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

Why Evaluate Ontology Technologies? Because They Work!

Ontology technologies are popular and attract much attention because they're cornerstones for realizing the Semantic Web. But, what are they and how many exist? One survey lists over 50 ontology editors (www.xml.com/2002/11/06/Ontology_Editor_Survey.html). Furthermore, I can list a plethora of ontology technologies, such as inference engines, annotation tools, ontology-based crawlers, and mining tools, not to mention ontologies themselves. Ontologies' key benefit is interoperability, so it should be fairly easy, for example, to create an ontology with one editor, store it, and upload it again to another editor for further work. I suggest you take some time for this experiment—randomly pick two editors from the list and try it yourself.

Doing this, you will see that setting up experiments is a major effort. Some communities took the chance—for example, with the Text Retrieval Conference (http://trec.nist.gov) and the Message Understanding Conference (www.itl.nist.gov/iaui/894.02/related_projects/muc)—and benefited. So far, however, few people have dared experiment at all with ontology technologies—the lack of experimentation being a common phenomenon in computer science. As Walter Tichy discusses,[1] computer scientists and practitioners defend the lack of experimentation with a wide range of arguments. However, Tichy refused these excuses and argued for experiments' usefulness. The field is wide open for ontology experiments.

In this installment of Trends and Controversies, you'll find statements from different perspectives. A common distinction exists between evaluating ontology tools and ontology content. Asuncion Gomez-Perez makes this distinction explicit and focuses on the former. Walter Daelemanns and Marie-Laure Reinberger focus on the latter, as does Nicola Guarino. In addition to these more technical views on ontologies, Natalya F. Noy points out ontology consumers' needs.

In the end, ordinary users will decide if they're happy using ontology technologies (at all) and whether the Semantic Web will become a truly global success. This will occur only if ontology technologies really work. So, let's prove that they do.

—York Sure

Reference

1. W.F. Tichy, "Should Computer Scientists Experiment More?" *Computer*, vol. 31, no. 5, 1998, pp. 32–40.

Evaluating Ontology Evaluation

Asunción Gómez-Pérez, *Universidad Politécnica de Madrid*

Before we can give industry recommendations for incorporating ontology technology into their IT systems, we must consider two types of evaluation: content evaluation and ontology technology evaluation. Evaluating content is a must for preventing applications from using inconsistent, incorrect, or redundant ontologies. It's unwise to publish an ontology that one or more software applications will use without first evaluating it. A well-evaluated ontology won't guarantee the absence of problems, but it will make its use safer. Similarly, evaluating ontology technology will ease its integration with other software environments, ensuring a correct technology transfer from the academic to the industrial world.

In this contribution, I explore both evaluation dimensions to try to answer the following questions:

- How were widely used ontologies (including Cyc, WordNet and EuroWordNet, Standard Upper Ontology, and the DAML+OIL library) evaluated during development or once they were implemented in an ontology language?
- How robust are ontology evaluation methods? What type of ontology components do they evaluate? Are they independent of the language used to implement the ontologies?
- How do ontology development platforms perform content evaluation? How mature are the evaluation tools incorporated on such platforms? Which types of errors do these tools detect?
- What are the criteria used for evaluating ontology tools? What are the results?

Ontology evaluation

Work on ontology content evaluation started in 1994.[1] In the last two years, the ontological engineering community's interest in this issue has grown and extended to the

1541-1672/04/$20.00 © 2004 IEEE
Published by the IEEE Computer Society

evaluation of technology used to build ontologies. You can find a survey on evaluation methods and tools in *Ontological Engineering*.[2]

Ontology content evaluation has three main underlying ideas:

- We should evaluate ontology content during the entire ontology life cycle.
- Ontology development tools should support the content evaluation during the entire ontology building process.
- Ontology content evaluation is strongly related to the underlying knowledge representation (KR) paradigm of the language in which the ontology is implemented.

Ontology technology evaluation's main underlying idea is that because ontology technology is maturing and should soon be ready for industry, we must evaluate and benchmark ontology technology to ensure a smooth transference. The evaluation should consider several factors—including interoperability, scalability, navegability, and usability.

The relationship between evaluating ontology tools and ontologies

Ontologies are built using different methodological approaches and with different ontology building tools that generate the ontology code in several languages. We can examine evaluation efforts under the following four perspectives.

Content

From a content perspective, many libraries exist where ontologies are published and publicly available (see the sidebar for some of the best-known libraries). There is no documentation available about how ontologies available in libraries as well as well-known and large ontologies (such as Cyc, some ontologies at the Ontoligua Server, and SENSUS) were evaluated. However, the ontology and Semantic Web communities have already used these ontologies to build many successful applications. We need studies that demonstrate that well-evaluated ontologies increase the performance of applications that use them.

Methodology

From a methodological perspective, the main efforts to evaluate ontology content occurred in the Methontology framework[2] and with the OntoClean method.[3] Methontology proposes that you evaluate ontology content throughout the entire lifetime of the development process. You should carry out most of the evaluation (mainly consistency checking in concept taxonomies) during the conceptualization activity to prevent errors and their propagation in the implementation. OntoClean is a method to clean concept taxonomies according to *metaproperties* such as *rigidity*, *identity*, and *unity*. Metaproperties are useful for removing wrong *subclass of* relations in the taxonomy. Both approaches evaluate only concept taxonomies—they don't propose specific methods for evaluating other types of components such as properties, relations, and axioms.

