Conceptual Modeling for Distributed Ontology Environments

Deborah L. McGuinness

Associate Director and Senior Research Scientist,
Knowledge Systems Laboratory, Gates Building 2A,
Stanford University, Stanford, CA 94305 USA
dlm@ksl.stanford.edu

Abstract. As ontologies become common in more applications and as those applications become larger and longer-lived, it is becoming increasingly common for ontologies to be developed in distributed environments by authors with disparate backgrounds. Ontologies that are expected to be collaboratively created and maintained over time by authors in many locations present special challenges to the problem of conceptual modeling. In this paper, we will discuss conceptual modeling issues and focus on those topics with elevated importance in distributed environments. We will draw on our experience creating and maintaining ontologies in differing knowledge representation and reasoning environments over the last decade. Many of our recent observations are drawn from our experiences in the DARPA High Performance Knowledge Base Program. This program generated dozens of knowledge bases authored by people of varying expertise in both knowledge representation and reasoning as well as domain experience. Our efforts in merging the ontologies, loading them for coordinated use, and modifying them to meet evolving needs shape much of the material in this paper. Additional sources of observations are from designing and building a number of e-commerce ontologies (with content merged from multiple sources) and also from a few families of description logic applications including the PROSE/QUESTAR family of configurators and the FindUR knowledge-enhanced search applications.

1 Introduction

In recent years ontologies have become the subject of interest in communities beyond just those of knowledge representation and library science. They moved out of the research labs and are included in most expert system applications in the 80s and 90s and many of them supported sophisticated inference as well as retrieval. Many times these ontologies were built by people highly trained in knowledge representation and reasoning. Many times the author(s) also became highly literate in the domain as well. Although challenging, conceptual modeling for these kinds of situations has been studied and literature exists on proposed strategies for conceptual modeling for projects in which the knowledge acquisition is a fairly centralized process.

Taxonomies, sometimes faceted, have also been the domain of library science for a number of years. They have been successfully developed and maintained by trained staff. One of the more famous examples is the Dewey Decimal System, which is still in use today over a century after it was conceived, and has an active staff associated with it [Vizine-Goetz and Mitchell, 1996]. We will not focus in this area where trained library scientists do their classification work but note that just as the knowledge representation community is now reaching out to the non-trained public, the library science community is also reaching out to "non-catalogers" (such as the work by the Dublin core). Arguably, much of this is driven by the ubiquity of the web and efforts such as the resource description framework (RDF) [Brickley and Guha, 2000].

Recently large ontologies have become more common in broad consumer applications ranging from search (such as Yahoo, Lycos, etc.), to e-commerce and auctions (such as Amazon, EBay, etc.), to configuration (such as Dell, PC-Order, etc.), to more general information sites (such as cnet.com). Sometimes the larger ontologies are so broad that, almost by their nature, they are best designed and maintained in a distributed manner by multiple experts. Sometimes the vocabularies are just simple taxonomies, but many recent background knowledge organizations present structure in the form of properties (such as wine properties in Virtual Vineyards or electrical component properties in specification search on necx.com). It is this class of ontologies that we will be considering in this paper – those where one expects the ontologies to be created and maintained by a staff and possibly not be highly coordinated in its evolution. We will include both the simple concept taxonomies as well as ontologies with structure and many inter-relationships.

One motivational model is the Open Directory project by DMOZ where the goal is to build an "Internet brain" by leveraging the expertise of many experts. (At publication time, there were over 25,000 registered experts, which is up 25% in the last four months, over 250,000 categories, and over 1.75 million sites.) This may be at the extreme end of the spectrum since the ontology is fairly simplistic in nature and enormous in scope, but there are a number of e-commerce and other ontology efforts that have "cybrarian or ontologist" staffs in the dozens and content areas that are very broad. It is our speculation from empirical observation that ontology staffs will continue to grow, the training of entry-level staff will broaden, and ontologies will continue to become more ubiquitous. Given these speculations, the role for ontology environments in distributed settings is becoming more critical.

