Information Brokers: Gathering Information from Heterogeneous Information Sources

Richard Fikes, Adam Farquhar, Wanda Pratt
Knowledge System Laboratory
Stanford University
{fikes,adam_farquhar,pratt}@ksl.stanford.edu

Abstract

The Internet provides dramatic new opportunities for gathering information from multiple, distributed, heterogeneous information sources. However, this distributed environment poses difficult technical problems for the information-seeking client, including finding the information sources relevant to an interest, formulating questions in the terms that the sources understand, interpreting the retrieved information, and assembling the information retrieved from several sources into a coherent answer. In this paper, we describe techniques that will enable vendors and buyers to build and maintain network-based *information brokers* capable of retrieving information about services and products via the Internet from multiple vendor catalogs and data bases for both human and computer-based clients.

1. INTRODUCTION

The Internet provides dramatic new opportunities for gathering information from multiple, distributed, heterogeneous information sources. However, this distributed environment poses difficult technical problems for the information-seeking client, including finding the information sources relevant to an interest, formulating questions in the terms that the sources understand, interpreting the retrieved information, and assembling the information retrieved from several sources into a coherent answer. In this paper, we describe techniques that will enable vendors and buyers to build and maintain network-based *information brokers* capable of retrieving information about services and products via the Internet from multiple vendor catalogs and data bases for both human and computer-based clients.

The ability to obtain relevant information in a timely and cost efficient manner is central to the performance of most tasks. The widespread availability of computer-based information brokers will provide that ability by facilitating access to the broad range of information that is rapidly becoming available on the Internet. The general availability of the technology to build and maintain information brokers will enable the establishment of an industry whose primary products are computer-based network-accessible brokering services.

2. TECHNICAL BARRIERS

Effective information brokering involves a level of understanding of the information being brokered that requires the use of symbolic domain models to reason about relevance. Information brokering requires many capabilities, including

- Helping a human or computer client formulate a query in the broker's vocabulary about some class of products or services.
- Identifying information sources that are relevant to answering a query.
- Generating a plan to answer the query using the given set of relevant information sources.
- Executing the plan and integrating information from multiple sources. This involves translating the query into the information source's vocabulary and syntax, obtaining responses to the query, and translating the responses into the broker's vocabulary and syntax.
- Presenting the responses to the client. This involves explaining to a client how the response relates to the query, defining the terms used in the response, and suggesting alternative queries that may provide additional relevant information.

While some of these tasks are characteristic of many information retrieval activities, the Internet environment imposes special requirements: the need to deal with variety, change, and autonomy of both clients and information sources. The information seeking client:

- May be a human or a software agent representing a human's interests.
- Does not know the vocabulary or access methods of all the information sources.
- May not know the vocabulary or the range of services of the information broker.

The information sources:

- Are created and maintained by independent information providers.. A broker must provide value to the information providers so that they will be motivated to couple their sources to the broker.
- Have heterogeneous access methods (e.g., SQL databases, information agents, WAIS or HTTP document servers).
- May cover different domains to different degrees and use different vocabularies to model the domain.
- May be fully structured (e.g., database relations, sentences in a logic) or semi-structured (natural language documents on a document server, e-mail, indexed multimedia).
- May return information that is incomplete or irrelevant to a query.
- Are subject to change over time in both terminology and available information.

Because widespread use of the Internet is a relatively new phenomenon, current information retrieval technologies do not adequately deal with these problems.

Our main technical claim is that domain-independent information gathering schemes are limited to syntactic matching techniques and are too weak for effective information brokering. Like human brokers, effective computer-based information brokers will take advantage of specialized domain knowledge, such as:

- The terminology used to describe products in that domain.
- Functional descriptions of products that support queries from users who need products providing a specific functionality but who do not know the type of product that can supply this functionality.
- Abstractions and assumptions that will enable the agent to retrieve information that is relevant to a query.
- Methods for appropriately combining and summarizing retrieved information.

Building such brokers as ad hoc, monolithic applications will not scale, and the resulting brokers will not be able to interoperate with the new protocols and services being developed for the Internet.