DECK, xxxxx xxxxxx xxxxxx insert
a xxxxx xxxxxxx xxx
six line xxx xxxxxx xxxxxxxxx
body xxxxx xxxxx xxxxx
deck xx xxxxxxxx xxxxx xx
here xx xxxxxxxx xxxxx xx.

Implementation

From an implementation perspective, we can find important connections and implications between the components we use to build ontologies (concepts, relations, properties, and axioms); the knowledge representation paradigms we use to formally represent such components (frames, description logic (DL), first order logic, and so on); and the languages we use to implement them (for example, we can implement an ontology built with frames, DL in several frames, or DL languages). This is important from an evaluation perspective because different KR paradigms offer different reasoning mechanisms that we can use in content evaluation:

- We can use DL classifiers to derive concept satisfiability and consistency in the models implemented using subsumption tests. Such tests are commonly built using tableaux calculus and constraint systems.
- We can extend existing methods for evaluating frame-based concept taxonomies with the evaluation of new components (properties, relations, axioms).

Technology

From a technological perspective, ontology tool developers have gathered experience on evaluating tools working on the OntoWeb European thematic network's SIG3 (Special Interest Group on Enterprise Standard Ontology Environments). Different ontology tool developers have also conducted comparison studies of different types of ontology tools, which you can find in the OntoWeb deliverable D1.3.[4] Here I highlight three important findings from these studies.

First, the most well-known ontology development tools (OILed, OntoEdit, Protégé2000, WebODE, and WebOnto) provide constraint checking functionalities. Regarding taxonomy consistency checking, most of them can detect circularity errors. However, this ability isn't enough and should be extended.

Secondly, only a few specific tools exist for evaluating ontology content. ONE-T verifies Ontolingua ontologies' concept taxonomies; OntoAnalyser focuses on evaluating ontology properties, particularly language conformity and consistency; ODEClean is a WebODE plug-in that supports the OntoClean method; and OntoGenerator is an OntoEdit plug-in focused on evaluating ontology tools' performance and scalability.

Finally, different groups or organizations might develop ontologies, and ontologies might be available in different languages. In the Semantic Web context, some RDF(S), DAML+OIL, and OWL *checkers*, *validators*, and *parsers* exist, and several ontology platforms can import RDF(S), DAML+OIL, and OWL ontologies. As colleagues and I have demonstrated,[5] some parsers (Validating RDF Parser, RDF Validation Service, DAML Validator, and DAML+OIL Ontology Checker) don't detect taxonomic errors in ontologies implemented in such languages. So, if ontology platforms import such ontologies, can the platforms detect such problems? The same study reveals that most ontology platforms only detect a few errors in concept taxonomies before importing those

Ontology libraries

DAML: www.daml.org/ontologies
Ontolingua: http://ontolingua.stanford.edu
Protégé2000: http://protege.stanford.edu
SemWEbCentral: http://www.semwebcentral.org/index.jsp
SHOE: www.cs.umd.edu/projects/plus/SHOE/onts/index.html
WebODE: http://webode.dia.fi.upm.es/
WebOnto: http://webonto.open.ac.uk

ontologies. So, we must develop language-dependent evaluation tools that can evaluate ontologies in the traditional (Ontolingua, OCML, Flogic, and so on) and Semantic Web (RDF, RDFS, DAML+OIL, and OWL) languages. Each tool must take into account each languages' features to perform this evaluation.

Evaluating ontology technology

However, SIG3's main goal isn't to evaluate how ontology tools evaluate ontologies, but to compare and evaluate ontology technology to better assess the ontology technology transfer to industry. In fact, the dimensions this technology evaluation uses include:

- The expressiveness of the ontology editors' underlying KR model. The goal is to analyze which knowledge components can be represented in each tool and how each tool must represent different components. The first EON2002 workshop (http://km.aifb.uni-karlsruhe.de/eon2002), sponsored by OntoWeb, focused on this dimension.
- The quality of each tool's ontology export-import functions. The goal is to analyze how the quality of these functions affects how ontology tools exchange their ontologies and interoperate. The second EON2003 workshop's experiment focused on this dimension (http://km.aifb.uni-karlsruhe.de/ws/eon2003).

The experiments performed at EON2002 and EON2003 show that tools with similar underlying knowledge models preserve more knowledge in the knowledge exchange process and hence are more interoperable. These experiments have also shown that we could use RDF(S) as a common exchange format between ontology tools. However, because RDF(S) is less expressive than the knowledge model most of these tools provide, much knowledge is lost during transformation. Either these tools don't export all the knowledge represented in the ontologies or they generate ad hoc nonstandard RDF(S) sentences to preserve the knowledge in circular transformations, making it difficult for the other tools to "understand" them.

Future experiments will focus on other dimensions, such as

- *Scalability*: Analyzing how different ontology building platforms scale when managing large ontologies with thousands of components, and the time required to open and save ontologies, to create, update, or remove ontology components, to compute simple or complex queries, and so on.
- *Navegability*: Analyzing how ontology tools allow for navigating large ontologies—how easy is it to search for a component (graphically, text based, and so on), to extend the ontology with new components, to obtain a small part of the ontology, and so on?
- *Usability*: Analyzing user interfaces' clarity and consistency, the learning time for users, stability, help systems, and so on.