1.1 Three Motivational Application Areas

Since we believe that expected and actual ontology usage heavily impacts resulting ontology design, we feel it is instructive to provide a context from which to judge our observations and conclusions. This paper draws largely on three somewhat distinct ontology efforts including a decade of description logic applications (and description logic system and environment design and development), co-directing the Stanford University Knowledge Systems laboratory's high performance knowledge base program (and its frame system environment design and development), and commercial consulting in ontologies. We begin, of course, with rather extensive training in knowledge representation and reasoning.

The description logic applications fall broadly into a family of configurators (used by AT&T and Lucent to configure transmission equipment [Wright, et. al, 1993, McGuinness and Wright, 1998a, 1998b], McGuinness, et. al, 1995]), data mining applications [Selfridge, et. al., 1992], (a successor application was marketed by NCR), and ontology-enhanced search applications and environments (used in online calendars [Fuoss, et. al], electronic yellow-pages [McGuinness, 1998], AT&T World-net, and medical applications [McGuinness,1999a]). Additionally of course, there was active development on the representation and reasoning system [Borgida, et. al, 1989, Brachman, et. al, 1990] and the supporting environment for use which led to work on pruning languages [McGuinness, 1996, Baader, et. al, 1999, Borgida and McGuinness, 1996], explaining reasoning systems [McGuinness, 1996, McGuinness and Borgida, 1995, Borgida, et. al, 1998], and usability and environment work [McGuinness and Patel-Schneider, 1999, Brachman, et. al., 1999].

These applications typically had fairly intricate structure of the underlying representation, and at least in the configurators, had extensive use of the reasoner. Their representation in a description logic provided the opportunity for knowledge engineers to input term definitions with precise semantics. Most of the knowledge engineering was done in a structured and "project-managed" fashion so that it was possible to train people about how to build knowledge bases and how to extend them. Environments were built that supported knowledge engineers in building their own new applications so that knowledge representation experts were not required to do much work when new configurators were added (nor when new content areas were added within the framework of the knowledge-enhanced search applications). The data mining applications also had fairly intricate and inter-related object structure. The object structure was used to connect (and essentially to join) a number of legacy database systems. The inference level was not as extensive as that of the configurator applications. The main goal was to provide data analysts with natural and modular access to the data to support their quest for finding patterns in the data. The knowledge-enhanced search applications had the simplest modeling and reasoning structure. They were initially motivated by AT&T's Personal Online Services' browsing and search needs. They were expected to be used by a very broad user base. The application scope expanded to include more targeted user populations in narrower domains as well as broad user communities in wide domains. These applications hold the greatest similarities to the ontologies we observed in recent commercial ontology applications

The work on the High Performance Knowledge Base program [Cohen, et. al, 1998, MacGregor and McGuinness, 1999] used the Stanford ontology environment including Ontolingua [Farquhar, et. al., 1996], OKBC [Chaudhri, et. al, 1998], KIF [Genesereth and Fikes, 1992], and Chimaera [McGuinness, et. al, 2000a, 2000b] to build and maintain many large knowledge bases for intelligence analyst usage. It had a much more distributed nature than the work previously mentioned. The source documents appeared to be generated by multiple experts (usually not trained in knowledge representation). The knowledge engineers were distributed throughout the country and did not always have a lot of interactions with each other or with the original authors of the source documents. The knowledge representation experts were also distributed and sometimes time pressure did not allow them to coordinate their conceptual models as much as they would like. Additionally, ontologies were many times written by a single author or group for one purpose and then picked up by other groups for other purposes in the same content area.

The ontologies typically had fairly intricate structure both in terms of including objects with dozens of properties and many inter-relationships and also in terms of having fairly deep hierarchies. The controlled vocabularies of the upper level ontologies had significant size as well. Inference was sometimes complicated and also many times required common sense reasoning in broad areas (such as geographic reasoning over much of the world). Thus, these ontologies required breadth and depth in most dimensions that one would usually consider when measuring ontology complexity.

