

Ontologies for Corporate Web Applications

Leo Obrst, Howard Liu, and Robert Wray

■ In this article,¹ we discuss some issues that arise when ontologies are used to support corporate application domains such as electronic commerce (e-commerce) and some technical problems in deploying ontologies for real-world use. In particular, we focus on issues of ontology integration and the related problem of semantic mapping, that is, the mapping of ontologies and taxonomies to reference ontologies to preserve semantics. Along the way, we discuss what typically constitutes an ontology architecture. We situate the discussion in the domain of business-to-business (B2B) e-commerce. By its very nature, B2B e-commerce must try to interlink buyers and sellers from multiple companies with disparate product-description terminologies and meanings, thus serving as a paradigmatic case for the use of ontologies to support corporate applications.

The corporate world is poised to adopt the use of ontologies in web applications. Languages that allow the semantic annotation of information are becoming widely available; examples include the increasingly mainstream XML as well as newer languages such as the RESOURCE DESCRIPTION FRAMEWORK (RDF), the Defense Advanced Research Projects Agency (DARPA) AGENT MARKUP LANGUAGE + ONTOLOGY INFERENCE LAYER (DAML + OIL), and WEB ONTOLOGY LANGUAGE (OWL) motivated by the notion of a semantic web (Berners-Lee, Hendler, and Lassila 2001).^{2,3,4,5,6} Commercial organizations are seeking to codify web services using such formalizations as the universal description, discovery, and integration (UDDI) speci-

fication.⁷ There are efforts to standardize intelligent agent technology, such as the Foundation for Intelligent Physical Agents (FIPA).⁸ These efforts at standardization must use ontologies if emerging internet applications are to be powered by semantics, the meaning behind advanced applications and their enterprise-level and community-level transactions.

In this article, we discuss some issues that arise when ontologies are used to support corporate application domains such as electronic commerce (e-commerce) and some technical problems in deploying ontologies for real-world use. In particular, we focus on issues of ontology integration and the related problem of *semantic mapping*, that is, the mapping of ontologies and taxonomies to reference ontologies to preserve semantics. Along the way, we discuss what typically constitutes an ontology architecture and provide a short summary of ontology development tools. We situate the discussion in the domain of business-to-business (B2B) e-commerce. By its very nature, B2B e-commerce must try to interlink buyers and sellers from multiple companies with disparate product-description terminologies and meanings, thus serving as a paradigmatic case for the use of ontologies to support corporate applications.

Levels of Representation

The “vocabularies” for ontologies, as discussed in the introduction to this special issue, are distinct at different levels. Table 1 shows that there are a number of meta-object-level dis-

Level	Example Constructs
Knowledge Representation Language (ontology language): Metalevel to the Ontology Concept Level	Class, Relation, Instance, Function, Attribute, Property, Axiom
Ontology Concept Level: Object Level to the Knowledge Representation Language Metalevel to the Instance Level	Person, Location, Event, Medicine, Tractor, Food, and so on
Ontology Instance Level: Object Level to the Ontology Concept Level	John X. Smith, Harriet Beecher Stowe, Person 243904, CreditCardPurchaseTransactionEvent987102002, KubotaM120-4WD-98PTO-HP-V5-cylinder turbo-charged- direct-injection-diesel-engine

Table 1. Ontology Levels.

tinctions we need to make with respect to the languages. At the highest metalevel is the *knowledge representation language*, that is, the language one uses to model the ontology content at the underlying object level (the ontology concept level). Examples of a knowledge representation language include languages that preceded the semantic web, such as KL-ONE (Brachmand and Schmolze 1985), CLASSIC (Borgida and Patel-Schneider 1994), LOOM (MacGregor 1991), and other description logics; PROLOG and other logic programming and constraint logic languages (based on Horn clauses); KNOWLEDGE INTERCHANGE FORMAT (KIF) and its International Organization for Standardization (ISO) standard variant COMMON LOGIC;^{9,10} OPEN KNOWLEDGE BASE CONNECTIVITY LANGUAGE (OKBC) (Chaudrik et al. 1998); CYCL (the language that the CYC ontology is expressed in) (Guha and Lenat 1990);¹¹ and UNIFIED MODELING LANGUAGE (UML).¹² Semantic web languages include RDF/S,¹³ DAML + OIL,¹⁴ and OWL.¹⁵ At the second level, the ontology concept level, ontologies are defined using the constructs of the knowledge representation level. When you build an ontology, you define a specific concept (for example, the class (Person): *KR:CLASS(OC:Person)*).¹⁶ You probably specify that this class is a subclass of another class you've already defined, (Animal): *KR:SUBCLASS(OC:Person, OC:Animal)*; that is, person is a subclass of animal. At the lowest level, the ontology instance (OI) level, the constructs are instances of ontology concept-level constructs, for example, *KR:INSTANCE ((KR:CLASS OC:Person), OI:Person243904)*. We also observe that many times, the distinction between a concept and an instance is fuzzier than we might initially believe. For example, *Harriet Beecher Stowe* or a *KubotAM120...* tractor might be considered a concept, with specific occurrences or

references within texts or databases as instances—depending on your logical notion of an *individual*. A further complication, which is not described here, is to allow classes as instances (Welty 1998, 1995).

Obviously, the machine semantics is simple and inexpressive with respect to the complex, rich semantics of humans, but it's a beginning and very useful for our information systems. By designing and implementing a logical knowledge representation system and ontologies and getting the machine to make inferences that are extremely close to what humans would infer in comparable circumstances, we have imbued our systems with much more human-level semantic responses than they have at present, which is particularly true of B2B applications.

The Nature of the B2B Enterprise

B2B e-commerce is everything that land commerce is, and more: automated support for information and transaction flow and for vertical and horizontal commercial interoperability. B2B e-commerce includes the following: multiple marketplace platforms on the internet that support multiple trading models (auctions, reverse auctions, exchanges, request-for-proposal/request-for-quote (RFP/RFQ), bookstores, trading hubs, and so on) for and by commercial organizations, providing rich information content on products and services for both buyers and sellers (catalogs, product guides, market and domain editorial content, news, advertising) and support for buying and selling; financing; privacy-security; payment processing; order management; profiling-personalization; product configuration; planning-scheduling and forecasting; product life cycle and inventory management; business processes; work flow; and rules, logistics, distribution, and delivery.