The Knowledge Web Network of Excellence, which the EU funds, will follow up the OntoWeb evaluation innitiatives to ensure that ontology technology transfers to the industry market, taking into account the industrial needs identified from use cases and industrial scenarios.

Acknowledgments

The Knowledge Web Network of Excellence (FP6-507482), the Ontoweb Thematic Network (IST-2000-29243), the Esperonto project (IST-2001-34373), and the ContentWeb project (TIC-2001-2745) have all partially supported this work. Thanks to Óscar Corcho and Carmen Suarez de Figueroa Baonza for their comments.

References

1. A. Gómez-Pérez, *Some Ideas and Examples to Evaluate Ontologies*, tech. report KSL-94-65, Knowledge System Laboratory, Stanford Univ., 1994.
2. A. Gómez-Pérez, M. Fernández-López, and O. Corcho, *Ontological Engineering: With Examples from the Areas of Knowledge Management, e-Commerce and the Semantic Web*, Springer-Verlag, Nov. 2003.
3. C. Welty and N. Guarino, "Supporting Ontological Analysis of Taxonomic Relationships," *Data and Knowledge Eng.*, vol. 39, no. 1, 2001, pp. 51–74.
4. A. Gómez-Pérez, *A Survey on Ontology Tools*, OntoWeb deliverable D1.3.2002; http://ontoweb.aifb.uni-karlsruhe.de/About/Deliverables/D13_v1-0.zip.
5. A. Gómez-Pérez and M.C. Suárez-Figueroa, "Results of Taxonomic Evaluation of RDF(S) and DAML+OIL Ontologies using RDF(S) and DAML+OIL Validation Tools and Ontology Platforms Import Services," *CEUR Workshop Proceedings* (CEUR-WS.org), vol. 87, 2003.

Shallow Text Understanding for Ontology Content Evaluation

Walter Daelemans and Marie-Laure Reinberger

If an ontology is indeed a "formal, explicit specification of a shared conceptualization" (www.ktweb.org), the question we should focus on becomes "shared by whom, for what purpose, and for how long?" As has often been the case in knowledge representation research's history, too much effort is put on developing numerous "nice" techniques for making ontologies formal and explicit. Relatively little emphasis is placed on developing techniques for managing content collection and maintenance and, in the case of ontologies, on techniques for showing that an ontology indeed represents a consensual conceptualization and not just one person's ideas. We should guide evaluation toward ontologies' semantics, not just their syntax.

Ontologies have task-dependent and static natures, and most are created by people with a limited perspective on possible alternative conceptualizations. This means an enormous barrier exists to their large-scale development and maintenance in areas such as knowledge management and the Semantic Web. To solve this problem, ontology researchers should focus on semiautomatic text-analysis-based updating, enriching, filtering, and evaluation of ontologies.

Information extraction techniques

For a long time, the idea of using natural language processing tools to robustly analyze text from any domain and any genre was a fiction—it still is if you want a deep understanding of textual meaning. However, with the introduction of statistical and machine-learning techniques into language technology, many languages can now access tools that allow for recognizing and analyzing sentences' major constituents and their most important relations (subject, object, time, location, and so on). These tools can also help detect concepts (for example, not only instances of company names, person names and so on, but also very specific concepts such as protein names or diseases). These techniques are called *information extraction*, *named entity recognition*, and *shallow parsing*, and they often perform at reasonably high precision and recall levels (80 to 90 percent). For each sentence in a text, you can extract the main concepts and their (grammatical) relations this way.

Combined with natural language processing's standard pattern-matching and machine-learning techniques, these techniques also let you extract concepts and relations between concepts—in short, they let you extract ontological knowledge from text. Although researchers have explored this type of work for some time (at least since the start of this century[1]), only recently are those working in this area becoming more organized. For example, a recent special interest group of the EU network of excellence, Ontoweb, is devoted to this topic (www.ontoweb.org). You can see the approach's increasing maturity at the European Conference on Artificial Intelligence's 2004 workshop, which will focus on developing reliable quantitative methods to evaluate the quality of extracted ontological knowledge and objectively compare different approaches.

As soon as reliable ontology extraction from text becomes available, large-scale, semiautomatic ontology content creation will also be possible. In a typical setup, a human ontology engineer would start from a handcrafted initial ontology, collect texts about the concepts described (from the Web, company internal document libraries, and so on), and apply the ontology extraction tools to this textual material. This will reveal conflicting perspectives on conceptual relations (ontology evaluation), allow for populating the initial ontology with additional instances and relations (ontology extension), and, in time, allow for tracking changes to ontologies.

The Ontobasis project

Ontobasis is a Belgian IWT-funded project[2] with groups from Brussels and Antwerp (http://wise.vub.ac.be/ontobasis). We focus here on ontology extraction from text work done in Antwerp.

Our research focuses on using a shallow parser[3]—which robustly and efficiently analyzes unrestricted English text—and applying it to extracting ontological knowledge. For example, the shallow parser would analyze a sentence such as

> The patients followed a healthy diet, and 20% took a high level of physical exercise.

into a structure (simplified from real output) such as

> (Subject [The patients]) (Verb [followed]) (Object [a healthy diet]) and (Subject (Percentage [20%])) (Verb [took]) (Object [a high level]) (PP [of physical exercise]).

