The Information Manifold

Thomas Kirk
AT&T Bell Laboratories
tk@research.att.com

Alon Y. Levy
AT&T Bell Laboratories
levy@research.att.com

Yehoshua Sagiv
Hebrew University
sagiv@cs.huji.ac.il

Divesh Srivastava
AT&T Bell Laboratories
divesh@research.att.com

Abstract

We describe the Information Manifold (**IM**), a system for browsing and querying of multiple networked information sources. As a first contribution, the system demonstrates the viability of knowledge representation technology for retrieval and organization of information from disparate (structured and unstructured) information sources. Such an organization allows the user to pose high-level queries that use data from multiple information sources. As a second contribution, we describe novel query processing algorithms used to combine information from multiple sources. In particular, our algorithms are guaranteed to find *exactly* the set of information sources relevant to a query, and to *completely* exploit knowledge about local closed world information (Etzioni *et al.* 1994).

Introduction

We are currently witnessing an explosion in the amount of information that is available online. For example, the rapid rise in popularity of the World Wide Web (WWW) has increased the amount of information available over the Internet. As another example, large companies and institutions have a vast number of internal databases, which they are making available both internally and externally. Along with the rise in the number of information sources, there is also a growing number of systems and protocols for providing user friendly browsing of this information (e.g., Mosaic). Although browsing is an important form of obtaining information, it is limited, and often time-consuming. Previous work on enabling location of information on the WWW has focused primarily on building brokers that provide keyword based index services. Support for sophisticated querying and user-customized organization of information has been minimal.

This paper describes the Information Manifold (**IM**), an implemented system for retrieval and organization of information from disparate (structured and unstructured) information sources. **IM** clearly demonstrates the viability of Knowledge Representation technology to enable access to online information. **IM**'s architecture is based on a knowledge base containing a rich domain model that enables describing properties of the information sources. In particular, **IM**'s domain model includes the representation of topics of information sources, as well as properties having to do with the physical characteristics of the sources. The user can interact with the system by browsing the information space (which includes both the knowledge base and the external information sources). However, the presence of descriptions of the information sources also enables the user to pose high-level queries about sources, a capability that distinguishes it from current browsers.

When the external information sources are structured (e.g., databases, SGML documents), or can be viewed as partially structured (e.g., FTP sites, bibliography files), several interesting issues arise for efficient query answering. **IM**'s representation language enables describing the semantic content of structured sources in a way that can be used to answer queries that may involve accessing data in multiple sources. The bulk of this paper describes the query processor of **IM**, that answers user queries (posed in terms of the domain model), using the information sources. In particular, our techniques for answering queries make two contributions over previous related work:

- The language for representing contents of information sources is a combination of Horn rules and concepts from the CLASSIC description logic (Brachman *et al.* 1991). For this language we show it is possible to *efficiently* and *completely* determine which information sources (or portions thereof) are relevant to a given query. In contrast, previous work (e.g., SIMS (Arens *et al.* 1994)) provided no guarantees of minimality of the number of information sources (or portions thereof) deemed relevant. Furthermore, SIMS modeled information sources using only a description logic. The expressive power of Horn rules is necessary in order to model information sources that are relational databases. Furthermore, our techniques for determining relevance are sufficiently general, such that we can incorporate Horn rules with more expressive description logics, consider queries involving negation, and statements describing rela-

tionships *between* the information sources.

- Local closed world information (LCW), introduced in (Etzioni *et al.* 1994) enables us to express the fact that an information source has *complete* knowledge about some part of the domain. The query processor can use this knowledge to prune access to *redundant* information sources (i.e., sources that are relevant, but whose content is contained in the union of some other sources). First, we describe a richer language for stating LCW statements than described in (Etzioni *et al.* 1994), in which *constraint formulas* are used to describe more precisely the portion of the domain for which a source has complete information. Second, by using the close relationship between the problem of reasoning with LCW statements and the problem of determining independence of queries from updates (Levy and Sagiv 1993b), we obtain a general algorithm for inferring LCW of queries and for pruning redundant information sources.

We begin by describing the representation language used in **IM**, and how we describe properties of information sources. We then highlight the advantages obtained by using KR technology in this application. Next, we describe the query processor. Finally, we briefly describe how a user interacts with the **IM** system.