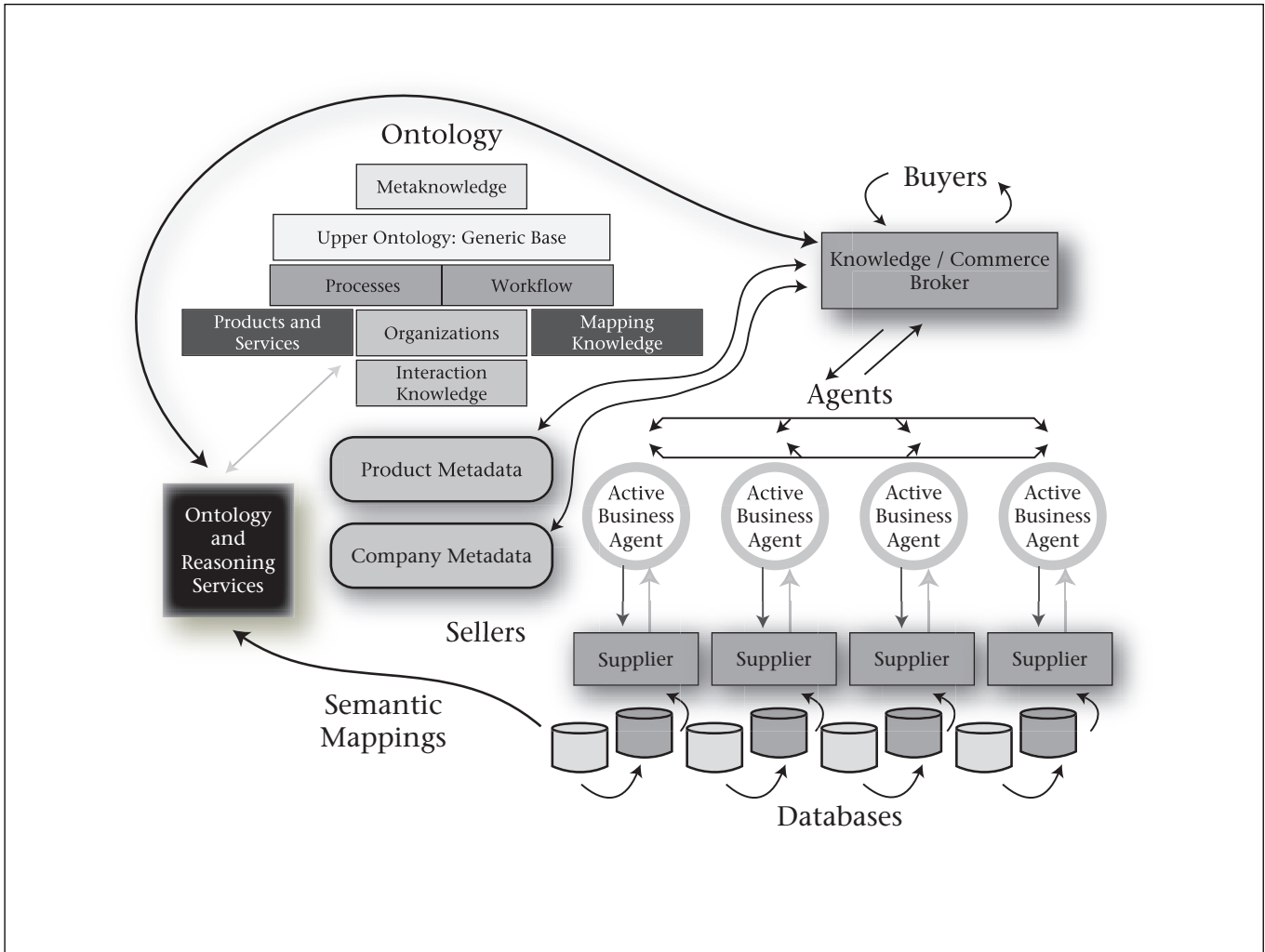


Figure 1. An E-Commerce Application Using Ontologies.

Why Ontologies Are Needed for B2B

B2B e-commerce needs ontologies. There are two primary uses for ontologies in B2B e-commerce. First, there is an informational need: Because the ontology is a structured conceptual model of the e-commerce vertical domain (and sometimes, quasihorizontal domains too), it supports parametric search and navigation using product and service knowledge by prospective buyers to discover what to buy and, subsequently, to determine pricing and availability. In this case, the fairly constant knowledge embodied in the ontology maps to the quickly changing data of the vendors. Furthermore, an ontology can model not only product and service knowledge but also knowledge about buyers and sellers, that is, users. By using user role knowledge (sometimes called user profiling or

personalization), for example, queries can be customized relative to the user's experience and interests.

Second, e-commerce also needs ontologies for transactional purposes: Knowledge of a company's organizational structure, work flow, processes, and products and services can be used to actually assist in buying and selling directly. For example, figure 1 depicts one view of an architecture and flow of knowledge within a prospective ontology-driven B2B marketplace infrastructure, linking buyers to semantically mapped suppliers using software agents or web-based service-oriented applications for both informational and transactional purposes. In this framework, multiple heterogeneous databases map to a common ontology that thus enables a meaningful comparative view to be displayed to a prospective buyer.

In this figure, an individual business product search agent (this could be a simpler, client ap-

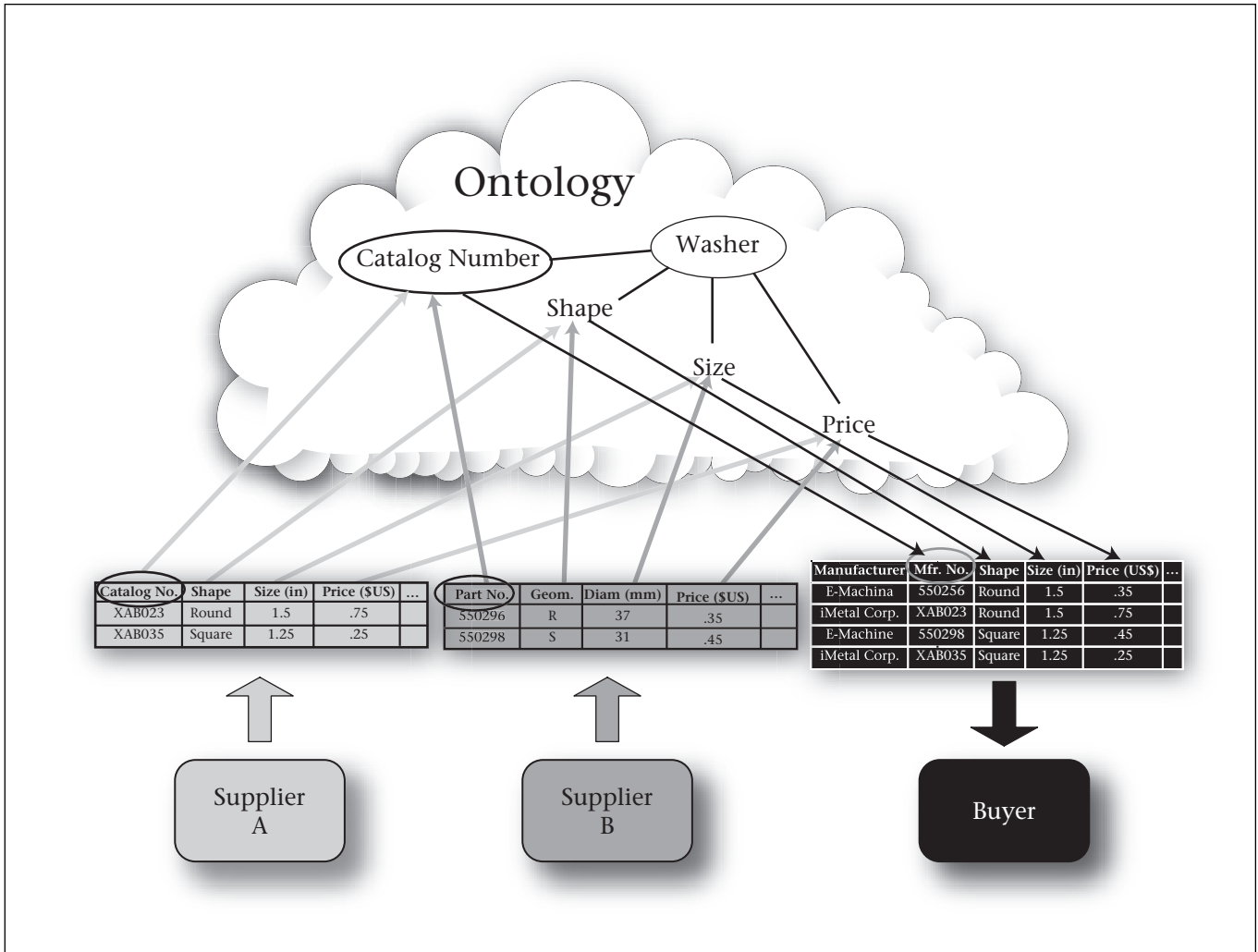


Figure 2. Buyers and Sellers Linked by Ontology.

plication) interacts with a given supplier; that is, it exists on the supplier’s site and is responsible for obtaining the data from this site. Each supplier has a database of products (the dark-gray cylinders) with its own format, structure, and semantics. The light-gray cylinders represent semantic mappings between the ontology and the specific supplier database.