Extracting word clusters

The shallow parser we use is efficient enough to analyze thousands of words per second, and we use it to analyze a corpus of texts related to the domain for which we're building an ontology. In the Ontobasis project, one of these domains is the Medline abstract language (a biomedical language). Any shallow parser has a relatively high error rate, so analyses will contain several errors. However, this isn't necessarily a problem because applications such as extraction of ontological relations from text allow frequency filtering when sufficiently large corpora are available. By taking into account only relations that are frequent enough in the corpus, we can exclude spurious relations due to the shallow parser's random errors.

The first step is to select a set of terms that are relevant in the domain. We can do this manually or by automatically analyzing documents about the domain of interest using standard terminology extraction techniques. Usually, these techniques are based on statistical analysis (TF-IDF) or mutual information for multiword terms, sometimes combined with linguistic pattern matching.

Once we have such a set of terms, we extract from the shallow parsed corpus all their occurrences. We then determine with which verbs they enter in subject, object, or other syntactic relations and how often. The linguistic motivation for this is that much of a term's meaning is implicit in its relations with other terms. Terms with similar syntactic relations to other terms are semantically related. Using clustering techniques, we can use these semantic similarities or semantic dependencies to group terms into classes, thereby providing terms that are possible candidates for extending the initial ontology or creating one from scratch. Following are some example term clusters extracted from the medical corpus:

- Hepatitis, infection, disease, cases, syndrome
- Liver, transplantation, chemotherapy, treatment
- Face mask, mask, glove, protective eyewear

Evaluating and extending ontologies

Objective quantitative evaluation of the output of this clustering stage isn't easy. Apart from an impressionistic idea of general quality, more sophisticated quantitative evaluation is difficult. The obvious possibilities—such as comparing extracted ontologies to existing ones in terms of computation of recall, overlap, precision, and so on—give us some indication but are limited because our approach is intended to evaluate and extend existing ontologies anyway. More progress in evaluating text-based ontology learning might come from carefully constructing a gold-standard ontology on the basis of manually analyzing a corpus.

Clustering terms in semantically related classes is only a first step in automatically extracting ontological knowledge. Given that we have some ontological knowledge—for example, that infections are transmittable—we can combine the classes constructed by clustering with pattern matching rules to considerably extend the number of relations in the ontology. For example, if we see that hepatitis and disease are linked to "infections" in a cluster, we can hypothesize that hepatitis and diseases in general are also transmittable. To

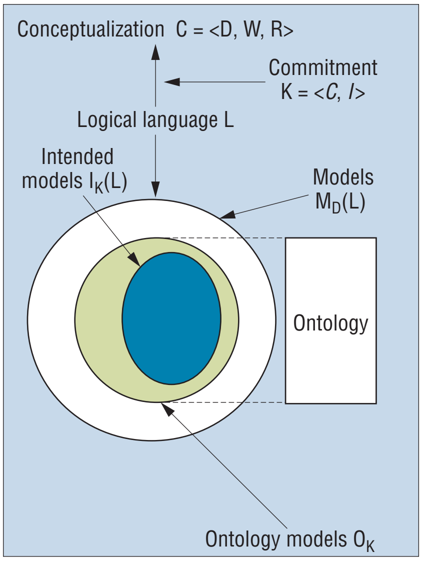

Figure 1. The relationship between an ontology and a conceptualization. Given a logical language L that implicitly commits to a conceptualization C, an ontology's purpose is to capture those models of L that are compatible with C. These models are called the intended models.

demonstrate, we use pattern matching to extract the following relations on the preposition "of":

> [recurrence transmission] of [infection hepatitis_B_virus viral_infection HCV hepatitis_B HCV_infection disease HBV HBV_infection viral_hepatitis]

By tuning the pattern matching to ontological relations such as part-whole relations and specialization-generalization relations, we can easily extend ontologies. However, both in the clustering step (which you could interpret as extending or evaluating the extension of concepts) and in the pattern-matching step (which you could interpret as populating the ontology with selected relations), human intervention is essential to evaluate the system proposals. The difference with purely handcrafted ontology development is that recognizing and evaluating proposed ontological structure is much easier, more complete, and faster than inventing ontological structures.

Language technology tools have advanced to such a level of accuracy and efficiency that it's now possible to automatically analyze huge amounts of text. Like most researchers in this field, we believe that this approach will solve some hard problems in ontology content creation, adaptation, and evaluation but will always require human interaction.

References

1. A. Gomez-Perez and D. Manzano-Macho, eds., *Deliverable 1.5: A Survey of Ontology Learning Methods and Tools*, Ontoweb deliverable, 2003; http://ontoweb.aifb.uni-karlsruhe.de/Members/ruben/Deliverable%201.5.
2. M.-L. Reinberger et al., "Mining for Lexons: Applying Unsupervised Learning Methods to Create Ontology Bases," *On the Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, LNCS 2888, Springer-Verlag, 2003, pp. 803–819.
3. W. Daelemans, S. Buchholz, and J. Veenstra, "Memory-Based Shallow Parsing," *Proc. Computational Natural Language Learning* (CoNLL-99), Assoc. for Computational Linguistics, 1999, pp. 53–60; http://ilk.uvt.nl/cgi-bin/tstchunk/demo.pl.