The work consulting on commercial ontologies [McGuinness, 1999b] is less academic but has been quickly deployed and thus has more recent empirical basis. The time frames common in startup applications allow (and force) quick deployment and evaluation of ideas. The ontologies were sometimes designed on paper and developed in house using internal (sometimes proprietary tools), sometimes designed and implemented using a combination of Stanford tools and CLASSIC and sometimes handed over to be maintained internally, and sometimes maintained through consulting. The ontologies for consulting purposes tended to contain less intricate structure and less nesting but sometimes had significant depth, breadth, and total size. In a number of commercial ontologies, it was important to include existing "standard" published vocabularies as a portion of the final ontology.

2 Distributed Ontology Desiderata

We will now present some general guidelines abstracted from the previously mentioned ontology experiences. For every principle, we will attempt to motivate its need and benefits. We will also include a discussion of some of the related challenges and include a list of operational guidelines.

2.1 Incorporate Standard Concept Vocabularies (and capture their semantics)

A goal is to include industry standard vocabularies that are familiar to the major classes of users of the system. Assuming the ontology will evolve, those classes of users include both the knowledge engineers as well as the users of the application. While it is important in centralized ontologies to use standard terms, it becomes more critical as ontologies become larger, more difficult to browse, more difficult to remember, and more complicated to extend and maintain.

Essentially every current generation set of principles for building an ontology includes articulating the expected uses of the ontology. We would add to this, the notion of articulating the expected user profiles (including both knowledge engineers assuming the ontology will evolve and also the application users). In the user articulation, there should be some characterization of what (if any) controlled vocabularies are expected to be familiar. Also, information about the expected ambiguity of an ontology should be included. For example, do the vocabularies provide necessary and sufficient conditions for membership in a class? Do they provide examples of class membership (and possibly examples of non-membership)? Are the ontologies just hierarchies of potentially ambiguous terms? Additionally, it should be noted if incoming data is likely to be consistent (or tagged) with a particular vocabulary. For example, if one is beginning a new e-commerce business-to-business application, one should consider how emerging standards like the UN/SPSC might interact with the resulting ontology. If, for example, any major sources of input data are expected to use a controlled vocabulary, then knowledge engineers will have an easier job if the application ontology is either compatible with or already includes the controlled vocabulary. Similarly, if end users are expected to be in environments where they are using terms from a controlled vocabulary, then they will have more intuitive search and browsing experiences with the application if those terms are recognized by the new application ontology.

If the application ontology contains all of the notions included in a user's vocabulary yet it uses different term names, then it will slow the user down and sometimes will make the user ineffective. For example, if the user expects to see "car" in the ontology and thus does a search for the string "car", he or she will fail to find cars in an ontology that uses only the terms "vehicle" and "auto" (if a standard syntactic retrieval is performed). There are a number of coping strategies for multiple names for the same notion such as thesaurus incorporation, query expansion, browsing support, translators, etc., but before they all end up as requirements to the system design, it is worth determining if a different base ontology would help solve a number of problems. In a growing number of areas, there appear to be some well-designed candidate ontologies that are becoming better supported and accepted. Examples include SNOMED and UMLS in medicine, UN/SPSC in e-commerce, etc.

Another issue related to standard vocabularies is that not only are they emerging, many different vocabularies are emerging as viable options. As the field of common ontologies sorts itself out, one can expect to feel compelling needs to support compatibility with multiple controlled vocabularies. This will lead to another principle discussed below concerning environmental support for interacting with and merging multiple ontologies with systematic environmental support tools.

As ontologies become larger, browsing becomes more difficult. For example, in the high performance knowledge base program upper level ontology, there exist thousands of terms and it is not always easy to find the term for which one is looking. Even after some "rationalization", undoubtedly, there are remaining issues in terms of alignment that should be handled. In our merging effort (required to load many independently developed ontologies), we discovered a number of missing sub-class links. For example, we found that "strait" was defined to be a "body-of-water" and that "narrow-body-of-water" was also defined to be a sub-class of "body-of-water", yet there was no sub-class relationship between "strait" and "narrow-body-of-water". We hypothesized, and later confirmed, that what was apparently a missed sub-class relationship was due to the fact that the author of the "strait" term was unaware of the previously defined term "narrow-body-of-water".

