Application of Ontology-Based Knowledge Representation to Design Reuse

Elisa Finnie, Eric Lai

Sandpiper Software, Inc.
{efinnie, elai}@sandsoft.com

Robert Engelmore, Adam Farquhar, Richard Fikes

Stanford University, Knowledge System Laboratory
{rse, axf, fikes}@ksl.stanford.edu

Abstract

Ontolingua, a language for ontology-based knowledge representation, provides the capability to construct comprehensive characterizations of knowledge bases. While the ability to characterize the content of a knowledge base is not new, Ontolingua includes a number of features that greatly enhance conventional data representation and modeling technologies through the incorporation of semantic context. In addition to supporting object-oriented modeling techniques, Ontolingua enables representation of constraints, definitions, and relationships among terms within ontologies. This facility provides a framework that supports automated translation among knowledge bases with differing data models and physical implementations. The ability to formally describe and unambiguously distinguish between diverse data sources is essential to enabling reuse of intellectual property. This paper presents a high-level view of ontology-based knowledge representation and an approach to solving the intellectual property reuse problem through the application of this technology.

1. INTRODUCTION

Not long ago, large-scale, homogeneous, custom data management systems were the only solution for particular problems, including those related to product data management and other design-to-manufacturing functions. When these applications no longer met user requirements, they were replaced with new homogeneous solutions -- also built from scratch, often by the original vendor in order to maintain application or data compatibility. Manual migration of legacy data was required in most cases. The process of upgrading or replacing these kinds of applications was extremely costly and time-consuming.

Today, multiple generations of systems and applications must operate together to provide adequate solutions to data management problems. A particular design, analysis, or report may require data from a number of different sources. Corporate restructuring (*i.e.*, mergers, acquisitions, and downsizing), has mandated the development of scaleable, interoperable data systems. Shortening the time to market, increasing market share, improving competitive position, improving financial results, and promoting teamwork are factors that magnify the need for solutions facilitating reuse of existing corporate knowledge and integration of intellectual property from multiple sources. Collaboration, both within and between organizations, is now an essential part of doing business. It extends beyond the classical bounds of a single corporation or enterprise, and requires a level of interoperability that remains largely unaddressed by the software development community. As a result, development of new monolithic enterprise information systems, traditionally organization-specific, requiring costly data migration, is no longer feasible or acceptable. Ontology-based knowledge representation substantially reduces the effort required to exploit existing knowledge bases and facilitates incremental data integration in new or enhanced applications and environments.

Historically, many applications were rigidly defined by their underlying database schemas and/or file formats, and users adapted, as required. Over time, forms-based client/server applications were developed, giving users limited capabilities to customize their "view" of information. While Web-based applications have revolutionized our ability to disseminate information, related customization features have yet to mature to the point where individual users can retrieve data in forms tailored to their functional requirements and personal preferences without requiring a significant programming investment. Ontologies enable flexible manipulation of information into desired forms through the creation of comprehensive data definitions and mappings.

The issues highlighted herein, including data interoperability, integration, and customization, apply to a multitude of business, product development, and engineering applications. In order to be successful, intellectual property management and design reuse applications must enable users to (1) determine whether or not something reusable or relevant to a given problem exists, (2) access the information once it has been located, and (3) integrate it into new or improved products and designs, including not only the descriptive metadata but the physical intellectual property or designs themselves. Moreover, design-to-

manufacturing data management applications must provide support for complex data types, integration with numerous special-purpose tools, and analysis, in addition to accessing diverse data sources. In many cases, sophisticated translators must be integrated with the data management capabilities to facilitate automated or semi-automated data exchange. Finally, any reuse or intellectual property management application must be easy to use, provide intuitive access to information, and present that information to users in forms that reflect both their functional requirements and personal preferences.

Ontolingua (Stanford's frame-based environment for creating and maintaining ontologies) facilitates description of the characteristics of knowledge bases in a manner suitable for automated manipulation of the underlying data (Gruber, 1993; Farquhar *et al.*, 1996). The flexibility and extensibility of Ontolingua enable information representation for a variety of domains at any level of detail. This framework provides a basis for the development and deployment of sophisticated information brokering systems that mediate the interaction between information producers, consumers, and repositories.

