems Lab., Stanford Univ., Stanford, Calif. 1992.
- R. Neches et al., "Enabling Technology for Knowledge Sharing," *AI Magazine*, Vol. 12, No. 3, 1991, pp. 36-56.
- T.R. Gruber, "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, Vol. 5, 1993, pp. 199-220.
- G. Schreiber et al., "CommonKADS: A Comprehensive Methodology for KBS Development," *IEEE Expert*, Vol. 9, No. 6,

Dec. 1994, pp. 28-37.

- K. Knight and S. Luk, "Building a Large-Scale Knowledge Base for Machine Translation," *Proc. Am. Assoc. Artificial Intelligence*, AAAI Press, Menlo Park, Calif., 1994.
- D. Fensel et al., "Using Ontologies for Defining Tasks, Problem-Solving Methods and Their Mappings," *Knowledge Acquisition, Modeling and Management*, E. Plaza and V.R. Benjamins, eds., Springer-Verlag, New York, 1997, pp. 113-128.
- R. Mizoguchi, J. Van Welkenhuysen, and M. Ikeda, "Task Ontology for Reuse of Problem Solving Knowledge," *Towards Very Large Knowledge Bases*, N.J.I. Mars, ed., IOS Press, Amsterdam, 1995.
- J.M. David, J.P. Krivine, and R. Simmons, *Second Generation Expert Systems*, Springer-Verlag, 1993.

B. Chandrasekaran is professor emeritus, a senior research scientist, and the director of the Laboratory for AI Research (LAIR) in the Department of Computer and Information Science at Ohio State University. His research focuses on knowledge-based systems, causal understanding, diagrammatic-reasoning, and cognitive architectures. He received his BE from Madras University and his PhD from the University of Pennsylvania, both in electrical engineering. He was Editor-in-Chief of *IEEE Expert* from 1990 to 1994, and he serves on the editorial boards of numerous international journals. He is a fellow of the IEEE, AAAI, and ACM. Contact him at the Laboratory for AI Research, Ohio State Univ., Columbus, OH,

43210; chandra@cis.ohio-state.edu; <http://www.cis.ohio-state.edu/lair/>.

John R. Josephson is a research scientist and the associate director of the Laboratory for AI Research in the Department of Computer and Information Science at Ohio State University. His primary research interests are knowledge-based systems, abductive inference, causal reasoning, theory formation, speech recognition, perception, diagnosis, the logic of investigation, and the foundations of science. He received his BS and MS in mathematics and his PhD in philosophy, all from Ohio State University. He has worked in several application domains, including medical diagnosis, medical test interpretation, diagnosis of engineered systems, logistics planning, speech recognition, molecular biology, design of electromechanical systems, and interpretation of aerial photographs. He is the coeditor with Susan Josephson of *Abductive Inference: Computation, Philosophy, Technology*, Cambridge Univ. Press, 1994. Contact him at the Laboratory for AI Research, Ohio State Univ., Columbus, OH, 43210; jj@cis.ohio-state.edu; <http://www.cis.ohio-state.edu/lair/>.

Richard Benjamins is a senior researcher and lecturer at the Department of Social Science Informatics at the University of Amsterdam. His research interests include knowledge engineering, problem-solving methods and ontologies, diagnosis and planning, and AI and the Web. He obtained his BS in cognitive psychology and his PhD in artificial intelligence from the University of Amsterdam. Contact him at the Dept. of Social Science Informatics, Univ. of Amsterdam, Roetersstraat 15, 1018 WB Amsterdam, The Netherlands; richard@swi.psy.uva.nl; <http://www.swi.psy.uva.nl/user/richard/home.html>.