3. AN INFORMATION BROKER ARCHITECTURE

We are developing a detailed architecture for network-based, domain-specific information brokers which is shown in Figure 1. The architecture includes the following modules:

Domain Model — A logical theory which describes the broker's domain of expertise. The theory specifies the broker's vocabulary of objects, relations, functions, and product classes that it uses to model the domain. The theory describes tasks that typical users of the system perform, the functions of the products in the domain, heuristics for identifying relevant information sources, etc.

Source Models — Structured descriptions of the *competence* of each information source that the broker uses. Each description includes a logical theory of the portion of the broker's domain of expertise about which the source provides information. The theory describes the source's vocabulary in which it accepts queries and provides information. A source description also includes specifications of which relations the source can provide instances of, whether the source has complete information about some relation, the cost of accessing the source, etc.

Formulator — Assists a client in the formulation of queries in the broker's vocabulary. The formulator includes:

Product Description Browser — A service for informing clients about the broker's query vocabulary and domain of expertise. The service provides a browsable product description taxonomy consisting of hierarchies of object-oriented product descriptions from the broker's domain model.

Query By Reformulation Assistant — A service which helps a client iteratively refine a query by providing example responses to portions or all of the query.

Alternatives Advisor — A proposer of alternative queries which may provide additional useful information or better satisfy a client's goals.

Planner — Formulates a plan for answering a query. The plan specifies a sequence of subqueries to find instances of a given relation or members of a given class, constraints against which retrieved instances are to be filtered, and answer composition operations to be performed on retrieved descriptions.

Executor —Answers the query by performing the query plan. It includes the following submodules:

Source Identifier — Identifies information sources that are relevant to a subquery by determining which source model contains relations or classes whose instances can be used to answer the subquery.

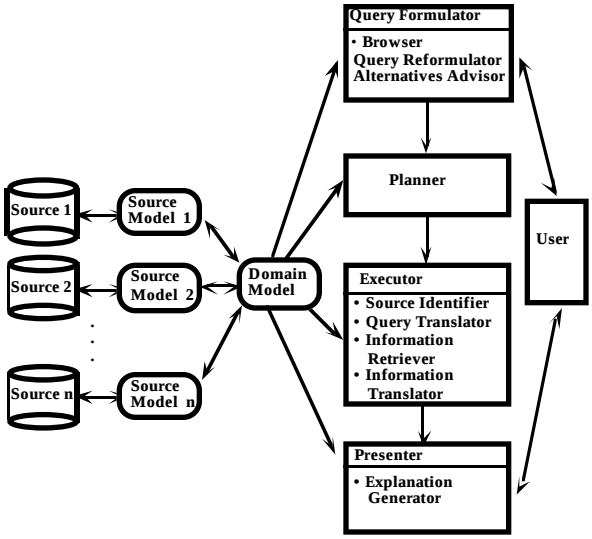

Figure 1: Information Broker Architecture

Query Translator — Translates a subquery from the broker's query language to each source's language.

Information Retriever — Obtains product descriptions by sending translated queries to information sources.

Information Translator — Translates product descriptions obtained from information sources into the broker's vocabulary and syntax.

Presenter — Presents to a client in an appropriate format the broker's response to a query. The response may include information that is known to answer the query, describes likely answers to the query, is relevant to answering the query, elaborates the answers to the query, or explains the answers to the query. The presenter makes use of an:

Explanation Generator — Explains the relevance of information presented in response to queries and the meanings of the terms in the presented information.

3.1. Domain and Source Modeling

We are developing brokers that maintain declarative, logic-based, object-oriented models of their domain of expertise and of the domains of expertise of each of their information sources. The broker's domain model specifies the vocabulary of object, relation, function, and product class names that the broker's clients can use to formulate queries. Each source description includes a logical theory of the broker's domain of expertise. The theory describes the vocabulary in which the source accepts queries and provides information.

We are developing a tool kit for broker developers based on the tools in the Ontolingua system (Gruber 1992; Gruber 1993a; Gruber 1993b). Ontolingua, which is being developed by the Knowledge Systems Lab (KSL) as part of the Knowledge Sharing Initiative (Fikes et al. 1991a; Patil et al. 1992), is an integrated tool system for developing domain-specific ontologies in the Knowledge Interchange Format (KIF) (Genesereth and Fikes 1992; Neches et al. 1991) and for translating the resulting ontologies into application-oriented representation languages. An ontology is a set of

relations and axioms that attempt to precisely characterize some domain of discourse. Ontolingua:

- Augments KIF with a frame language ontology that provides convenient representation primitives for specifying class-subclass taxonomies in an object-oriented style.
- Identifies many classes of errors in ontology specification such as under-constrained variables, undefined concepts, and missing theory inclusion relationships.
- Produces hypertext documentation for browsing.
- Provides high fidelity translation into multiple representation languages, including LOOM (MacGregor 1991), Prolog, the CORBA Interface Definition Language (IDL), and C-based CLIPS.