Toward a Formal Evaluation of Ontology Quality

Nicola Guarino

Like every software product, ontologies need proper quality control to be effectively deployed in practical applications. Unfortunately, adapting them to the evaluation metrics and quality enforcement procedures developed for software engineering doesn't work. An ontology's nature is very different from that of a piece of code. We can't evaluate ontologies in terms of their correctness with respect to a given process specification, described, for instance, using an I/O function. Indeed, ontologies are not software processes—rather, they belong to the class of data models. Regrettably, however, current criteria for data models' quality tend to be typically ad hoc, depending on their stakeholders' needs, with little agreement about criteria for good, stable data models, flexible to changing business practices.

Ontologies, on the other hand, are supposed to be shareable across different communities and applications—at least in their more ambitious application perspectives, such as the Semantic Web. In the past, some researchers have proposed criteria for evaluating how we represent ontologies.[1] However, I think the most urgent need is developing general, rigorous ways to evaluate ontologies with respect to their main purpose: specifying a given vocabulary's intended meaning.

Ontologies and conceptualizations

In this essay, I use the common definition of an ontology—that is, a "specification of a conceptualization,"[2] which I've discussed and formalized elsewhere.[3] A key observation emerging from my analysis is that ontologies are only approximate specifications of conceptualizations. So, it seems appropriate to evaluate them on the basis of the degree of such approximation. This idea isn't so obvious to implement, however, because the relationship between an ontology and a conceptualization is rather delicate and requires some technical clarification.

Consider Figure 1, which is based on the picture I present in my earlier work.[3] For this discussion, let's assume an informal, intuitive understanding of what a conceptualization is: a set of conceptual relations, intended as systematic ways an agent perceives and organizes a certain domain of reality, abstracting from the various actual occurrences of such reality (so-called situations or possible worlds). For example, the conceptual relation "being bigger than" belongs to my conceptualization because I know how to recognize its instances in various situations.

According to this intuition, I proposed to formally describe a conceptualization in terms of Montague's semantics, as a triple $\mathbf{C} = <D, W, \Re>$, where D is a set of relevant entities, W is a set of possible states of affairs (or worlds) corresponding to mutual arrangements of such entities, and $\Re$ is a set of conceptual relations, defined as functions from *W* into suitable relations on *D*. If we want to talk about a conceptualization **C** using a logical language **L**, we must assign a certain preferred interpretation to its nonlogical symbols (predicates and constants)—that is, we need to commit to **C** by means of a suitable interpretation function $\mathtt{I}$. Figure 1 shows the commitment of **L** to **C** as a couple $\mathbf{K} = <\mathbf{C}, \mathtt{I}>$.

Consider the set $M_D(L)$ of models of **L** relative to the domain *D*; in general, this is huge (although finite if *D* is finite). However, we want to focus only on the intended models **K** induces—that is, the set $\mathbf{I_K}$. The ontology's role emerges here. As the figure shows, an ontology is simply a logical theory designed in such a way that the set $\mathbf{O_K}$ of its models relative to the conceptualization **C** under the commitment **K** is a suitable approximation of the set $\mathbf{I_K}$ of the intended models. In other words, an ontology's purpose is to exclude nonintended models—those models outside the gray oval (for example, those models that let something be "bigger than" itself).

Coverage and precision

Typically, as a result of the ontology axioms, the set $\mathbf{O_K}$ will properly cover $\mathbf{I_K}$. In general, however, we have five possible situations:

1. $I_{\mathbf{K}} \cap O_{\mathrm{K}} = \varnothing$
2. $I_{\mathbf{K}} = O_{\mathrm{K}}$
3. $I_{\mathbf{K}} \subset O_{\mathrm{K}}$
4. $O_{\mathbf{K}} \subset I_{\mathrm{K}}$
5. $I_{\mathbf{K}}$ and O_{K} do properly overlap

Situation 1 isn't very interesting; we would say in this case that the ontology is totally "wrong" with respect to the particular conceptualization. Situation 2 is (apparently) an ideal case, which is almost impossible to reach. We should note however that, even in this case, we can't always say that the ontology fully captures the conceptualization because multiple worlds in the conceptualization might correspond to just one ontology model. This problem is bound to the distinction between the *ontological* notion of "world" (or state of affairs) and the *logical* notion of "model,", but I won't discuss that here.

Figure 2 shows Situations 3 through 5 and introduces the first two dimensions I use to formally evaluate ontologies: *coverage* and *precision*. Assuming that the domain D is finite (which implies that all the model sets in the figure are finite), we can define them as

$$(\mathrm{D1}) \qquad \mathrm{C}=\frac{|I_{\mathbf{K}} \cap O_{\mathbf{K}}|}{|I_{\mathbf{K}}|} \qquad \text{(coverage)}$$

$$(\mathrm{D2}) \qquad \mathrm{P}=\frac{|I_{\mathbf{K}} \cap O_{\mathbf{K}}|}{|O_{\mathbf{K}}|} \qquad \text{(precision)}$$

We can immediately recognize that these two dimensions are analogous to those used in information retrieval. The difference is that, in our situation, the term "coverage" seems more appropriate than "recall." To emphasize the analogy, imagine an ontology as a device whose purpose is to retrieve the intended models.