These semantic mappings take the form of links between ontology concepts and their closest correlates in the supplier database, typically to specific tables, columns, and ranges of values represented by row entries in these columns, but potentially also include scripts to parse and extract specific values from coded values (attribute values are sometimes “packed” into product identifiers, for example, that then act as unique keys in the database) and then routines to transform the data representation into the canonical ontological form (example: meters to feet). The mappings are de-

termined in advance of run-time querying and typically require interaction among ontologists and supplier business analysts and database administrators to establish the semantics on both sides and, thereby, the appropriate semantic mappings. The supplier’s business agent uses the semantic mappings to send back relevant data about the supplier’s products to the knowledge-commerce broker-server application (it is both a broker and a server, although the two functions could be separated). The broker application interacts with prospective buyers, and it in turn knows about the product and service ontology. A prospective buyer searches for products and services by using the broker, either by querying or navigating a catalog structure. The broker then interprets the buyer’s query with respect to the ontology concepts and attributes and, using metadata about both the products and the companies, issues a distributed query to the appropriate business

agents. These agents then use their local mappings files to obtain the specific supplier data, apply the necessary transformations, and send it back to the broker. The broker then aggregates the data results and displays them in a normalized form to the prospective buyer, along with desired cost, availability, and distribution and shipping factors.

If the prospective buyer wants to actually buy specific products, these products are added to the “shopping cart.” At the end, the buyer can initiate a buying transaction, which triggers other ontological process transactions and inference that satisfy the constraints of the transaction and the individual buyer and supplier business-process requirements.

The use of ontologies in e-commerce thus goes a long way toward solving two problems: (1) the heterogeneous vendor database problem and (2) the standards and common vertical conceptual model problem. For the heterogeneous vendor database problem, distributors, manufacturers, and service providers have radically different databases that differ significantly in format, structure, and meaning, as in figure 2. The database community itself has studied this problem for some time (Kashyap and Sheth 1997, 1996; Mena et al. 1998; Smith and Obrst 1999; Wiederhold and Genesereth 1997). For catalog integration using a Bayesian approach, see Agrawal and Srikant (2001); for integrating vocabularies using a Bayesian approach, see Omelayenko (2002). For the standards and common vertical conceptual model problem, what is the meaning of the terminology used in the product and service space and the relationships between terms? That is, what are the concepts underpinning common business terminology, and can this meaning be made sound, consistent, extensible, reusable, modular, and logical?

Ontologies need to be built to support the representation requirements of many applications, all of which presume some form of classification of products and services. Many business classification systems are ad hoc, inconsistent, and nonintegrated, with very little association between classification systems. The distinction must be made between representation and presentation for corporate customers. *Representation* is the underlying structure and codification of the product and service knowledge space to be supplied by the eventually developed ontologies. This representation should be semantically sound, consistent (although incomplete in the sense that additional refinement could always be made), controlled, modular, and reusable and provide some support for application presentation needs.

Presentation, however, is largely the responsibility of an application. Some applications could choose to use their own terminology and classification display, as long as the terminology and structure have linkages to the underlying representation. An application intended for a buyer, for example, might display different terminology that is differently structured than an application intended for a seller. Furthermore, even within a buyer application, the terminology and displayed structure could be different based on the role of the prospective buyer-user. For example, a technically savvy engineer using a catalog search application would typically use search terminology (or, equivalently, navigation through a classification system or taxonomy) based on technical specifications. Within the ontology, this terminology would be entity centric and use entity-centric concepts. An entity in this usage is typically a thing, that is, a product object. However, a nontechnical purchasing analyst would typically use search terminology based on his/her own company’s environment or processes. Within the ontology, the terminology would thus use process- or function-centric representation. For example, although a chemist might search for a particular chemical by using terminology about chemical properties or formulas, a paint buyer for an automobile or aircraft manufacturer would search using notions of heat resistance, rust inhibition, and color. In either case, however, the navigational path used (by relational links or inference) should arrive at the identical, parameterized product or service. Ontologies would thus at least partially support multiple ways of presenting the product and service information. Typically, in an ontological approach, one could have separate “views” or “contexts” (we prefer the word *contexts* to distinguish it from the normal database view) that would explicitly be representable as a separate ontology or theory (we use these terms interchangeably) that just “uses” concepts represented in other ontologies, that is, in other portions of the entire ontological space. In fact, our discussion later on about the mapping problem is relevant to this notion, too, because it really is a semantic mapping established between the representation and the presentation. The presentation can be arbitrary application-specific terms, but they still need to be “represented” either internally in the underlying ontology or externally in some associational data structure that has a mapping to the ontology.

Figure 3 displays a prospective notional ontology architecture that displays the relationships among portions of the whole ontological

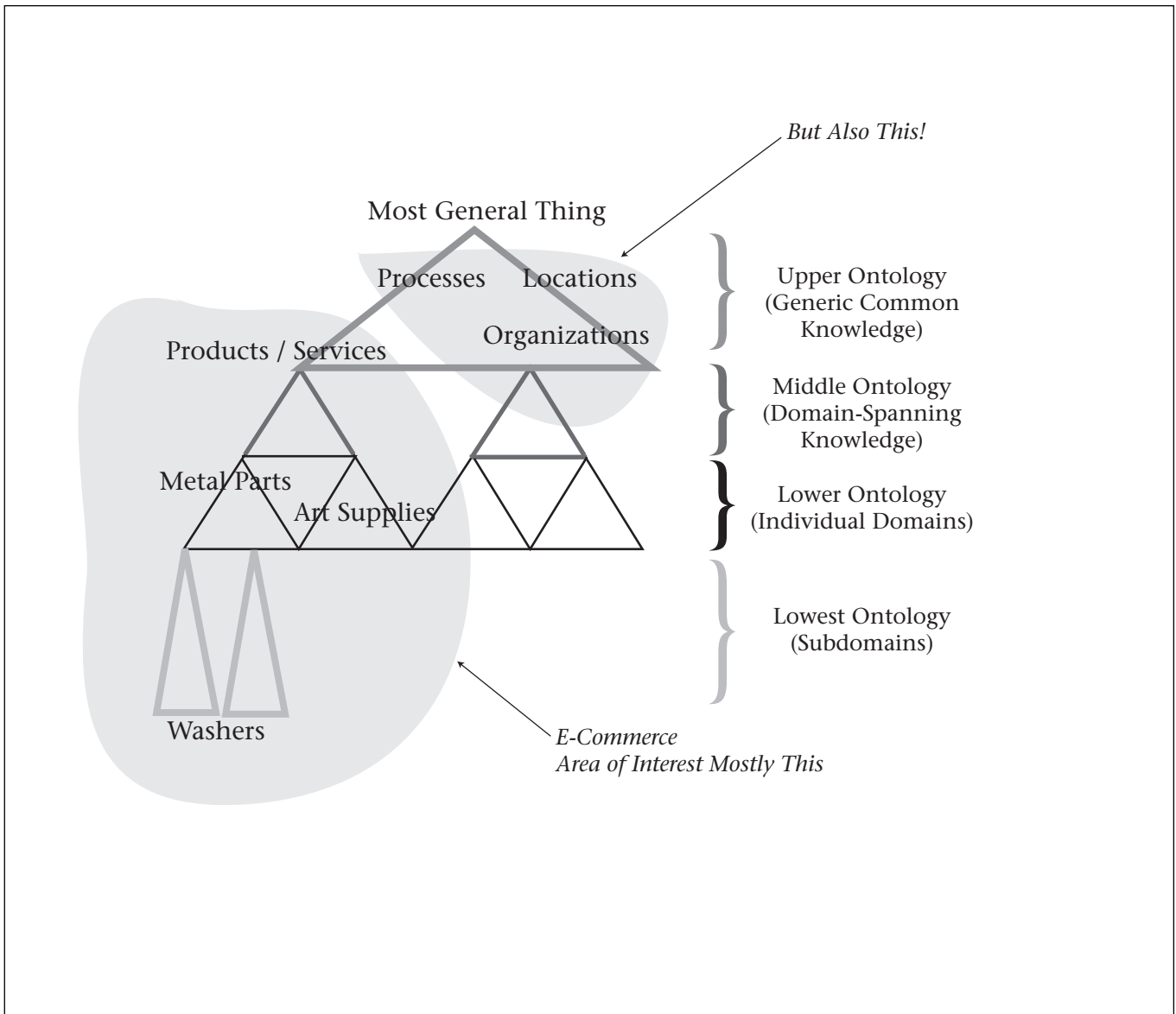


Figure 3. Upper, Middle, and Lower Ontologies.

space. The highlighted areas of the ontology are those usually deemed most appropriate for B2B e-commerce and include both products and processes. However, this architecture is only a simple educational device; the reality is much more complicated. Of course, it soon becomes evident when working as an ontologist in the product and service space that everything becomes relevant, from the upper-ontology notions (Guarino and Welty 2002) of part and whole (mereology),¹⁷ notions of necessity and sufficiency of properties, distinguishability, change of properties, task, and process concepts, to so-called middle-ontology notions of product and service, to domain-ontology lev-

els. Nearly everything can be considered a domain ontology in the experience-induced general view practical ontologists end up with. Everything is a theory (ontology), and every theory (ontology) can possibly be related to every other.