The hypertext documentation produced by Ontolingua is of particular importance to information brokering because it enables the definitions of product description terminology to be easily accessed by both broker developers and clients. Ontolingua generates hypertext webs of ontologies in the format of the World Wide Web (WWW) that read like reference manuals instead of source code. Putting ontologies in WWW format also makes it easy to integrate formal KIF theories with semiformal text used in documentation. For example, the introductory documents explaining the purpose and use of an ontology can have hypertext pointers directly from the use of a word in text to its formal definition in KIF. Similarly, terms used in the formal definitions can point directly to their definitions.

3.2. Domain-Specific Query Formulation Assistance

An important service that the information broker provides is assistance with the formulation of queries. For example, clients may not know or understand the idiosyncratic vocabularies used by vendors to describe the features of their products and may not know how to relate their functional objectives for the product to those product feature descriptions. The broker must use its knowledge to appropriately constrain the query and elicit information that will be sufficient to answer it. Furthermore, the broker must ensure that an answer to the query will provide sufficient information to satisfy the client's goals.

The broker architecture includes the following query formulation facilities:

- A product description browsing service that explains the vocabulary used in the descriptions.
- A query vocabulary for retrieving products by functional objectives.
- A "query by reformulation" service which helps a client iteratively refine a query by providing example responses to portions or all of the query.
- A proposer of alternative queries which may provide additional useful information or better satisfy a client's goals.

The basic facility in our broker architecture for assisting clients with query formulation is a browsable product description taxonomy consisting of hierarchies of object-oriented product descriptions from the broker's domain model. The class descriptions in the taxonomies will indicate to a client both the types of products accessible by the broker and the vocabulary that can be used in queries. The product taxonomies will be accessible as both a fully cross-referenced HTML document for browsing by human clients and as a formally defined knowledge base of structured descriptions for computer-based clients.

A broker can assist a client by providing a sample of possible answers to a query or portions of a query in situations where finding all answers to the complete query would require significant time or produce a large number of answers. The user then has the option of refining the query based on the feedback or authorizing the broker to continue retrieving answers to the initial query. If undesired answers are returned by the query, the client can examine those items to determine which characteristics would have eliminated those items. The client can use this information to appropriately refine the query. The descriptions may also suggest additional features that the client had not previously considered. Such *query by reformulation* techniques (Tou et al. 1982; Yen, Neches, and DeBellis 1988) are particularly useful for assisting clients who are unfamiliar with the descriptive vocabulary available for use in queries or with the range of information that is available for responding to queries.

An important application of the information broker's domain-specific expertise is the ability to suggest alternative queries to a client that may provide a more desirable solution to the client's goals. (E.g., consider departing from a neighboring airport when all flights are booked or extending a trip over a Saturday night to reduce the cost of the airline ticket.) In order to suggest useful alternatives, the broker needs to be able to assume or determine the client's goals and to know for a given query-goal pair what alternative queries might be useful.

3.3. Query Planning

Given a query from a client in some formal language, an information broker must decompose the query into a sequence of subqueries that can be sent to relevant information sources, constraints against which retrieved descriptions are to be filtered, and answer composition operations to be performed on retrieved descriptions. We expect each subquery to be a request for tuples that satisfy a given relation or structured descriptions of members of a given class.

We will develop a query planner to perform this decomposition by adapting existing techniques from the data base community to the information brokering environment. Because of the dynamic nature of the network environment in which information brokers will work, we anticipate a strong need for intermixing query planning and execution. That is, because the planner may not know what information sources are available to answer the query or how many answers may be found for a given subquery, the planner may not be able to form a complete plan that will be effective or efficient. We will explore strategies which execute the next steps in a query plan as soon as they are determined and then continue generation of the remainder of the plan taking into account the results of the steps already executed.