In order to solve this problem, one cannot do simple synonym matching. Here an explicit representation of the semantics of the terms would be useful. If one had a precise definition, say that narrow bodies of water have a width of less than two miles and straits have widths of less than a mile, then in a classification system such as a description logic, a reasoner could automatically recognize that one is a sub-class of another. Still, even without precise definitions and classifiers, additional support can be provided. For example, in the initial (less specific) encoding of strait as a "body-of-water" (but not a "narrow-body-of-water"), a sibling analysis would note that "strait" is a sibling to "narrow-body-of-water" since they are both direct sub-classes of "body-of-water". Sibling analysis would then ask if the new term – "strait"- can be made a sub-class (or super-class) of its current siblings and also if it is of the same level of specificity as its siblings.

Additionally, some representation languages allow knowledge engineers to represent classes that are disjoint from each other. Such classes may be considered members of a disjoint partition. In ontologies that contain partition or covering information, questions should be asked as to whether the new class should be added to any partitions under its parent class. For example, consider an ontology that contains a partition under the class "weapon" that distinguishes between the disjoint classes "biological-weapon" and "chemical-weapon". An addition of a new sub-class of weapon, say "nuclear-weapon" should trigger questions such as

- Should "nuclear-weapon" be either a super-class or sub-class of "chemical-weapon" or biological-weapon".
- If "nuclear-weapon" is accurately a direct sub-class of weapon, then is it disjoint from the other sub-classes of weapon, and should it be added to the weapon class partition.

Some operational guidelines include:

- Articulate anticipated ontology usage as well as expected user profiles.
- Use a controlled vocabulary that is familiar to users if one exists.

- Specify mappings between multiple standard controlled vocabularies if multiple vocabularies are standard in the domain(s) of interest.
- Allow for user extensibility of the mappings (thus, support users in adding new synonyms into a thesaurus).
- Allow for controlled vocabulary extensions.
- Provide supporting mechanisms such as query expansion in search to help facilitate sub-class matches.
- Specify semantics of terms.
- Provide semantic retrieval instead of just syntactic retrieval.
- Provide additional structural retrieval methods such as sibling retrieval and analysis.
- Provide partition extensibility.

2.2 Incorporate Standard Property Vocabularies (and capture their semantics)

As we mentioned in the introduction, it is becoming increasingly common for web ontologies to include inter-relationships. There has been more effort dedicated to developing standard hierarchies of classes (or noun phrases) than there has been on generating standard groupings (or hierarchies) of properties (or relationships between objects). Still, there are a number of organizations such as NIST, UN (in the UN/SPSC effort), NLM, etc., which are attempting to generate standard vocabularies of property names. Thus, we observe that even today a number of standard role vocabularies exist and we speculate that their growth will accelerate.

We also note that just as there are issues with retrieving classes from large and unfamiliar ontologies, there are also (at least as many) issues with finding role names in the same ontologies. All of the same problems exist and there may be less synonym support for simple synonym matching or query expansion. We have observed many times that role proliferation can quickly become a problem in large knowledge bases and we speculate that one major reason is because knowledge engineers are not finding existing useful roles. For example, a knowledge engineer who does not know that shipping-weight is already in the ontology may add a new role called weight. If the knowledge engineer did not know that shipping-weight existed, then there would be no connection between weight and shipping-weight. Now both terms will be in the ontology and with time, one could expect that some knowledge engineers find weight and thus fill in that role (without filling in shipping-weight) and other knowledge engineers will find shipping-weight and fill in that role (without filling in weight) and some may manage to fill both in (possibly inconsistently). Thus, some objects will have weights but not shipping weights and vice-versa. Possibly, even more problematically, there will be no enforceable constraints between the two roles. (Using commonsense, a shipping-weight would be expected to be equal to or slightly higher than the actual product weight. So if an object had an actual weight of 3 pounds and a shipping weight of one pound, then there would seem to be parts missing and thus an inconsistency.) If there is no relationship between the roles, this inconsistency could not be detected and we would also have no way of letting the system do some work of deducing fillers or at least lower bounds on fillers for object shipping-weights (or possibly estimates of actual-weights) for users.