The Knowledge Systems Laboratory (KSL) at Stanford University has pioneered research in knowledge sharing technologies that include the development of KIF (the Knowledge Interchange Format; Genesereth and Fikes, 1992), Ontolingua, and ontology-based representation. Sandpiper Software, Inc., in conjunction with Stanford, is now applying these concepts to the development of general information brokering tools tailored to enable reuse of intellectual property supported by the Defense Advanced Research Projects Agency (DARPA)/Tri-Services Rapid Prototyping of Application-Specific Signal Processors (RASSP) program.

The purpose of this paper is to introduce the reader to ontology-based representation and illustrate its applicability to the design reuse domain.

2. ONTOLOGIES AND ONTOLINGUA

The following section provides a brief overview of the concepts, terminology, and tools available today to facilitate ontology-based knowledge representation.

2.1. Ontologies

An ontology names and describes the entities that exist in a domain, along with the predicates that represent relationships among those entities. The ontology provides a vocabulary for representing and communicating knowledge about the domain and a set of relationships that hold among the terms of that vocabulary at a logical, or conceptual level. An ontology provides a *logical* representation of an underlying database, file system, spreadsheet, or other type of structured or semi-structured data. In other words, the ontology describes information based on its *semantic context*, rather than describing it at the syntactic, or implementation level. It is precisely this distinction that enables unambiguous differentiation between multiple data sets. Given an ontology that represents each source, a common broker ontology can provide unambiguous access to multiple databases that describe similar information differently. The common broker ontology can also distinguish between multiple databases that contain fields that are named identically but are not functionally or semantically equivalent.

2.2. Ontolingua

Ontolingua (Gruber, 1993; Farquhar *et al.*, 1996) is a frame-based language enabling the construction and maintenance of ontologies, and is derived from the KIF language. The Knowledge Interchange Format (KIF) is a knowledge representation language based on first-order predicate logic and set theory. It was originally designed as an intermediate language for the translation of knowledge bases. Ontolingua provides extensions to KIF, and is sufficiently rich to represent all known concepts in the electronic systems design and manufacturing domain.

An ontology editor provides the capability to construct ontologies from libraries of pre-defined foundation components. Examples of such components include:

- quantity and dimensions (*e.g.*, weight, length, voltage)
- units of measure (*e.g.,* watt, degree, inch, US dollars)
- comparability and order
- functions (*e.g.*, data type conversions, mathematical equations; Gruber and Olsen 1994)

Where pre-defined components are insufficient or do not exist, a knowledge engineer can construct additional axioms using KIF statements. Any new or modified axioms may be saved as additional standalone components for use in other ontologies. Stanford/KSL has developed a set of pre-defined components relevant to numerous application domains. Additional components, including a baseline library that represents the RASSP Reuse Design Object Classification

Hierarchy will be defined for the RASSP program (see Section 4).

The Ontolingua development environment includes Web-based facilities for browsing and creating ontologies. These tools and core libraries of ontologies are available to users via the World Wide Web and the Internet (http://ontolingua.stanford.edu).

2.3. Key Technical Advantages

Ontolingua enables the construction of object-oriented characterizations of bodies of knowledge. It provides the capability to define classes, attributes, and relationships among them. Ontolingua facilitates the establishment of correspondences between terms across different vocabularies, as well as characteristics relevant to their interpretation. Ranges, valid values, constraints, and units of measure may be associated with standard or user-defined classes, attributes, and data types in the data model. Object-oriented class definitions permit rigid type checking, inheritance, and class-specific methods. Relationships may be established between classes, and constraints may be developed that span multiple classes.

Ontolingua was designed as a means of representing ontologies in a portable form. The ability to translate the characteristics of knowledge bases into (and out of) Ontolingua has been carefully considered, and is reflected throughout its design.

Numerous object-oriented data modeling tools are available in the commercial market today. All provide the capability to define data models in terms of classes, attributes, and relationships. Most facilitate source code generation (*e.g.,* C, C++, Java) from the data model and support maintenance of currency between them. Few, if any, however, provide the capability to specify properties and constraints associated with individual attributes in a data model. Associated properties are essential to meeting the objectives presented in Section 3.