3.4. Query Plan Execution

The query plan executor answers the query by identifying which information sources are relevant to answering a subquery, translating a subquery into the vocabulary and syntax of each identified source, retrieving information by sending translated queries to each identified source, and translating the answers retrieved by an information source into the broker's

Table Name			
Product	**name**	**size**	**cost**
	television-1	19	256
	simm-1	256	8
Product-Type	**name**	**type**	
	television-1	television	
	simm-1	memory-chip	

Figure 2: Partial Contents of Database

Table Name	Field	Data Type
Product	name	char
	size	int
	cost	int
Product-Type	name	char
	type	char

Figure 3: Schema Definitions

vocabulary. We will focus on translation problems in query plan execution.

We use a context logic (Buvac and Mason 1993; Guha 1991; McCarthy 1993) to represent the schemata and views of the individual information sources, shared ontologies, implicit semantics, and integrated information. Context logic allows us to incrementally strengthen the representation of an information source. An initial representation can be generated automatically from an information source's schema. The representation can be strengthened over time by adding axioms that make explicit the assumptions behind the schema and translate from the information source's vocabulary into the terms that are shared with other information sources. Clients may query the information source as in a loosely coupled system that provides a uniform query interface to a number of heterogeneous systems.

Consider the simple relational database specified in Figure 2. Using the schema definition in Figure 3, the tables of a relational database can be translated into assertions in first order logic, as shown in Figure 4. Each table in the database corresponds to a relation in the logic and each database schema definition corresponds to an axiom.

```
(∀ x,y,z product(x, y, z)
  ⇒ string(x) & integer(y) & integer(z))
relation(product) & arity(product, 3) &
  primary-key(product, 1)
(∀ x,y,z product_type(x, y)
  ⇒ string(x) & string(y))
relation(product_type) & arity(product, 2) &
  primary-key(product_type, 1)
```

Figure 4: Representation of the product and product-type schema in logic.

Although this context represents the database schema and its contents in logic, it suffers from two basic problems:

- The representation does not resolve the ambiguities when attributes are used polymorphically within a single table. For example, the size attribute in the product database stores the size in whatever unit is appropriate for the particular product. For televisions this may be the number of inches across the diagonal of the screen, for memory chips this may be in kilobytes. A logician might consider this to be a sloppy representation, but it is typical of the representations that occur in actual databases.
- The representation allows values to have a non-unique denotation. A single value may be used in a database to mean different things in different tables, tuples, or columns. For example, in the product database of Figure 1b, the number 256 appears in both the size and cost columns. This is not, in itself, an inconsistency. If we want to make the units explicit, however, care must be taken to avoid assigning incompatible interpretations to different occurrences of 256.

These problems are related, and can be solved with a combination of existential quantification and systematic renaming. For example, we could write an axiom to disambiguate the product relation such as:

```
∀ x,y,z product(x, y, z) ⇔
   (∃ y',z' product-1(x, y', z') &
     magnitude(z', us-dollars) = z)
```

That is, we could introduce a new relation `product-1` that is similar to product, except that new objects, y' and z', are introduced to represent the size and cost. The magnitude of the cost, y', must then be equal to the number in the database product table. The existential quantification is an important, but straightforward, trick. The renaming, however, is rather clumsy, and it becomes even more awkward once multiple information sources are integrated.

Context logic provides us with a more elegant and powerful mechanism to overcome the representational problems without the clumsy renaming. Context logic (McCarthy 1993) is an extension of first order logic in which 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. Contexts are logical individuals and, as such, can be quantified over. Furthermore, it is possible to write axioms that span several contexts. 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. Context logic allows us to restate the disambiguation axiom without renaming:

```
istrue(c1, product(x,y,z)) ⇔
   istrue(c2, (∃ y',z' product(x,y',z') &
           magnitude(z', us-dollars)= z))
```

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 5 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 5: Lifting axioms between the syntactic context and the semantic context

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. Translating the size column is somewhat more complicated because the unit varies with the type of the product. The last two axioms associate a product type with the unit most naturally used to measure its size.

The author of the lifting axioms must choose an appropriate way to represent the intended implicit semantics of the database. In Figure 5, the functions magnitude and natural-size-units were used along with the constants us-dollar, bit, and inch. The decisions required to construct these representations can be subtle. For instance, we have chosen to use a magnitude function that takes two arguments: a quantity and a unit. This is because the unit used to measure a quantity is not an inherent property of the quantity, whereas its dimension is. For example, the dimension of all prices is currency, but the magnitude of a price can be measured by any unit of currency such as dollars or yen.