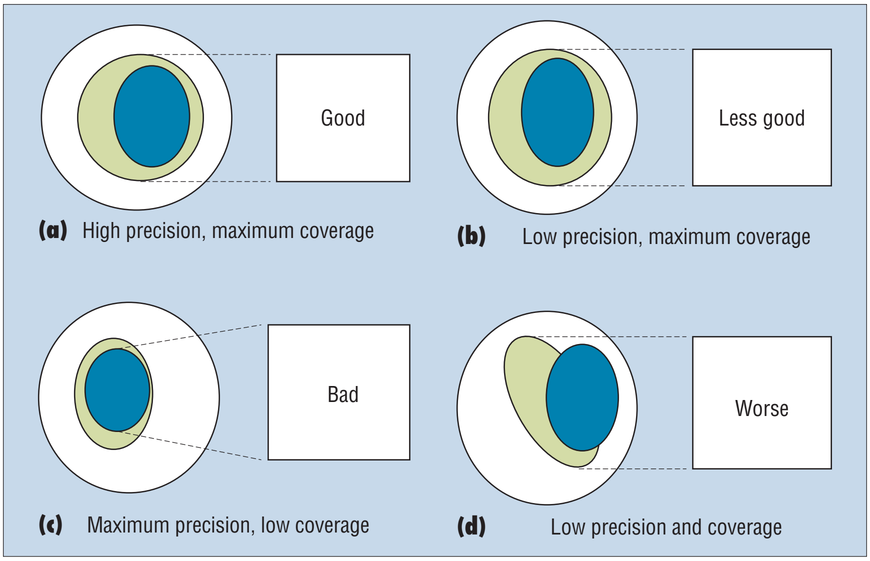
Good
(a) High precision, maximum coverage
Less good
(b) Low precision, maximum coverage
Bad
(c) Maximum precision, low coverage
Worse
(d) Low precision and coverage

Figure 2. A comparison of different ontologies with respect to coverage and precision.

Figure 2 depicts some typical situations exhibiting different degrees of coverage and precision. Clearly, coverage is important for an ontology; if it goes under 100 percent, some intended model isn't captured. Precision is often less important, especially if a certain user community knows in advance the meaning of the terms the ontology describes. However, unprecise ontologies can generate serious problems in cases where it's necessary to check whether two concepts are disjoint. Consider Figure 3, which you can read in two ways. In the first reading, assume that, $\mathbf{I_K}(A)$ and $\mathbf{I_K}\ (B)$ denote the set of all possible instances of the two concepts (unary predicates) *A* and *B*, that is, their possible intended interpretations under the commitment **K**. In this example, the two concepts are disjoint by hypothesis. However, if the ontology O is (more or less) unprecise, it might allow an overlap in the extension of the two concepts, as in the present example. So, logically speaking, the ontology O "believes" that A and B can have common instances.

The situation is even worse if you want to align imprecise ontologies that have different commitments, say $\mathbf{K}_A$ and $\mathbf{K}_B$. In this case, you can read Figure 3 as in the previous figures. Assume that the outside circle denotes the set of all possible models of a certain language **L**, while **O**(*A*) and **O**(*B*) are model set of two different (rather imprecise) ontologies, relative to the same language **L**. Because of their imprecision, the two ontologies could have some models in common, indicating that they "agree" on something, but this might be a false agreement because no intended models are involved. So, we might risk relying on the two ontologies' syntactic interoperability, with no warranties concerning the actual intended meaning of the terms they define.

This is why I believe that so-called lightweight ontologies can't generally guarantee interoperability, and why we must develop axiomatic theories based on "deep" ontological principles.

The role of examples and counter-examples

I've introduced the basis for a new formal framework for evaluating and comparing ontologies by measuring their "distance" from a reference conceptualization. This is a work in progress, and the chances of getting a quantitative metric are limited to the case of finite domains. However, even in the case of infinite domains, we can obtain interesting results by focusing on finite lists of examples and counterexamples, so that we can do evaluations at least with respect to such. This would certainly be important practically. For instance, these examples might be encoded in a form that facilitates immediate, visual validation by a team of domain experts (not ontology experts) and might be supplemented by "competency questions"[4] to characterize the expected reasoning tasks. I'm thinking of annotated multime-