Behavioral models that are distinct from data models may be used to define methods to *manipulate* attribute data. A number of object-oriented modeling tools provide facilities for the creation of methods. These generation capabilities, however, do not explicitly bind the methods to attributes that constrain their behavior. Without the ability to bind properties and constraints to attributes, many semantic characteristics of a given data model must be implemented in source code as part of each application that accesses that model. If these characteristics are not rigorously documented, both as a part of a comprehensive data dictionary and in the

individual applications, other related applications developed at a later date may be incomplete, misuse the data model, provide incorrect results, or in some cases actually corrupt the knowledge base. Ontolingua enables rigorous formal definition of terms, classes, attributes, and relationships, and supports the specification of properties and constraints that may be associated with any or all of these constructs.

3. WHY IS THIS IMPORTANT TO DESIGN REUSE?

Ontology-based data representation supports three key functional areas essential to intellectual property management:

- Interpretability -- the ability to exploit implicit context contained in the data.

- Interchangeability -- the ability to migrate a data set to or from a legacy system.

- Interoperability -- the ability to access data from multiple diverse data sources on-line without necessarily requiring migration of the repository.

These areas are described in detail in the following sections.

3.1. Interpretability

Historically, the semantic context for a particular database or application was rigidly defined in either the database schema or source code, or was implicitly understood by a database administrator or knowledgeable user (*e.g*., the price field is expressed in dollars). Users were required to learn how to generate specific reports, and needed substantial support if those reports changed or if new reports were required. This approach was successful, as long as the schema remained static and/or the software engineers and database administrators who designed it stayed with the organization for the lifetime of the application. Neither of these premises are valid in today's corporate environment, however. By completely defining the semantic context for all data items in terms of formal ontologies, data within a given repository may be translated to (and from) the representation desired by any end user to serve functional needs and reflect personal preferences. Well formed ontologies provide complete, unambiguous documentation of data models.

Through the use of *context logic*, ontologies provide the capability to resolve ambiguities that occur between dissimilar data sources (Fikes et al., 1996). Context logic is an extension of first-order logic in which

3

sentences are not simply true, but are true within a context. The key extension is a modality *istrue*, which takes two arguments: a context and a sentence. It asserts that the sentence is true in the specified context. An axiom that lifts sentences from one context to another context is known as a *lifting axiom*. Lifting axioms provide a very powerful and expressive means of shifting information from one context to another. They can be used to perform renaming, change structure, and make implicit assumptions explicit.

Two contexts are used to represent each information source. The *syntactic context* is a direct translation of a database schema into logic *without* resolving semantic conflicts, so that the translation can be done automatically. The *semantic context* holds the translation with the semantic conflicts resolved. The lifting axioms that perform the translation from the syntactic context into the semantic context cannot be automatically generated, because they are making the semantics that were not represented in the database schema explicit. Figure 1 shows lifting axioms to define the semantic context for the product database.

```
istrue(SemC1, product_type(x, y)) <==
  istrue(SynC1, product_type(x, y))
istrue(SemC1, $ y',z'
  (magnitude(y', natural-size-units (x))=y &
    magnitude(z', us-dollars) = z))
  <== istrue(SynC1, product(x, y, z)))
istrue(SemC1, natural-size-units(x)=bit*1024
  <== product-type(x, memory-chip))
istrue(SemC1, natural-size-units(x) = inch
  <== product-type(x, television))
```

Figure 1. Resolving ambiguity using context logic

The first axiom simply lifts all product-type facts from the syntactic context into the semantic context. The second axiom lifts tuples from the product table into the semantic context, but it disambiguates the meaning of the numbers in the table. Every number in the cost column becomes a quantity whose magnitude, when measured in US dollars, is the original number. The interpretation of size is dependent on the product type. In the case of memory chips, size is represented in multiples of 1024 bits (Kilobits); for a television monitor, it is interpreted in diagonal inches across a screen. If a user prefers to see units represented in Megabits (multiples of 1024 Kilobits) for memory chips, ontologies (a source ontology and a user ontology) can provide the definitions necessary to properly compare or display data.