The vision, of course, for using a common representation is to enable a consistent ontological or conceptual search across data and applications, so that semantically meaningful documents and data concerning products and services are returned to the user. This search, by definition, included the notion of parametric search, which is related to the notion of product configuration, that is, a search informed by

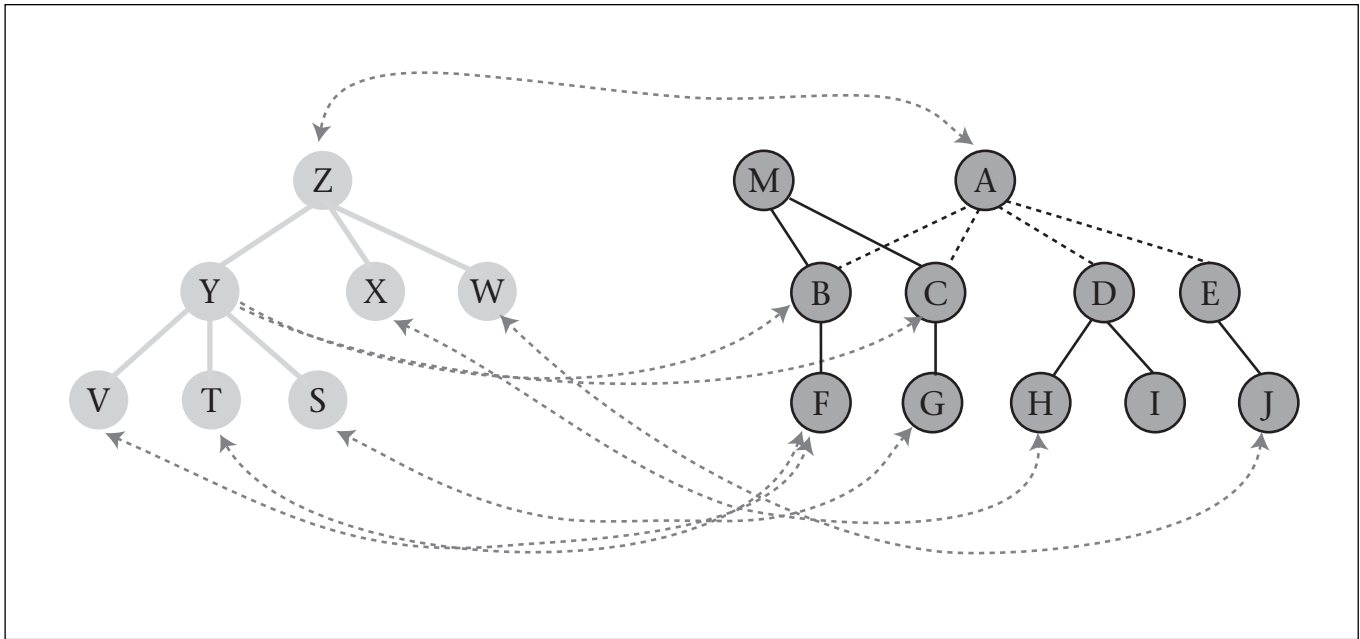


Figure 4. A Simple, Informal Application Taxonomy (left) Mapped to an Ontology (right).

ontological and other properties of the searched-for product or service. Correspondingly, a common representation supports ontological classification of products and services: search assisting primarily buyers, product classification assisting primarily sellers.

Ontologists, Domain Experts, and Ontology Architecture

Domain experts provide the knowledge for the ontologies. Ontologists determine how to represent this knowledge. Ontologists usually teach domain experts some of the fundamental concepts about ontologies and ontological engineering, along with how to use ontology editing tools. In an intensive, time-limited B2B development effort, ontologists and domain experts can begin working simultaneously, with all team members acting productively as quickly as possible, with minimal time for planning. Under these circumstances, as soon as domain experts begin creating domain ontologies unassisted, ontologists can shift their responsibilities to tasks optimally done in advance of ontology development: formulating designs for an overall ontology architecture, setting guidelines for building ontologies, integrating the ontologies with applications, and enhancing the ontology-building environment to make the ontologies more expressive and more maintainable. Ontology development can be considered a type of software de-

velopment and thus should follow the general principles of the software project life cycle. Emerging ontological methodologies (Fernández, Gómez-Pérez, and Juristo 1997; Gruninger and Fox 1995); Uschold and King 1995) should be used.

Referring back to figure 3, we see the typical organization of an ontology architecture. This architecture applies not only to B2B e-commerce but also more generally to any ontology-supported application area needing rich semantics.

The upper ontology (or set of integrated ontologies) is the layer that tries to characterize very basic commonsense knowledge notions. This layer makes distinctions between tangible object and intangible object, real geophysical terrain and map, the different relations of part-of (mereology) and connected-to (topology), and notions of space and time (point or interval based, dense, branching, and so on). There is a current effort under the Institute of Electrical and Electronics Engineers (IEEE) to standardize an upper ontology(ies) that would then be freely available to any ontological engineer seeking to build domain ontologies.¹⁸ The middle ontology layer represents knowledge that spans domains but possibly not every domain. An example is information technology, which might not apply to a domain such as marsupials but might apply to the domain of ethology (animal behavior) or ecology. The lower ontology consists of domains and sub-domains, for example, medicine and pediatric medicine.

The Mapping Problem

In this section, we briefly discuss one crucial issue in developing ontologies to support B2B e-commerce: the problem of mapping reference ontologies with well-defined semantics to other ontologies, taxonomies, and standards-based classification systems that are less semantically sound and coherent. The semantic mapping problem has a long and growing history because it is a difficult problem across many communities: the database, the thesaurus, and the ontology communities (Doerr 2000; Falkenhainer, Forbus, and Gentner 1990; Haas et al. 1999; Mitra, Wiederhold, and Kersten 2000; Musen and Noy 2002; Omelayenko 2002]; Rahm and Bernstein 2001). One IEEE standard upper ontology (SUO) candidate, the information flow framework,¹⁹ based on Barwise and Seligman's (2000) information flow theory, in fact is an elaborate metaontology to facilitate ontology integration in a very general way based on category theory.

Ideally, ontologies provide a semantic infrastructure that can be used for all applications. To provide this semantic framework across the applications and data of an e-commerce business (independently developed and without commitment to ontological infrastructure), existing and planned information resources must be connected to the ontology framework. We term this connection a *mapping*. A mapping is a many-to-many relationship between source data and an ontology. Sources for mapping could be another ontology, some standard taxonomy, or an application's data structures.

Figure 4 illustrates a mapping. Solid lines indicate a well-defined subclass relation in the ontology (right) and the taxonomy (left); the solid line represents a parent-child relation with ill-defined semantics. The heavy, dotted lines without arrows represent other ontological relations. The thin, double-headed lines depict a mapping between nodes in the ontology and data structures in the application taxonomy.