When we reviewed a number of knowledge bases that contained large numbers of concepts and roles, we found this role proliferation and lack of connection between related roles to be extremely common. Another connection between roles that was missed routinely was an inverse relationship. For example, one source of data may include people, countries, and information about which people are leaders of countries. Thus, an ontology might include a property named "leader-of-country" and in the case of the individual "Clinton", "leader-of-country" would be filled with "United-States". Similarly another data source may include the property "has-leader" that has a domain including countries or possibly even more specifically a property with the singular domain of countries that might be called "country-has-leader". If there were no inverse relationship stated between "leader-has-country" and "country-has-leader" then it would appear that an application did not know who the filler of "country-has-leader" is on the United States even though it knows that United States fills the role "leader-has-country" on Clinton. In our analysis of a number of knowledge bases designed for simpler purposes and later extended for broader usage, we found this kind of role inconsistency and lack of inverse relationships to be quite common.

We also found that objects can quickly gather many properties in ontologies but many times only a few of the properties are useful for a particular purpose. It is not uncommon in our configuration examples and also in our data mining examples, for objects to have hundreds of properties. It is also not uncommon for the user to be only interested in a few property values. Large and complicated objects typically require some sort of pruning mechanism to help users focus their attention on the aspects of the object presentation that is relevant to them. Some knowledge representation systems like CLASSIC[Brachman, et. al., 1991] include a compositional language extension that allows knowledge engineers and users to specify (context-sensitive) matching patterns to be used when displaying or explaining objects. Simpler approaches include simple markup languages or just flags that tell the system if it should display portions of objects.

Some operational guidelines include:

- Use a controlled vocabulary that is familiar to users if one exists.
- Specify domains and ranges of roles (for example, the domain of "leader-of-country" is "person" and the range is "country").
- Specify inverse relationships between roles (for example, "leader-of-country" is the inverse of "country-has-leader").
- Specify active inferences to infer constraints between role values (for example, "shipping-weight" is greater or equal to the "actual-weight").
- Specify conversion rules for presenting different views of fillers for properties (for example, provide a rule that can calculate a filler for the price role in German Marks if a price filler is known in United States Dollars and a multiplier is available).

- Provide some sort of markup language to allow knowledge engineers and users to prune out roles for certain presentation (and explanation) views.

2.3 Utilize (or Develop) Environments to Support Ontology Evolution

As ontologies become larger and more structure is put in place, it becomes increasingly important to use tool environments to enforce consistency and also to aid users in focusing their attention in the areas where they are needed to resolve problems. It seems to be a given that most large ontology applications will end up needing to merge multiple ontologies together given that most specifications will require supporting more than one standard vocabulary. Merging small ontologies may not be difficult to do manually, but once ontologies become large, it becomes more critical to provide systematic tool support. Merging tools should identify terms that clearly should be merged (i.e., objects with the same names with the same internal structure and semantics), terms that may be merged (i.e., terms with names that are known to be synonyms for each other with internal structure that is not incompatible), terms that may not be merged (i.e., terms that are known by their definition to be disjoint).

Merging tools also can be used to focus the attention of the user into areas that are likely to require human re-work. For example, if there is a term that appears as a suffix of a number of other terms, it may be the case that sub-class relationships should exist. For example, if one ontology includes the term “weapon” and another ontology includes the terms “biological-weapon” and “chemical-weapon”, it is a likely guess that “weapon” should be a super-class of both “biological-weapon” and “chemical-weapon”. Merging tools may also include the sibling and partition analysis that we mentioned in Section 2.1.

Verification and validation of ontologies is also an increasingly important task, particularly when ontologies are generated as the result of merging two or more sources. When ontologies become too large for experts to scan, they need support in identifying problem areas. There are a number of things that can be done automatically. For example, authoritative sources, if they exist, may be used to check data when it is input from multiple sources. General deductions may be enforced such as “no element may be an instance of more than one class in a disjoint partition”.