A similar data set in a Japanese repository might include memory chips in Megabits, screen size in centimeters, and price information in Yen. In such a case, ontologies allow values in each data set to be correctly interpreted and, if necessary, converted to units that can be compared, manipulated, or displayed.

Attribute-bound properties defined within an ontology enable error-checking, validation, and normalization *(e.g.,* across multiple vendors and sources) during the data entry process. The framework can also support default or appropriate values for a query, if desired.

3.2. Interchangeability

An ontology provides a detailed mapping of the characteristics shared between two bodies of knowledge (*e.g.,* databases). An ontology serves as a vehicle that facilitates automated migration of static data sets from (or to) a legacy system. To date, the lack of formal methodologies, processes, and an appropriate logical framework to enable exchange of data with legacy or parallel systems has been one of the biggest impediments to scalability.

Ontologies facilitate the translation process by defining the entities, attributes, attribute characteristics, constraints, and relationships that comprise the mapping between two bodies of knowledge. The common vocabulary of source and target ontologies enables correlation of characteristics between two or more data sets and documents both their similarities and differences in an unambiguous way.

3.3. Interoperability

In the previous section, we introduced the concept of using ontologies to convert static data sets between systems, applications, and diverse repositories. Extensions to this technology can provide on-line interoperability as well. The key differentiating factor between interchangeability and interoperability may be performance for a transaction-oriented system, depending on other issues such as network throughput, hardware platforms, and target applications performance. Various aspects of corporate restructuring as related to integration of diverse business applications and the requirement to support reuse of intellectual property underscore the importance of data interoperability.

Interoperability enables incremental integration of dissimilar data without necessitating migration. One or more ontologies can characterize the data contained in multiple heterogeneous information sources. New and existing clients may quickly be given on-line access to

a newly integrated repository. Despite potential performance limitations related to production requirements, interoperability can enable prototyping and gradual migration of existing applications and data according to business or mission requirements. Data migration may also be postponed until a given application and related repository design are complete, and even then, can be performed incrementally.

Ontologies also provide a convenient mechanism for translation of multiple source data models into a common representation (*e.g.,* for use by a common information broker). This representation may or may not be the same as any of the physical source implementations or user views of the data. Having a common dialogue minimizes the number of point integrations required for additional data sources and applications and provides a uniform interface suitable for user- or domain-specific processing agents.

4. OPEN ISSUES

Sandpiper Software and Stanford/KSL are currently exploring a number of issues concerning the application of ontology-based representation to intellectual property reuse. Topics include the development of rich, standards-based ontologies, performance in a real-world environment, and integration of structured, semi-structured, and unstructured data.

4.1. Ontology Development

Numerous government, industry, and academic organizations have established standard taxonomies and the terminology specific to a variety of application domains. Examples in the electronic systems domain include the IEC 1360 standard taxonomy for electrical and electronic devices, extensions to this standard under development by the Pinnacles Group (a consortium of semiconductor manufacturers), and the standard taxonomy and specification currently under development by the Virtual Socket Interface (VSI) Alliance for representation of ASIC cores. These standards, where available and accepted by the industry, should be used as a basis for the development of a baseline set of ontologies that are published, made available for general use, promoted by the industry, and standardized. Such a set of rich, rigorously defined ontologies relevant to the electronic systems domain is under development for the RASSP program. This set, which defines structural, physical, performance, quality, and functional characteristics of diverse engineering design data in terms of classes, attributes, relationships, and constraints, is called the RASSP

Reuse Design Object Classification Hierarchy. Although this development effort may not be completed prior to the end of the RASSP program (July 1997), any resultant ontologies will be published and made available via the Stanford/KSL Ontolingua Server. These ontologies can serve as a strong basis for the development of additional standards-based ontologies for the product development and design-to-manufacturing communities.

Although the development of rich, well-formed, standards-based ontologies is required to enable interoperability between diverse data sources and to gain industry acceptance regardless of the application domain, it is not inexpensive. A team of knowledge engineers and domain experts must develop sets of domain-specific baseline ontologies, to distribute them to a variety of appropriate corporations and organizations for review, and to consolidate feedback. This process is similar to developing any kind of industry standard, and will require a substantial resource commitment from numerous organizations over time in order to be successful.