3.5. Results Presentation and Explanation

Effective information brokering requires presenting information obtained in response to a query in an easily understandable format and assisting the client in understanding that information. The task is nontrivial because the results may not provide precisely the information needed or intended. The query reformulation process will typically be incomplete and approximate so that the response may include information that is known to answer the query, describes likely answers to the query, is relevant to answering the query, or elaborates the answers to the query. Also, when the amount of information gathered is large, summarization will be required for human readers.

In order to enable brokers to assist their clients in understanding retrieved information, we will include in our broker architecture an explanation generation facility that can be used to provide clients with explanations of the rationale for the relevance of information presented in response to queries and of the meanings of the terms occurring in the presented information. For example, if the query is to find airline flights from London to Washington D.C. on a given date, and the retrieved information describes flights from Heathrow to Dulles, an explanation could be added to Heathrow (e.g., as a hypermedia link) saying that it is a London airport and to Dulles saying that it is a Washington D.C. airport.

We are developing tools that enable a broker to compose explanations from term definitions and from the inferences it makes, and to annotate the information it provides to a client with those explanations, either as hypertext links for human clients or as relational links for computer-based clients. The tools are based on the explanation technology we have developed in the How Things Work Project (Fikes et al. 1991b) for providing interactive documentation of engineering designs. That technology dynamically generates device descriptions and causal explanations of simulated device behavior from symbolic device models, mathematical simulation models, and simulator output (Gautier and Gruber 1993). The explanations are produced as HTML documents that are generated dynamically by a network server when a user requests an explanation.

4. RELATED WORK

There is a growing body of work that addresses the problems of gathering and integrating information from multiple heterogeneous information sources. Three relevant pieces of work include SIMS (Arens et al. 1993; Arens and Knoblock 1992), CARNOT (Huhns et al. 1992), and the work of Siegel and colleagues (Goh, Madnick, and Siegel 1994; Sciore, Siegel, and Rosenthal 1994). The work described in this paper differs in three major ways. First, the context logic provides a sound formal basis for describing the semantics of multiple sources and for articulating the relations between them. Second, a library of reusable ontologies reduces the cost of building these descriptions and helps users to understand the meaning of the terms that they can use in their queries. Third, the explanation techniques help users to understand the results of a query.

5. DISCUSSION

There are several powerful consequences of using our approach to integrate information sources including the ability to:

- Integrate new information sources incrementally.
- Share assumptions among information sources without making them explicit.
- Exploit shared ontologies.
- Provide a richer model of integration that goes beyond global schema or federated schema methodologies.

One important consequence of our approach is that it eliminates most of the up-front cost of integrating a new information source. The cost is reduced because the information source context can be automatically generated from the source's export schema. Once this has been done, it is possible to make queries in the context of the new information source as if it were a loosely coupled heterogeneous database system. The query must be expressed in the vocabulary of the new source, but the syntax and interface are consistent with the old sources. Once the information source context has been established, it is possible to incrementally add lifting axioms to populate the source's semantic context. Furthermore, this incremental integration can be performed in response to perceived and actual usage patterns, rather than expectations about usage. Our approach is in stark contrast to the global schema approach in which the ontology of the new

information sources must be completely decontextualized and translated into the existing global schema.

It is possible for a new information source to exploit commonalties with an existing source in two ways. First, we can copy lifting axioms from the old source into the new source's semantic context. Second, we can write lifting axioms that map from the new source's context into the old source's context. The key point here is that the relations between contexts are much richer than theory inclusion. Lifting axioms may connect sibling contexts and exploit commonalties in numerous ways. This is similar to, but more powerful than, the federated database approach in which the schemas of subsets of the databases known to the federation are combined. Context logic enables shared implicit assumptions to be shared across information sources without the need to first disambiguate them.

The semantic contexts make use of shared ontologies. Ontologies for domains such as quantities, finance, product descriptions, and so on, will ease the work of integrating information sources. In our formulation, each ontology is defined in its own context. The semantic contexts of information sources may then include the ontology contexts. If the semantic contexts of several information sources share common ontologies, it is much easier to operate across them. This is one way of decomposing the otherwise daunting problem of constructing a global context.