On the left, an e-commerce application uses a taxonomy with ill-defined semantics to represent some information. For example, node *Z* could represent some industrial process. Node *Y* could represent products resulting from this process, *X* the equipment used in the process, and *W* the employees involved in the process. The relation between nodes is an undefined parent-child relation with no inheritance; the information is used in the application to group these related concepts together. On the right, an ontology represents much of the same information but with a well-defined semantics for each relation. For example, nodes *B* and *C* could represent products resulting from the in-

dustrial process *A*. The relation between *A* and these nodes is not subclass; it might be "generated-from," assuming the semantics of this relation was defined. Because *B* and *C* are products, they are subclasses of a more general product node (*M*), perhaps in a middle or upper ontology. Unlike the application taxonomy, a different, well-defined relation relates *A* to *D*, and so on.

One criticism of this scenario is that the application taxonomy is deficient. In other words, what is needed is not a mapping but a clearly defined semantics in the application taxonomy. With such a definition, some automated merging process could (conceivably) merge the nodes. However, practically, such merging is not possible.

Mappings provide an intermediate solution to this problem. Once *Z* is mapped to the ontology, other applications can recognize that *Z* is similar to *A*. This example is only one of several different kinds of mappings typically encountered when attempting to use ontologies to support applications. The following sections explore three different kinds of mappings: (1) taxonomic standard to ontology, (2) application to ontology, and (3) ontology to ontology. We consider the different types of mapping separately because the more well defined the semantics of the source to be mapped to the ontology, the more straightforward the mapping process and the more semantically rich and powerful the resulting mapping.

Taxonomic Standard to Ontology Mapping

An e-commerce company will use open standards whenever possible simply to facilitate interaction with other companies. A case in point is the use of the Universal Standard Products and Services Classification (UNSPSC).²⁰ The UNSPSC hierarchy includes some 11,000 codes representing a taxonomic structure given by segment, family, class, and commodity. For example, segment 32 is "Electronic Components and Supplies;" family 3210 is "Printed circuits and integrated circuits and microassemblies."

Because no attributes are currently (which is changing) associated with nodes in the UNSPSC, one has to develop attributes. Furthermore, e-commerce typically needs substructure below the lowest level of the UNSPSC (commodity level) to specify products and their properties in some detail. Because the UNSPSC was originally developed not to reflect a buyer or seller's perspective but to permit financial roll up for accounting purposes, the taxonomy is inade-

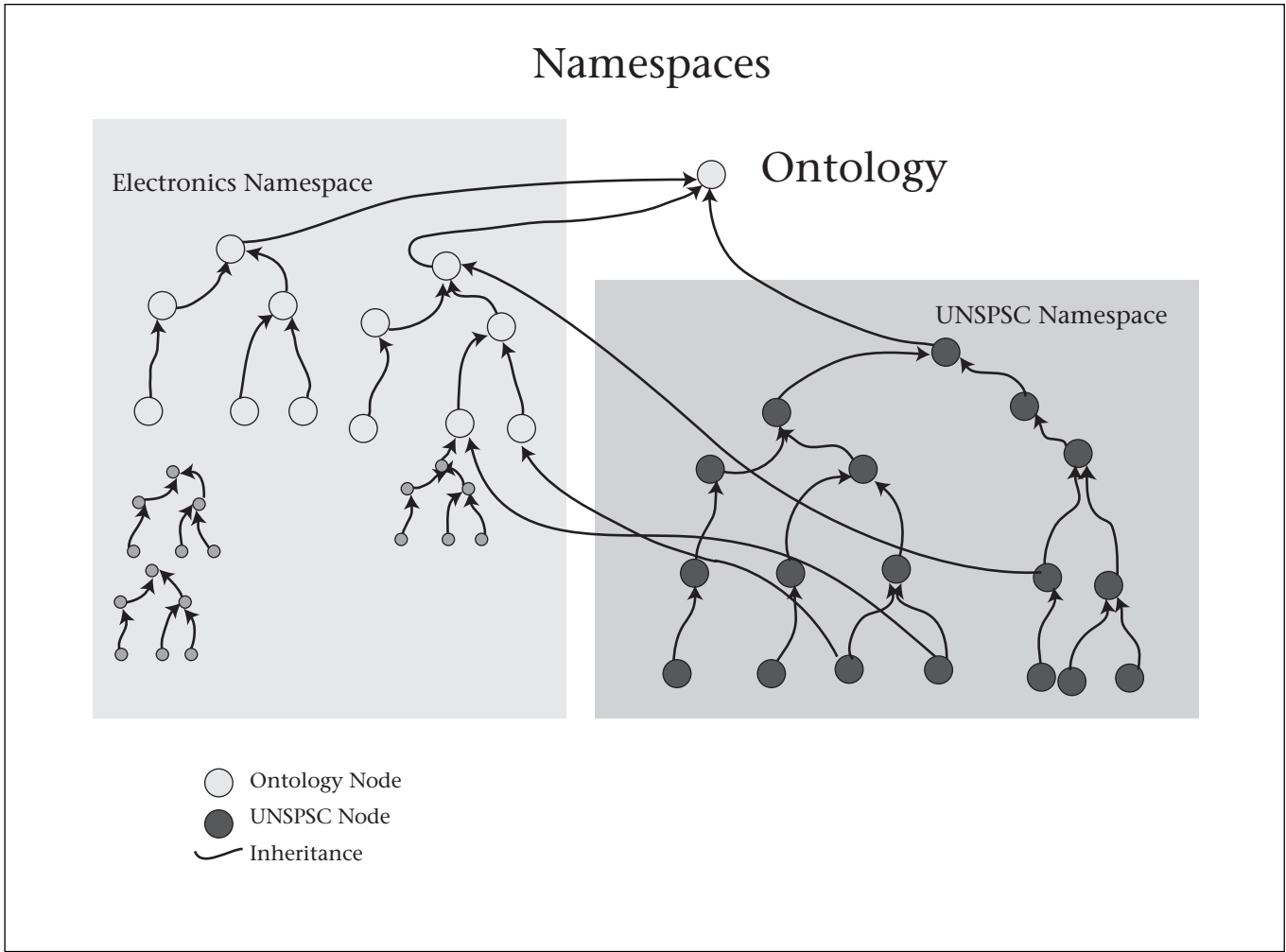


Figure 5. UNSPSC Mapped to Electronics Domain Ontology.

quately defined semantically for ontology purposes. Still, one wants to map to the standard to support companies that adhere to it. One solution is to use *Nebenstruktur* (shadow structure), which links the more semantically well-defined reference ontology (developed from an industry supplier and buyer perspective) to the UNSPSC (figure 5). These links can be provided by overloading subclass inheritance, thus allowing the inspection or extraction of a purely UNSPSC-based taxonomy with attached attributes and substructure. Attributes can be defined for classes in the reference ontology. By having classes (or subclasses defined below the bottom-most nodes) in the UNSPSC ontology be subclasses of the reference ontology classes, the reference attributes are inherited down to the UNSPSC nodes. One variation on this arrangement is to “type” the relations used in the entire ontological space with distinct, transitive subclass relations for every nonreference ontology supported and, thus, mapped to.

Application to Ontology Mapping

Any application data might need to be mapped to an ontology. Before considering how to approach such mappings, the goals for creating the mappings and their use should be identified.

Representational adequacy: A mapping solution should represent the mapping between the source and ontology completely and consistently. Completeness implies that any information in the application that could be queried by an external application has been mapped. Consistency implies that the most appropriate node in the ontology is mapped to the external structure. In some cases, this requirement means creating new ontology nodes. Thus, there is a tension between completeness and consistency that must be resolved by the methodology for mapping.