Another portion of a diagnostic tool is a more subjective analysis. For example, it is rarely the case that cycles should exist in class graphs. For example, it is unlikely that we would want to say that a cat is a mammal, a tiger is a cat, and a mammal is a tiger. (Just as an empirical observation, cycles do show up in a number of ontologies currently deployed on web sites today as we discovered from crawling a number of sites. It is our analysis, that most, if not all of the cycles we found can be and should be broken.) Cycles can be identified and brought to the attention of the user. Similarly, there are many “rules of thumb” that can be used to “critique” an ontology. For example, typically one finds multiple sub-classes of a super-class. If a class has one and only one sub-class, it is usually an indication that that portion of the hierarchy has not been completed. A note in a log of all of the single child classes can be useful for knowledge engineers in focusing their attention on the areas of the taxonomy that are likely to be incomplete

An environment should provide support for multiple versions. In the simplest case, applications will have the ontology that is currently deployed and the one that is under construction. More typically, in a distributed environment, there may be multiple sub-portions of the development ontology undergoing simultaneous development.

An environment should include support for extending both high level and lower level ontologies. That may be best done by a combination of social and programmatic support. As ontologies become larger, systematic and consistent naming and organizational rules become more important. Simple naming rules can help quite a bit. In many of our larger applications we used consistent naming conventions such as prefixing all roles with "has-" so that a quick glance would determine if "mother" is the class of female persons who have at least one child or if "mother" is the binary relation between children and their mothers. If "mother" is the class and "has-mother" is the binary relation, then there is no ambiguity in the name. Environments that are used to generate ontologies can easily support and enforce naming conventions.

Many conventions may be useful and/or required. For example, if one is conducting business in a state that taxes differently on luxury vs. non-luxury items, then it is critical for all products for sale to have a way of figuring out if they are luxury items or not. Otherwise, pricing cannot be done on orders. Required properties can be clearly stated in a published ontology construction methodology and also may be enforced by ontology editing and analysis environments. We are exploring options of supporting required naming conventions in commercial ontology environments.

Some operational guidelines include:

- Incorporate an ontology merging tool into the environment.
- Employ focus of attention techniques to help assist humans in their analysis tasks.
- Provide a diagnostic tool set that analyzes provably incorrect information.
- Provide a diagnostic tool that suggests possible problems.
- Provide a critique analysis that suggests potentially better representation style.
- Provide a version control mechanism.
- Provide support for extending upper and lower level ontologies.
- Make conventions explicit on how the ontology is constructed and how one should make extensions – including naming conventions, organizational principles, what new levels mean (and when they should be added or deleted), when partitions are used and how they are added/modified), etc.

3 Discussion

Others have preceded us in the areas of conceptual modeling. We embrace most of the guiding principles laid down in the fields of conceptual graphs, description logics, general frame systems, object-oriented modeling, knowledge acquisition, etc. for building consistent, and principled ontologies. We are also not the first to focus in the areas of conceptual modeling for large or distributed environments (consider the work done at Cycorp or the work of [Swartout, et. al, 1996, Erikson, et. al, 1999, Fridman, et. al, 1999, Oliver, 1999], etc.). We may have had a wider range of experiences in our ontology environments including multiple description logic environments along with frame system and theorem provers and also may have had broader ranging needs across the application families. The work has been driven by large corporations, such as AT&T, Lucent, NCR, and recently Cisco, Internet startups, and government-funded academic research. Our recent work on merging and ontology diagnostics attempts to synthesize the learnings across all of the domains and may provide a unique combination of facilities aimed at today's world of extensible, distributed ontologies with emphasis on diagnostics and ontology critiquing. Our goal with this paper is to provide a focus for the evolving field of distributed ontology management. Another goal is to attempt to encourage the publication of suggested protocols for distributed ontology generation and maintenance.

4 Summary

In this paper, we have focused on issues that have greater importance when one is managing ontologies in a distributed manner. We have attempted to synthesize our guidelines from a wide range of applications developed in varying knowledge representation and reasoning environments with a breadth of ultimate application goals. We presented some structure in guidelines for conceptual modeling in distributed environments and welcome evolutionary suggestions to help refine the desiderata.

Acknowledgements: Our work on ontologies for government analysts was supported by the DARPA High Performance Knowledge Base program and built on the shoulders of much previous work done by many researchers at Stanford (particularly Fikes, Rice, and Farquhar), SRI, and at other HPKB contractor's organizations. Our work on description logic applications was supported by Bell Laboratories and AT&T Laboratories Research and also built on previous work by our main collaborators: Borgida, Brachman, Patel-Schneider, and Resnick. Our work on knowledge-enhanced search was funded by AT&T and was heavily influenced by our many applications partners (particularly Resnick, Beattie, Manning, Fuoss, Solomon, Moore, and Maulitz).