We have presented an information broker architecture that will help with query formulation, identify information sources relevant to those queries, retrieve information from various information sources, and present the responses as well as relevance rationale. Our approach focuses on using domain knowledge and context logic to provide a scalable approach to gathering information from heterogeneous sources.

6. REFERENCES

Arens, Y., C. Y. Chee, C.-N. Hsu, and C. A. Knoblock. 1993. Retrieving and Integrating Data from Multiple Information Sources. International Journal on Intelligent and Cooperative Information Systems 2 (2): 127-158.

Arens, Y. and C. Knoblock. 1992. Planning and Reformulating Queries for Semantically-modeled Multidatabase Systems. In Preceedings of the 1st International Conference on Information and Knowledge Management:92-101.

Buvac, S. and I. Mason. 1993. Propositional Logic of Context. In Proceedings of the Eleventh National Conference on Artifical Intelligence:412-419: AAAI Press/MIT Press.

Fikes, R., M. Cutkosky, T. Gruber, and J. van Baalen. 1991a. Knowledge Sharing Technology Project Overview. KSL 91-71. Stanford University, Knowledge Systems Laboratory.

Fikes, R., T. Gruber, Y. Iwasaki, A. Levy, and P. Nayak. 1991b. How Things Work Project Overview. KSL 91-70. Stanford University, Knowledge Systems Laboratory.

Gautier, P. O. and T. R. Gruber. 1993. Generating Explanations of Device Behavior Using Compositional Modeling and Causal Ordering. In Proceedings of the Eleventh National Conference on Artificial Intelligence. Washington, D.C.: AAAI Press/The MIT Press.

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

Goh, C. H., S. E. Madnick, and M. D. Siegel. 1994. Context Interchange: Overcoming the Challenges of Large-scale Interoperable Database Systems in a Dynamic Environment. In Proceedings of the Third International Conference on Information and Knowledge Management (CIKM-94). Gaithersburg, Maryland.

Gruber, T. R. 1992. Ontolingua: A mechanism to Support Portable Ontologies. KSL 91-66. Stanford University, Knowledge Systems Laboratory.

Gruber, T. R. 1993a. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In International Workshop on Formal Ontology, ed. Nicola Guarino. Padova, Italy.

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

Guha, R. V. 1991. Contexts: A Formalization and Some Applications. PhD Thesis, Stanford University.

Huhns, M., N. Jacobs, T. Ksiezyk, W. M. Shen, W. Singh, and P. Cannata. 1992. Enterprise Information Modeling and Model Integration in CARNOT. In Enterprise Integration Modeling: Proceedings of the First International Conference: MIT Press.

MacGregor, R. 1991. The Evolving Technology of Classification-based Knowledge Representation Systems. In Principles of Semantic Networks: Explorations in the Representation of Knowledge, ed. John Sowa:385-400. San Mateo, CA: Morgan Kaufmann.

McCarthy, J. 1993. Notes on Formalizing Context. In Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence.

Neches, R., R. E. Fikes, T. Finin, T. R. Gruber, R. Patil, T. Senator, and W. R. Swartout. 1991. Enabling Technology for Knowledge Sharing. AI Magazine 12 (3): 16-36.

Patil, R. S., R. E. Fikes, P. F. Patel-Schneider, D. McKay, T. Finin, T. R. Gruber, and R. Neches. 1992. The DARPA Knowledge Sharing Effort: Progress report. In Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference. Cambridge, MA: Morgan Kaufmann.

Sciore, E., M. Siegel, and A. Rosenthal. 1994. Using Semantic Values to Facilitate Interoperability among Heterogeneous Information Systems. Transactions on Database Systems 19 (2): 254-290.

Tou, F. N., M. D. Williams, R. E. Fikes, D. A. Henderson, and T. W. Malone. 1982. RABBIT: An Intelligent Database Assistant. In *Proceedings of the National Conference on Artificial Intelligence*:314-318. Pittsburgh, PA.

Yen, J., R. Neches, and M. DeBellis. 1988. Specification By Reformulation: A Paradigm for Building Integrated User Support Environments. In Proceedings of the National Conference on Artificial Intelligence:814-818.