Heuristic adequacy: A mapping solution

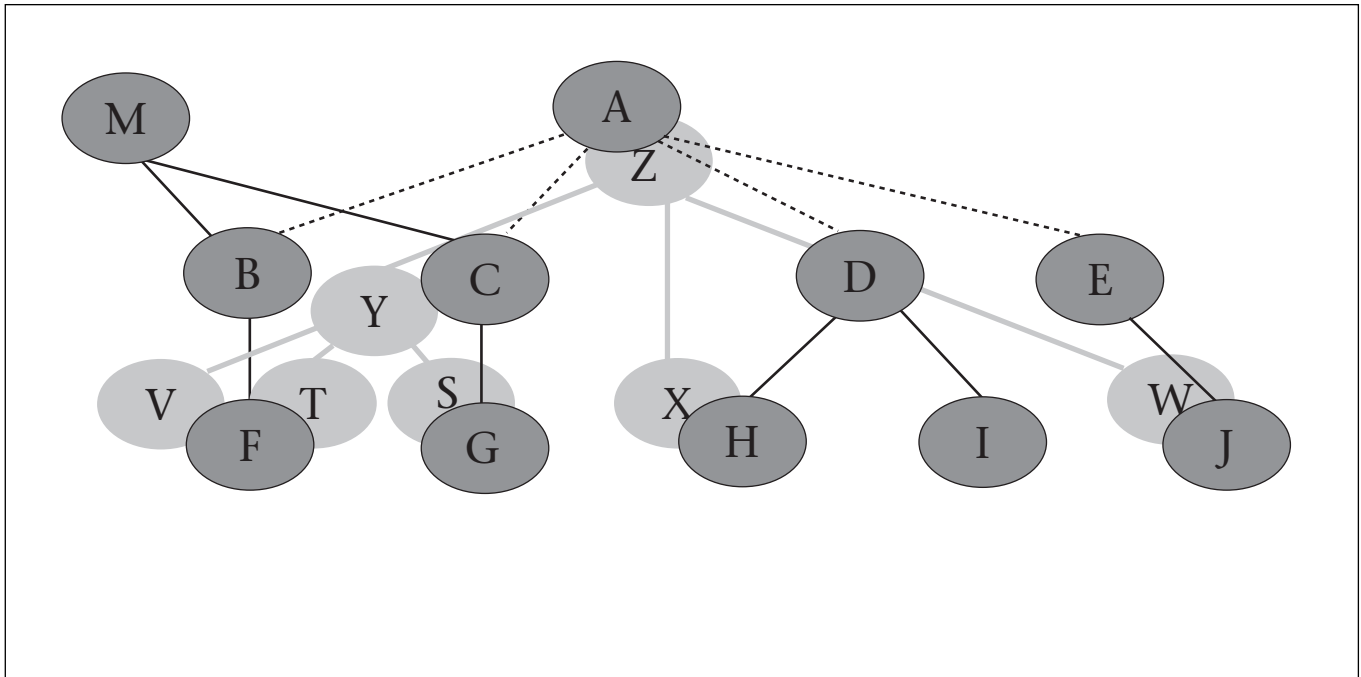


Figure 6. A Model of an Application in the Ontology.

must be relatively easy to create and convenient to access. Because the mappings will be *dynamic* (application data and ontologies will be changing over time), an individual mapping should not require significant computation or user interaction.

One representation: A mapping solution should not introduce duplication in representation.

Knowledge engineering cost-automation support: A mapping solution should not increase the work load or day-to-day activities of ontologists and domain experts. Thus, the solution should provide support to automate mappings. Because not all mappings can be automated, an infrastructure for mapping should be developed that will allow nondomain or nonontology experts to create mappings that cannot be done automatically.

Given these goals, how should the mappings between an application and an ontology be implemented? We consider three possibilities: (1) the mapping table, (2) the model of application, and (3) the applications mapped within the ontology.

Mapping table: A simple way to establish a mapping would be to create a table (for example, in a database) in which each row of the table indicates an association between some application data structure and the ontology. The table could support many-to-many mappings by duplicating either an application structure or an ontology node in some rows.

Completeness can easily be determined by ensuring that each application structure in the table is mapped to at least one ontology node.

Model of application: Figure 6 shows an example of building a model of the application in an ontology. All ontology-relevant application structures are included in the ontology model. A mapping relation is introduced that defines the association between an ontology node and nodes in the application model. In this approach, there is an ontological concept such as *claw hammer*, an application-specific concept *hammer as used in carpenter's catalog XYZ*, and a mapping relation that conveys the information that the “hammer as used in Catalog XYZ” corresponds to the claw hammer node in the ontology. The subclass relation can be used to represent the relations in the application ontology because the application concepts in the application taxonomy (rather than in the ontology proper) are subclasses in the context of the application. That is, *hammer* in the XZC Catalog might be a direct subclass of *tool* in the same catalog, even though in ontological space, there is much more distance between these concepts.

Application mapped within the ontology: Figure 7 illustrates another mapping solution. In this case, an application-specific mapping relation relates nodes in the ontology to their use in an application. The claw hammer node in the ontology is now not only related to other nodes in the domain ontology using domain

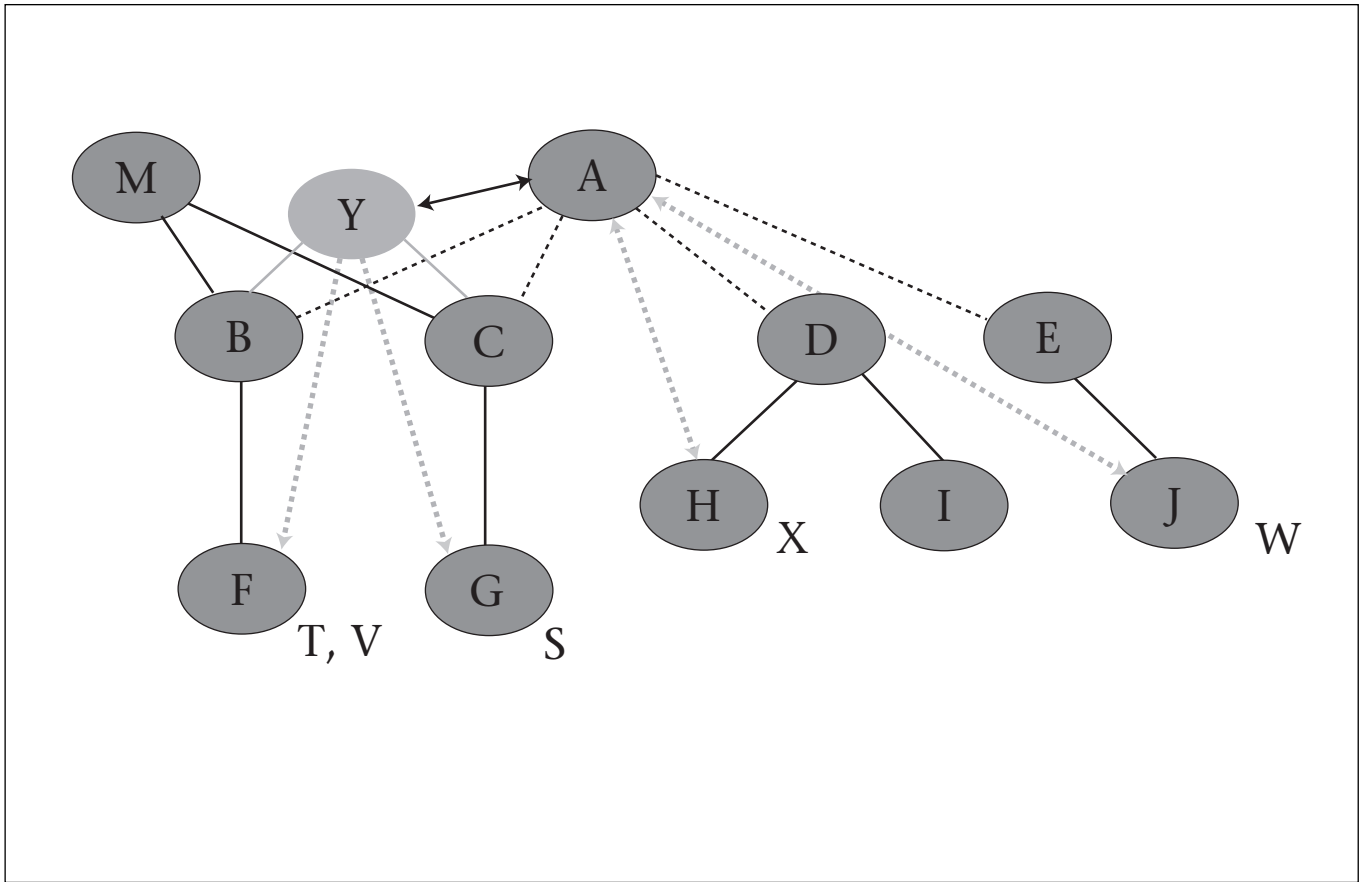


Figure 7. Mapping an Application in the Ontology.

relations (for example, *claw hammer* is a subclass of *hammer*) but also to nodes using application relations (*claw hammer* is a type of product in carpenter catalog XYC).