The Representation in IM

The knowledge base in **IM** is used for two purposes. First, it provides a uniform conceptual model of the domain with which the user interacts and poses queries. Second, it is used to represent properties of external information sources, and in particular, their semantic content.

Our domain consists of several rich hierarchical structures. As examples, we have a rich topic hierarchy for information sources, and the types of Internet information sources (i.e., based on the protocols, structure, etc.) form a natural hierarchy. Therefore, a KL-ONE style description logic provides a natural way of modeling our domain. However, in order to model relational databases, and relations between different databases we also need the representational power of Horn rules. Hence, the representation language used in **IM** is a combination of Horn rules and the CLASSIC description logic. The **IM** knowledge base has several components.

- A terminological component in the CLASSIC language: the terminology contains unary relations (called concepts) which represent classes of objects in the domain and binary relations (called roles) which describe relationships between objects. Concepts and roles can be either primitive or complex. Complex concepts and roles are defined via *descriptions* built from the following set of constructors.

  $C \sqcap D$ (conjunction)
  $\forall R.C$ (universal quantification)
  $(\geq n\, R) \mid (\leq n\, R)$ (number of role fillers)
  (fills R a) (filler of a role)
  (oneOf R $\{a_1, \ldots, a_n\}$) (role filler restrictions)

- A set of Horn rules: in addition to arbitrary n-ary predicates, the antecedents of the rules can contain unary literals of concepts defined in the terminological component (but not binary roles),[1] and the interpreted binary predicates $(=, \neq, <, \leq)$.
- A set of ground atomic facts for concept, role and ordinary predicates.
- A set of integrity constraints of the form:

  $$R(\bar{X}) \Rightarrow C(\bar{X})$$

  where C is a DNF formula involving concept predicates and the interpreted binary predicates, and R is an ordinary predicate.

Hereafter, a *constraint formula* is a formula with free variables $X_1, \ldots, X_n$, containing conjunctions and disjunctions of atoms of the form $C(\alpha)$ and $\alpha_1 \theta \alpha_2$, where C is a concept in the terminology, α_1 and α_2 are either variables or constants, and $\theta \in \{<, \leq, \neq, =\}$.

Example 1: Suppose our system is providing access to multiple information sources providing flight quote information and telephone directories. In that case, we can conceptualize the domain using the following relations:

- $quote(Ag, Al, Src, Dst, C, D)$, which denotes that a travel agent Ag quotes a price of C to travel from Src to Dst on airline Al on date D.
- $dir(Cust, Ac, TelNo)$, which gives the directory listing of customer $Cust$ as area code Ac and phone number $TelNo$.

The representation of the domain also has a rich hierarchy of concepts describing various types of telephone customers. The concept *customer* is a primitive concept that includes all customers and specifically the disjoint subconcepts *business* and *residential*. Each instance of a business customer has a role *business_type*, specifying the types of businesses it performs. Given these primitive concepts, we can define a concept *travelAgent* by the description:

(business ⊓ (fills business_type "Travel"))

Integrity constraints are used to specify types of the attributes of the domain relations. For example, the attribute $Cust$ of relation *dir* is constrained to be of type *customer*, the attribute Ag of relation *quote* is constrained to be of type *travelAgent* and the attribute C of *quote* is constrained to have non-negative values. □

[1] See (Levy and Rousset 1995) for an extension that also allows role predicates in the antecedents.

The **IM** knowledge base contains ontologies for representing various aspects of the domain. In particular, we represent physical properties of information sources, such as their addresses (i.e., URLs), the protocols used to access them (e.g., FTP, HTTP), and their internal structure (e.g., hypertext, relational database, knowledge base). The knowledge base also consists of a rich topic hierarchy, ontologies for representing properties of people, organizations, geographic locations and time.

Representation of Information Sources

A key component of the knowledge base is the representation of the contents of external information sources. Many sources have no internal structure and for them, **IM** only represents their physical and ownership properties, and their topics. However, many sources do have internal structure or can be viewed as having some internal structure, and this can be exploited in answering sophisticated queries. In order to do so, we need a representation of the contents of the information sources, and specifically a semantic mapping to the relations in the knowledge base. In this section, we describe how we represent such mappings in the **IM** knowledge base.