This paper is dedicated in loving memory to my father, Richard Charles McGuinness, who passed away during the preparation of the paper. His consistent love, support, and optimistic spirit were a key to my success in work and in life. His joyous spirit lives on but his physical presence will be sorely missed.

References

1. Franz Baader, Alex Borgida, Ralf Kuesters, and Deborah L. McGuinness. "Matching in Description Logics". In Journal of Logic and Computation -- Special Issue on Description Logics. Volume 9, number 3, June 1999.
2. Alex Borgida and Deborah L. McGuinness. "Inquiring about Frames." In Proceedings of Fifth International Conference on the Principles of Knowledge Representation and Reasoning Cambridge, Massachusetts, November, 1996. Morgan Kaufmann. Also appears in Proceedings of International Workshop on Description Logics, Cambridge, Mass., November 1996.
3. Alex Borgida, Enrico Franconi, Ian Horrocks, Deborah L. McGuinness, and Peter Patel-Schneider. "Explaining ALC subsumption" Proceedings of the International Workshop on Description Logics - DL-99, pp. 33-36, Linköping, Sweden, July 1999.
4. Ronald J. Brachman, Peter G. Selfridge, Loren G. Terveen, Boris Altman, Alex Borgida, Fern Halper, Thomas Kirk, Alan Lazar, Deborah L. McGuinness, Lori Alperin Resnick. "Knowledge Representation Support for Data Archaeology." In Proceedings of the First International Conference on Information and Knowledge Management, November, 1992.
5. Ronald J. Brachman, Alex Borgida, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lori Alperin Resnick. "Living with CLASSIC: When and How to Use a KL-ONE-Like Language." In Principles of Semantic Networks: Explorations in the representation of knowledge, ed. John Sowa. San Mateo, California: Morgan Kaufmann, 1991, pages 401--456.
6. Brickley and Guha, editors. Resource Description Framework (RDF) Schema Specification 1.0 W3C Candidate Recommendation. March, 2000. http://www.w3.org/TR/2000/CR-rdf-schema-20000327/.
7. Vinay K. Chaudhri, Adam Farquhar, Richard Fikes, Peter D. Karp, & James P. Rice. "OKBC: A Programmatic Foundation for Knowledge Base Interoperability". Proceedings of the Fifteenth National Conference on Artificial Intelligence; Madison, Wisconsin; July 26-30, 1998. Also, KSL Technical Report KSL-98-08.
8. Paul R. Cohen, Robert Schrag, Eric Jones, Adam Pease, Albert Lin, Barbara Starr, David Easter, David Gunning and Murray Burke. 1998. The DARPA High Performance Knowledge Bases Project. In Artificial Intelligence Magazine. Vol. 19, No. 4, pp.25-49.
9. DMOZ – Open Directory Project. http://www.dmoz.org.
10. H. Eriksson, R. W. Fergerson, Y. Shahar, & M. A. Musen. Automatic Generation of Ontology Editors. Twelfth Banff Knowledge Acquisition for Knowledge-based systems Workshop, Banff, Alberta, Canada, 1999.
11. Adam Farquhar, Richard Fikes, and James Rice. "The Ontolingua Server: a Tool for Collaborative Ontology Construction". In Proceedings of Knowledge Acquisition Workshop, Brian Gaines ed. Banff, Canada 1996.
12. N. Fridman Noy and Mark A. Musen. SMART: Automated Support for Ontology Merging and Alignment. In the Proceedings of the Twelfth Workshop on Knowledge Acquisition,