In March 1996, the *Ad Hoc* Working Group on Ontology Standards of the American National Standards Institute (ANSI) initiated an effort to standardize an ontology representation language. This language will facilitate construction of standard, primarily taxonomic ontologies with significant breadth (but little depth), making ontologies and foundation components from a variety of sources accessible.

4.2. Performance

To date, Ontolingua has been used primarily in proof-of-concept demonstrations for creating and maintaining ontologies, emphasizing functionality. The Ontolingua development environment generates ontologies consisting of KIF source statements (axioms) which execute on a LISP-based interpreter. Unfortunately, interpreters cannot always deliver the level of performance required for enterprise-scale systems with large numbers of users and high transaction rates.

Conversion of the KIF "source code" to an executable form will be necessary in order to achieve the desired level of performance. Stanford/KSL has developed tools to translate KIF into the Common Object Request Broker Architecture (CORBA), Version 2.0, Interface Definition Language (IDL), as well as several other knowledge representation languages. These formats and languages may not provide an adequate basis for automated ontology development from legacy sources, or for an internal representation of KIF capable of

supporting a network-based information broker in an environment with multiple, distributed databases, however. Tools that support incremental integration of ontologies and new data sources, synchronization of configuration information (*e.g*., directory services, physical source data characteristics), query synchronization, and so forth, will also require an optimized, executable form of KIF. Investigation of various technologies that can support execution of the KIF language is underway.

4.3. Data Integration

At present, tools to facilitate generation of basic ontologies (*i.e.,* including basic structural information) from CORBA IDL, certain types of standards documents, entity-relationship diagrams, and database schemas have been developed or are under development by Stanford/KSL. These tools must be productized and augmented with others that can generate ontologies and foundation components from a variety of structured, semi-structured, and unstructured data sources (*e.g*., paper and on-line documents, drawings, Web pages, databases, spreadsheets, multimedia data) in an automated or semi-automated manner. While the tools certainly increase the efficiency of the ontology extraction process, significant effort may still be required to restructure and augment the basic ontologies to create complete, well-formed ontologies suitable for use by an information broker. Also, as most legacy intellectual property is currently managed in relational databases, spreadsheets, and unstructured forms, methods for extraction of ontology information and integration of the individual data items must be efficient, easy to use, and expedite data migration (where required).

5. SUMMARY AND CONCLUSIONS

Ontology-based technologies provide a basis for the development of advanced information management solutions supporting data interoperability, integration, and rich customization capabilities. They make significant contributions to key problems in enterprise data management and data mining, applicable not only to reuse of intellectual property, but to many product development, design-to-manufacturing, and business-related applications.

Additional effort is required to fully exploit the potential of these technologies, including both applied research and industry-wide standards promotion.

6. REFERENCES

Farquhar, A., Fikes, R. and Rice, J*., The Ontolingua Server: a Tool for Collaborative Ontology Construction*, In Proceedings, Knowledge Acquisition Workshop, ed. Brian Gaines. Banff, Canada, 1996.

Fikes, R., Farquhar, A., and Pratt, W., *Information Brokers: Gathering Information from Heterogeneous Information Sources*, In Proceedings, Florida Artificial Intelligence Research Symposium, May 1996.

Finnie, E., *RASSP Advanced Reuse Data Manager, 19-20 November, 1996*, Presented by Sandpiper Software, Inc. at the Lockheed Martin Advanced Technology Laboratories RASSP Program Review, Camden, New Jersey, November, 1996.

Genesereth, M., and Fikes, R., *Knowledge Interchange Format Version 3.0 Reference Manual*, Logic Report 92-1, Logic Group, Computer Science Department, Stanford University, June, 1992.

Gruber, T., *A Translation Approach to Portable Ontology Specifications.* Knowledge Acquisition 5 (2): 199-220, 1993.

Gruber, T., and Olsen, G., *An Ontology for Engineering Mathematics*, Technical Report 94-18, Knowledge Systems Laboratory, Stanford University, 1994.

Rice, J., Fikes, R., Piernot, P., and Gruber, T., *Using the Web Instead of a Window System*. In Conference on Human Factors in Computing Systems (CHI96): 103-110. Vancouver, CA: Addison Wesley.