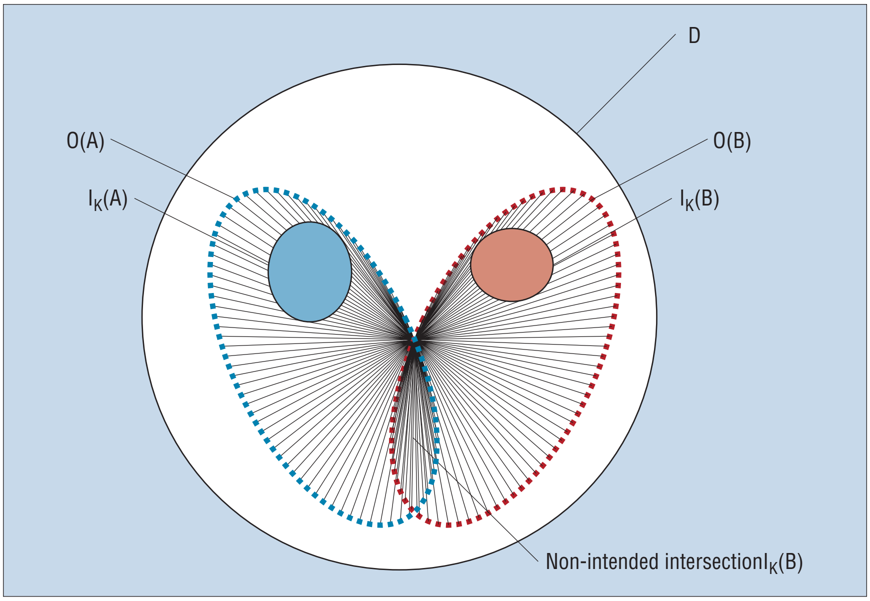

Figure 3. Unprecise ontologies introduce the risk of "false agreements," owing to the fact that they might overlap on unintended models.

dia documents, something similar to sophisticated versions of the illustrated dictionaries children use. After all, these are ways to convey words' intended meaning. It shouldn't be difficult to analyze the correspondence between these examples and counterexamples, on one hand, and intended-and nonintended models, on the other. This way, we should be able to get quantitative metrics corresponding to the criteria I've presented and evaluate, compare, and even certify ontologies with respect to lists of validated examples.

References

1. A. Gómez-Pérez, M. Fernandez-Lopez, and O. Corcho, *Ontological Engineering,* Springer-Verlag, 2004.
2. T.R. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," *Int'l J. Human and Computer Studies*, vol. 43, nos. 5–6, 1995, pp. 907–928.
3. N. Guarino, "Formal Ontology in Information Systems,"*, Proc. Int'l Formal Ontology in Information Systems* (FOIS 98), IOS Press, 1998, pp. 3–15.
4. M. Uschold and M. Gruninger, "Ontologies: Principles, Methods, and Applications," *Knowledge Engineering Review, vol. 11, no. 2, 1996, pp. 93–155.*

Evaluation by

Ontology Consumers

Natalya F. Noy

When we talk about evaluating ontologies today, what we usually have in mind is some sort of "objective" evaluation of how "good" an ontology is. Methodologies such as OntoClean[1] help validate taxonomic relationships with respect to general ontological notions such as essence, identity, and union. Others suggest assessing ontologies' completeness, consistency, and correctness in terms of consistency of inference, lack of redundancy, lack of errors, and so on.[2] Another existing thrust in comparing and evaluating ontologies[3] is providing information on their intrinsic properties that range from such features as authors' names or an ontology's accessibility and price to formalisms and methodologies used for its development. Furthermore, many have argued that the only true way to evaluate an ontology is to use it in applications and assess the applications' performance.

Although all these evaluation types or comparison methods are necessary, none are helpful to *ontology consumers*, who need to discover which ontologies exist and, more important, which ones would be suitable for their tasks at hand. Knowing whether an ontology is correct according to some specific formal criteria might help in our ultimate decision to use an ontology but will shed little light on whether or not it's good for a particular purpose or task. As ontologies become the backbone of the Semantic Web and come into widespread use in many disciplines (such as biomedical informatics), their main consumers will be developers who must decide which one to use for their projects. It is these often naive ontology consumers who desperately need help determining what's available and how good it is for them.

One reason ontologies have become popular is that, as shared, agreed-on descriptions of domains different agents use, they hold the promise of facilitating interoperation among software resources—a key requirement, for example, for the Semantic Web to succeed. In other words, if I'm developing a Semantic Web service and choose to reuse an ontology to support it rather than create a new ontology, I get the interoperation with others using the same ontology "for free." Additionally, I save the time and money required to develop an ontology and get the benefit of using an ontology that others have already tested.

Unfortunately, as the number of existing ontologies and ontology libraries grow, reusing ontologies becomes harder rather than easier. Almost nothing today will help an aspiring ontology consumer discover which of the existing ontologies are well suited for his or her tasks, which ontologies others have used successfully for similar tasks, and so on. We need not only a system for evaluating ontologies objectively from some generic viewpoint (we have that already, to some extent), but also practical ways for ontology consumers to discover and evaluate ontologies. Information such as the number of concepts or even an ontology's complete formal correctness is probably not the most important criteria in this task (although it's often the easiest to obtain).

Several techniques could help. We must focus on developing these techniques and services if we ever want ontologies' use and, more important, reuse to be commonplace.

Ontology summarization

To decide whether to buy a book, we read the blurb on the book jacket; to decide whether a paper is relevant to our work, we read its abstract. To decide whether a particular ontology fits our application's requirements, we need some abstract or summary of what this ontology covers. Such a summary can include, for example, a couple of top levels in the ontology's class hierarchy—perhaps a graphical representation of these top-level concepts and links between them. We can generate these top-level snapshots automatically or let ontology authors include them as metadata for an ontology.

The summary can also include an ontology's *hub* concepts—those with the largest number of links in and out of them. What's more interesting, we can experiment with metrics similar to Google's PageRank: the concept is more important if other important concepts link to it. This computation can take into account specific links' semantics (giving a subclass-superclass link a lower value than a property link, for instance) or exclude some links or properties. By experimenting with these measures, we can discover which ones yield the concepts that users deem important. The hub concepts are often much better starting points in exploring and understanding an ontology than the top level of its class hierarchy.