A complete analysis of these approaches is beyond the scope of this article and would include notions of ontology mapping and merging and context. The mapping table solution is simple to implement but lacks powerful connections to the ontology that could potentially help automate the maintenance of mappings as the ontologies and applications change. The increasing power of the other two approaches places demands on the representation of the mappings: Because the ontologies themselves are being changed, ontologists will need to be closely involved in the representation of the models and mappings, in addition to defining mapping relations.

Also, formal application requirements are required to determine how powerful the resulting mappings need to be. For simple interlingual (that is, as an intermediate language) use of the mappings, the mapping table appears to be a sufficient solution. The other approaches have higher initial costs but perhaps can be

maintained more easily and certainly will provide a more powerful substrate for reasoning because both the structure of the application, as well as its data, is represented (explicitly in the model approach, implicitly in the internal approach).

Conclusion

We presented some issues in the development of ontologies for use by corporate web applications and in particular have focused on the example of B2B e-commerce, where the ontologies that are developed will typically be in the product and service space. We discussed the typical ontology architecture. We also looked at one problem in particular, the ontology or semantic mapping problem, and investigated some possible solutions. This problem will always need to be resolved by these developing ontologies to support corporate applications, given the disparate nature of these applications in the real world. Despite these problems, however, we remain convinced that the increased use of ontologies in the corporate world is inevitable. Given the increasingly complex re-

quirements of applications, the need for rich, consistent, and reusable semantics, the growth of semantically interoperable enterprises into knowledge-based communities, and the evolution and adoption of semantic web technologies, ontologies represent the best answer to the demand for intelligent systems that operate closer to the human conceptual level.

Acknowledgments

We would like to express our appreciation for the encouragement and support we have received from many individuals, including our colleagues Pat Cassidy, Lisa Colvin, Don McKay, Mike Molloy, Jon Pastor, Eric Peterson, Lori Wilson, Yao Zhu, Craig Schlenoff, Ugo Boggio, Randy Clark, Frank Manshedi, Ashwin Parikh, Keith Thurston, Joe Whelan, and Christy Obrst as well as the anonymous reviewers. We also note that the views expressed in this article are those of the authors alone and do not reflect the official policy or position of The MITRE Corporation or any other company or individual.

Notes

1. An earlier version of this article appeared as Obrst, L.; Wray, R.; and Liu, H. Ontological Engineering for B2B E-Commerce. In Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS-2001), 117–126. New York: Association of Computing Machinery.
2. www.w3.org/TR/REC-xml.
3. www.w3.org/RDF/.
4. www.daml.org/2001/03/reference.html.
5. www.w3.org/2001/sw/WebOnt/.
6. OIL. www.ontoknowledge.org/oil/.
7. www.uddi.org/.
8. FIPA. www.fipa.org/.
9. NCITS.T2/98-004: logic.stanford.edu/kif/dpans.html. See also cl.tamu.edu/discuss/kif-100101.pdf.
10. cl.tamu.edu/.
11. www.cyc.com/tech.html.
12. UML. www.rational.com/uml/resources/documentation/index.jsp.
13. www.w3.org/RDF/.
14. www.daml.org/2001/03/reference.html.
15. www.w3.org/2001/sw/WebOnt/.
16. We use this quasipredicate notation for the next few examples, with prefixes indicating the construct level without formally defining it, that is, Level:Predicate(Level:Arg1, Level:Arg2).
17. suo.ieee.org/refs.html.
18. suo.ieee.org/refs.html.
19. suo.ieee.org/IFF/.
20. www.eccma.org/unspsc/.

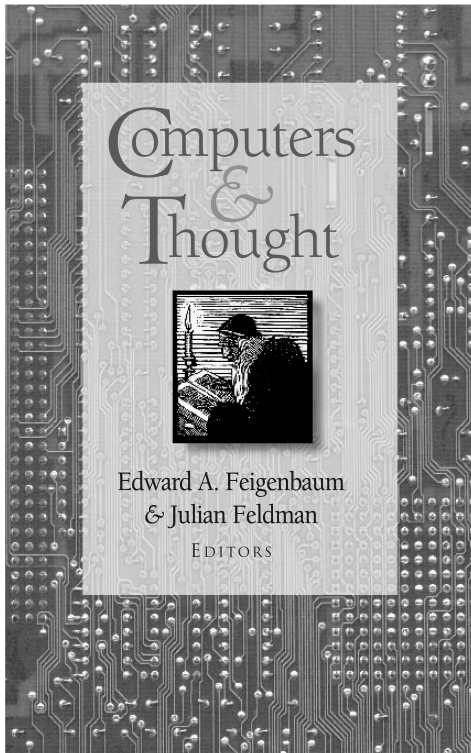
References

- Agrawal, R., and Srikant, R. 2001. On Integrating Catalogs. In Proceedings of the Tenth International World Wide Web Conference, 603–612. New York: Association of Computing Machinery.
- Barwise, J., and Seligman, J. 1997. *Information Flow: The Logic of Distributed Systems*. Cambridge, U.K.: Cambridge University Press.
- Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The Semantic Web. *The Scientific American* 284(5): 34–43.
- Borgida, A., and Patel-Schneider, P. 1994. A Semantics and Complete Algorithm for Subsumption in the CLASSIC Description Logic. *Journal of Artificial Intelligence Research* 1(1): 277–308.
- Brachman, R., and Schmolze, J. 1985. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science* 9(2): 171–216.
- Chaudri, V.; Farquhar, A.; Fikes, R.; Karp, P. D.; and Rice, J. P. 1998. Open Knowledge Base Connectivity Specification. Specification V. 2.0.31, SRI and Knowledge Systems Laboratory, Stanford University.
- Cohen, P.; Schrag, R.; Jones, E.; Pease, A.; Lin, B.; Starr, D.; Easter, D.; Gunning, D.; and Burke, M. 1998. The DARPA High-Performance Knowledge Bases Project. *Artificial Intelligence* 19(4): 25–49.
- Crampes, M., and Ranwez, S. 2000. Ontology-Supported and Ontology-Driven Conceptual Navigation on the World Wide Web. In Proceedings of the Eleventh Association of Computing Machinery Conference on Hypertext and Hypermedia, 191–199. New York: Association of Computing Machinery.
- Doan, A.; Mahavan, J.; Domingos, P.; and Halevy, A. 2002. Learning to Map between Ontologies on the Semantic Web. Paper presented at the Eleventh International Conference on the World Wide Web, 7–11 May, Honolulu, Hawaii.
- Doerr, M. 2000. Semantic Problems of Thesaurus Mapping. *Journal of Digital Information* 1(8).
- Falkenhainer, B.; Forbus, K.; and Gentner, D. 1990. The Structure Mapping Engine: Algorithm and Examples. *Artificial Intelligence* 41(1): 1–63.
- Fernandéz, M.; Gómez-Pérez, A.; and Juristo, N. 1997. METHONTOLOGY: From Ontological Art to Ontological Engineering. Paper presented at the Spring Symposium on Ontological Engineering 24–25 March, Stanford University.
- Gruber, T. 1993. A Translation Approach to Portable Ontology Specifications. *Knowledge Acquisition* 5:199–220.
- Gruninger, M., and Fox, M. 1995. Methodology for the Design and Evaluation of Ontologies. Department of Industrial Engineering, University of Toronto.
- Guarino, N. 1998. Formal Ontology in Information Systems. In *Formal Ontology in Information Systems*, ed. N. Guarino, 3–15. Amsterdam, The Netherlands: IOS.
- Guarino, N., and Giarretta, P. 1995. Ontologies and Knowledge Bases: Toward a Terminological Clarification. In *Toward Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing*, ed. N. Mars, 25–32. Amsterdam, The Netherlands: IOS.