Modeling and Management. Banff, Canada, 1999. Available as SMI technical report SMI-1999-0813
13. Mike R. Genesereth and Richard Fikes. “Knowledge Interchange Format”, Version 3.0 Reference Manual. Knowledge Systems Laboratory, KSL-92-86, 1992.
14. Robert MacGregor and Deborah L. McGuinness. “DARPA's High Performance Knowledge Base (HPKB) Program”. Proceedings of the International Workshop on Description Logics - DL-99, Linköping, Sweden, July 1999.
15. Deborah L. McGuinness. Ontology-enhanced Search for Primary Care Medical Literature. In the Proceedings of the International Medical Informatics Association Working Group 6-Medical Concept Representation and Natural Language Processing Conference, Phoenix, Arizona, December 16--19, 1999.
16. Deborah L. McGuinness. “Ontologies for Electronic Commerce”. Proceedings of the AAAI '99 Artificial Intelligence for Electronic Commerce Workshop, Orlando, Florida, July, 1999.
17. Deborah L. McGuinness. “Ontological Issues for Knowledge-Enhanced Search". In the Proceedings of Formal Ontology in Information Systems. June 1998. Also published in Frontiers in Artificial Intelligence and Applications, IOS Press. Washington, DC, 1998. Available from http://www.research.att.com/~dlm/findur .
18. Deborah L. McGuinness. “Explaining Reasoning in Description Logics". Ph.D. Thesis, Rutgers University, 1996. Technical Report LCSR-TR-277.
19. Deborah L. McGuinness and Alex Borgida. “Explaining Subsumption in Description Logics." In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, August, 1995.
20. Deborah L. McGuinness, Richard Fikes, James Rice, and Steve Wilder. “An Environment for Merging and Testing Large Ontologies”. In the Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, USA 12-15 April 2000. Available from http://www.ksl.stanford.edu/software/chimaera.
21. Deborah L. McGuinness, Richard Fikes, James Rice, and Steve Wilder, ``The Chimaera Ontology Environment," To appear in the Proceedings of the The Seventeenth National Conference on Artificial Intelligence (AAAI 2000), Austin, Texas, July 30 - August 3, 2000.
22. Deborah L. McGuinness and Peter F. Patel-Schneider. “Usability Issues in Knowledge Representation Systems". In Proceedings of the Fifteenth National Conference on Artificial Intelligence, Madison, Wisconsin, July 1998. This is an updated version of “Usability Issues in Description Logic Systems" published in Proceedings of International Workshop on Description Logics, Gif sur Yvette, (Paris), France, September 1997.
23. Deborah L. McGuinness, Lori Alperin Resnick, and Charles Isbell. “Description Logic in Practice: A CLASSIC: Application." In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montreal, Canada, August, 1995.
24. Deborah L. McGuinness and James R. Wright. ``Conceptual Modeling for Configuration: A Description Logic-based Approach." In the Artificial Intelligence for Engineering Design, Analysis, and Manufacturing Journal - special issue on Configuration, 1998.
25. Deborah L. McGuinness and James R. Wright. “An Industrial Strength Description Logic-based Configurator Platform". IEEE Intelligent Systems, Vol. 13, No. 4, July/August 1998, pp. 69-77.
26. Diane Oliver. “Change Management for Shared and Locally Divergent Healthcare Terminologies.” In the Proceedings of the International Medical Informatics Association Work-

ing Group 6- Medical Concept Representation and Natural Language Processing Conference, Phoenix, Arizona, December 16--19, 1999.
27. Bill Swartout, Ramesh Patil, Kevin Knight and Tom Russ. “Toward Distributed Use of Large-Scale Ontologies.” In Proceedings of the Tenth Knowledge Acquisition for Knowledge-based Systems Workshop, November 9-14, 1996. Banff, Alberta, Canada.
28. Diane Vizine-Goetz and Joan S. Mitchell. 1996. "Dewey 2000." Annual Review of OCLC Research July 1995. Dublin, OH: OCLC Online Computer Library Center, Inc.:16–19. Available at: http://www.oclc.org/oclc/research/publications/review95/part1/vizine.html.
29. James R. Wright, Elia S. Weixelbaum, Karen Brown, Gregg T. Vesonder, Steven R. Palmer, Jay I. Berman, and Harry H. Moore, “A knowledge-based configurator that supports sales, engineering, and manufacturing at AT&T network systems," in Proceedings of the Innovative Applications of Artificial Intelligence Conference, pp.183--193, 1993.