Epinions for ontologies

In addition to reading a book's blurb to determine if we want to buy it, we often read reviews of the book by both book critics and other readers. Similarly, when choosing a movie or a consumer product, such as a coffee maker or a pair of skis, we use the Web to find others' opinions. You've probably visited such sites as the Internet Movie Database (www.imdb.com) or Amazon.com for reviews. A similar network for ontologies would help guide our ontology-consumer friend in finding whether a particular ontology would be suitable for his or her project. The reviews should include not only an ontology's qualitative assessment (Is it well developed? Does it have major holes? Is it correct?) but also, and perhaps more important, experience reports. Suppose a person whose reviews of ontologies I generally trust has successfully used a particular wine ontology to develop an agent that pairs wines with food. The report of a successful use by this trusted person strongly suggests to me that I could use this ontology as a component of my agent for creating restaurant menus that include suggested wines with each course. In fact, some communities are beginning to organize such portals (see, for example, obo.sourceforge.net).

Epinions (www.epinions.com) takes the concept of consumers providing reviews for products further, letting its users establish Webs of Trust—networks of reviewers whose reviews and ratings they trust. Letting ontology consumers create their own Webs of Trust could also be extremely helpful. Some might be more interested in ontologies' formal properties, and their networks would include reviewers that pay particular attention to the formal aspects. Others might care much more about intuitive and simple concept organization and hence have a different set of reviewers in their Webs of Trust. You could argue that fewer consumers need ontologies than coffee makers and that we'll never achieve a critical mass of reviews to make such a service valuable. However, a Google search for "ontology" produces more than a million hits, and most refer to the computer science notion of ontology. Add to that ontologies disguised as terminologies, standard vocabularies, or XML Schemas, and we might well have the critical mass.

Views and customization

To evaluate an ontology properly, users might need to see a view of an ontology that takes into account their expertise, perspectives, the required level of granularity, or a subset of the domain the ontology they're interested in covers. For instance, if we're developing an application that studies breast cancer, we might want to use a standard anatomy ontology, such as the Foundational Model of Anatomy.[4] However, the FMA is huge and complex (67,000 distinct concepts at the time of this writing). We might choose to use only a subset of it that includes breast and related organs. Similarly, while FMA takes a structure-based view of anatomy and is developed as a general reference model, a radiologist or someone writing medical simulations might use different terms or view some relationships differently.

If we can let ontology developers annotate concepts and relations with information about which perspectives these terms and relations should appear in and how to present or name them, we'll be able to present these different perspectives automatically. Similarly, an ontology developer might want to indicate that certain concepts or relations should be displayed only to users who identify themselves as experts (presenting a simpler, trimmed-down view for novices). For an ontology consumer, it's often much easier to evaluate a smaller ontology with only the concepts related to his or her concepts of interest than to evaluate a large general reference resource.

Looking forward

Naturally, even if we succeed in creating usable, comprehensive tools and services that let ontology consumers find the right ontologies and reuse them rather than develop their own, we won't fully eliminate the proliferation of similar or overlapping ontologies. Someone will always want to use his or her own ontology rather than reuse an existing one, benefits of sharing and interoperability notwithstanding. There might be good reasons for this approach, from institutional (the requirement to use only proprietary information), to practical (the need to interoperate with legacy systems), to many others. What we can do, however, is reduce the number of cases of developers creating their own ontologies simply because they couldn't find and properly evaluate existing ones.

I have discussed only some of the ways that could help ontology consumers (rather than ontology developers and experts) evaluate existing ontologies for their use. Many more reasons must exist, and I hope this area will get more attention from ontology and Semantic Web researchers in the near future. ■

References

1. N. Guarino and C. Welty, "Evaluating Ontological Decisions with OntoClean," *Comm. ACM*, vol. 45, no. 2, 2002, pp. 61–65.
2. A. Gòmez-Pèrez, "Ontology Evaluation," *Handbook on Ontologies*, S. Staab and R. Studer, eds., Springer-Verlag, 2003, pp. 251–274.
3. J. Arprez et al., "Reference Ontology and $(\mathrm{ONTO})^2$ Agent: The Ontology Yellow Pages," *Knowledge and Information Systems*, vol. 2, no. 4, 2000, pp. 387–412.

York Sure is an assistant professor at the University of Karlsruhe's Institute of Applied Informtics and Formal Description Methods (Institute AIFB). Contact him at Inst. AIFB, Univ. of Karlsruhe, 76128 Karlsruhe, Germany; sure@aifb.uni-karlsruhe.de.

Asunción Gómez-Pérez is the director of the Ontological Engineering Group at UPM. Contact her at asun@fi.upm.es.

Walter Daelemans is professor of Computational Linguistics at the University of Antwerp, Belgium and part-time professor of Machine Learning and Language Technology at Tilburg University, The Netherlands. Contact him at walter.daelemans@ua.ac.be.

Marie-Laure Reinberger holds a postdoctoral position at the University of Antwerp in the Ontobasis project. Contact her at marielaure.reinberger@ua.ac.be.

Nicola Guarino is a senior research scientist at the Institute of Cognitive Sciences and Technologies of the Italian National Research Council (ISTC-CNR), where he leads the Laboratory for Applied Ontology (LOA) in Trento. Contact him at guarino@loa-cnr.it.

Natalya F. Noy bio missing?