- Guarino, N., and Welty, C. 2002. Evaluating Ontological Decisions with ONTOCLEAN. *Communications of the ACM* 45(2): 61–65.
- Guarino, N.; Welty, C.; and Smith, B., eds. 2001. The Proceedings of the Second International Conference on Formal Ontology in Information Systems (FOIS-01), 16–19 October, Ogunquit, Maine.
- Guha, R., and Lenat, D. 1990. *CYC: A Mid-Term Report*. Technical Report ACT-CYC-134-90, Microelectronics Technology and Computer Corporation (MCC), Austin, Texas.
- Haas, L.; Miller, R.; Niswonger, B.; Roth, M.; Schwarz, P.; and Wimmers, E. 1999. Transforming Heterogeneous Data with Database Middleware: Beyond Integration. *Data Engineering Bulletin* 22(1): 31–36.
- Kashyap, V., and Sheth, A. 1996. Schematic and Semantic Similarities between Database Objects: A Context-Based Approach. *Very Large Databases Journal* 5(4): 276–304.
- Kashyap, V., and Sheth, A. 1998. Semantic Heterogeneity in Global Information Systems: The Role of Metadata, Context, and Ontologies. In *Cooperative Information Systems*, eds. M. Papzoglou and G. Schlageter, 139–178. San Diego, Calif.: Academic.
- MacGregor, R. 1991. Inside the LOOM Description Classifier. *SIGART Bulletin* (Special Issue on Implemented Knowledge Representation and Reasoning Systems) 2(3): 88–92.
- McGuinness, D.; Fikes, R.; Rice, J.; and Wilder, S. 2000. An Environment for Merging and Testing Large Ontologies. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, 483–493. San Francisco, Calif.: Morgan Kaufmann.
- Mena, E.; Kashyap, V.; Illarramendi, A.; and Sheth, A. 1998. Domain-Specific Ontologies for Semantic Information Brokering on the Global Information Infrastructure. In *Formal Ontology in Information Systems*, ed. N. Guarino, 269–283. Amsterdam, The Netherlands: IOS.
- Mitra, P.; Wiederhold, G.; and Kersten, M. 2000. A Graph-Oriented Model for Articulation of Ontology Interdependencies. In *Proceedings of Extending Database Technologies, EDBT 2000*, 86–100. Lecture Notes in Computer Science 1777. New York: Springer Verlag.
- Musen, M., and Noy, N. 2002. Evaluating Ontology Mapping Tools: Requirements and Experience, SMI-2002-0936, Informatics Lab, Stanford University.
- Naiman, C., and Ouksel, A. 1995. A Classification of Semantic Conflicts in Heterogeneous Information Systems. *Journal of Organizational Computing* 5(2): 167–193.
- Noy, N.; Ferguson, R.; and Musen, M. 2000. The Knowledge Model of Protege-2000: Combining Interoperability and Flexibility. Paper presented at the Second International Conference on Knowledge Engineering and Knowledge Management (EKAW-2000), 2–6 October, Ivans-Les-Pins, France.
- Obrst, L.; Wray, R.; and Liu, H. 2001. Ontological Engineering for B2B E-Commerce. In Proceedings of the International Conference on Formal Ontology in Information Systems (FOIS-2001), 17–20 October, Ogunquit, Maine.
- Omelayenko, B. 2002. Integrating Vocabularies: Discovering and Representing Vocabulary Maps. Paper presented at the International Semantic Web Conference 2002, 9–12 June, Sardinia, Italy.
- Ouksel, A. 1999. Ontologies Are Not the Panacea for Data Integration. *Journal of Parallel and Distributed Systems* 7(1): 7–35.
- Patil, R.; Fikes, R.; Patel-Schneider, P.; McKay, D.; Finin, T.; Gruber, T.; and Neches, R. 1992. The DARPA Knowledge Sharing Effort: Progress Report. In *Proceedings of the Knowledge Representation and Reasoning Conference (KR-92)*, eds. B. Nebel, C. Rich, and W. Swartout, 777–788. San Francisco, Calif.: Morgan Kaufmann.
- Rahm, E., and Bernstein, P. 2001. On Matching Schemas Automatically. Technical Report, 29, University of Leibzig.
- Sheth, A., and Ouksel, A., eds. 1999. *SIGMOD Record* (Special issue on Semantic Interoperability) 28(1).
- Skvortsov, N., and Kalinichenko, L. 2000. An Approach to Ontological Modeling and Establishing Intercontext Correlation in the Semistructured Environment, Paper presented at the Second Russian Conference on Digital Libraries, 25–27 September, Protvino, Russia.
- Smith, K., and Obrst, L. 1999. Unpacking the Semantics of Source and Usage to Perform Semantic Reconciliation in Large-Scale Information Systems. *SIGMOD Record* (Special Issue on Semantic Interoperability) 28(1).
- Uschold, M., and King, M. 1995. Toward Methodology for Building Ontologies. Paper presented at the Workshop on Basic Ontological Issues in Knowledge Sharing at the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), 20–25 August, Montreal, Canada.
- Welty, C. 1998. The Ontological Nature of Subject Taxonomies. In *Formal Ontology in Information Systems*, ed. N. Guarino, 317–327. Frontiers in AI Applications Series. Amsterdam, The Netherlands: IOS.
- Welty, C. 1995. Toward an Epistemology for Software Representations. In Proceedings of KBSE-95, The Tenth Knowledge-Based Software Engineering Conference, 148–154. Washington, D.C.: IEEE Computer Society.
- Wiederhold, G., and Genesereth, M. 1997. The Conceptual Basis for Mediation Services. *IEEE Expert* 12(5): 38–47.



Leo Obrst is a senior artificial intelligence scientist at the MITRE Center for Innovative Computing and Informatics, where he leads the information semantics team. He has worked in formal natural language semantics, computational linguistics, knowledge representation, and ontological engineering for 20 years. His Ph.D is in theoretical linguistics from the University of Texas at Austin. He is a member of the W3C Web Ontology Working Group developing owl;



A classic
...
still available
from AAAI

(members receive a 20% discount!)

650-328-3123
445 Burgess Drive
Menlo Park, CA 94025
www.aaai.org/Press/

a technical editor for the information flow framework metaontology candidate under the IEEE standard upper-ontology working group; and coauthor of the recent book, *The Semantic Web: A Guide to the Future of XML, Web Services, and Knowledge Management* by M. Daconta, L. Obrst, and K. Smith (Wiley 2003). His current research interests include semantic integration and interoperability, ontology mapping, and semantic web ontology languages. His e-mail address is lobrst@mitre.org.



Howard H. Liu is the chief technical lead at Applied Solar Technologies, a photovoltaic design and manufacturing company he co-founded in 1996. Previously, he was an ontologist at VerticalNet, an electronic-commerce company, and a consultant as a software system architect. He received an M.A. and a B.S. cum laude and with highest honors from the Department of Mathematics at the University of California at Los Angeles. He has co-authored research articles on ontological engineering and e-commerce and actively pursues his interests in mathematics, music, languages, and ontology-driven information systems. His e-mail address is liuh@e-math.ams.org.



Robert E. Wray is a senior scientist at Soar Technology. His research and experience encompass many areas of AI research and include agent-based systems and agent and cognitive architectures, machine learning, knowledge representation and ontology, neural networks, and cognitive psychology. He received a Ph.D. in computer science and engineering from the University of Michigan. His doctoral research focused on maintaining logical consistency in agent reasoning systems, and his innovations were incorporated into the SOAR architecture. Currently, he is Soar Technology's lead scientist for both the VIRTE (virtual technologies and environment) program and the third and fourth phases of the Air Force Research Lab AMBR (agent-based modeling and behavior representation) program. His e-mail address is wrayre@acm.org.

Post Your Job Listings on AAAI's web site! See www.aaai.org/Magazine/Jobs for details.