An external information source is viewed as containing extensions of a collection of relations. These relations can be either explicitly stored in the source (e.g., a database source), or computed when queried (e.g., an FTP directory can be viewed as a relation containing the set of files in a directory, which can be retrieved by an "ls" command).

For reasons explained subsequently, we restrict the form of the occurrences of relations corresponding to an information source in the knowledge base. Specifically, except for completeness information (i.e., LCW statements) described in detail later in the paper, these relations can appear only in the antecedents of the Horn rules or of the integrity constraints, i.e., they *cannot* appear in the consequents of Horn rules.

Example 2: Consider the airline flight domain. Fly-by-Night Airlines provides two external source relations: $fbn\_flights(Flt, Src, Dest)$, which denotes that flight Flt of Fly-by-Night Airlines is from Src to $Dest$, and $fbn\_quote(Ag, Flt, C, D)$, which denotes that a designated travel agent Ag of Fly-by-Night Airlines quotes a price of C to travel by flight Flt on date D. The domain relation *quote* can be related to the contents of the source relations $fbn\_flights$ and fbn_quote using a Horn rule as follows:

$$fbn\_flights(Flt, Src, Dest) \wedge fbn\_quote(Ag, Flt, C, D) \Rightarrow quote(Ag, \text{'Fly-by-Night'}, Src, Dest, C, D).$$

As an example of an integrity constraint involving an external information source relation, the New Jersey directory information source $nj\_dir(Cust, Ac, TelNo)$ can have the following integrity constraint:

$$nj\_dir(Cust, Ac, TelNo) \Rightarrow (Ac = 908) \vee (Ac = 609) \vee (Ac = 201).$$

□

Utility of KR Technology

Current tools for retrieving information from the Internet (e.g., browsers such as Mosaic, or index services such as WAIS) provide very limited representation of the information sources. In particular, Mosaic allows the caching of pointers to information sources in a flat list, and WAIS provides a keyword based lookup in a collection of documents. For unstructured sources, using a knowledge representation system enables more sophisticated representations of information sources which in turn allow us to make inferences resulting in the ability to answer more sophisticated queries. For example, instead of simply storing pointers to FTP sites of colleagues in the Mosaic Hotlist, we can represent properties of the colleague (e.g., their areas of expertise, their institution). Then we can pose sophisticated queries such as "Give me the FTP sites of colleagues whose area of expertise is software agents and who are university professors".

Inferences made by the system are also useful in the process of populating the knowledge base. For example, when finding a new information source and adding its description to the knowledge base, the system can automatically fill in some properties of the information source. As an example, by specifying some topics associated with the information source, the system can automatically place it in a hierarchy of topics represented in a domain model, which allows retrieval of this information source for related queries.

Our hybrid representation language was chosen to be as expressive as possible, yet guaranteeing that inferences can be made efficiently (Levy and Rousset 1995).

Answering Queries Using Structured Sources

One of the *key* advantages of having a declarative representation of the contents of information sources is the ability to answer queries that use facts stored in a structured information source. In this section, we discuss how the query processor of **IM** answers queries using multiple structured sources.

In our discussion, we assume that a *query* is an atom of the form $R(\bar{X})$, where R is a relation in our domain model. That is, the query asks for the tuples $\bar{a}$ that are instances of $\bar{X}$ such that $R(\bar{a})$ is entailed by the knowledge base.

Recall that the knowledge base does not store all the information about all the relations in its domain model. Some of the relations are only conceptual, and their extensions can be computed from external information sources. Therefore, answering a query proceeds as follows:

- Determine which portions of the domain relations are needed to answer the query. That is, for every relation E in the domain model, we compute a constraint formula C_E, such that the facts of E that satisfy C_E are needed to answer the query.
- Determine which portions of the information sources are needed to compute the desired portions of the domain relations. That is, for every domain model relation E, we decide which portions of the external sources are needed to compute the facts of E that satisfy C_E.
- Formulate subqueries to each of the relevant information sources.
- Combine answers from the information sources to compute the needed portions of the domain relations.
- Compute the answer to the query from the domain relations.

There are two distinct kinds of derivations performed in computing the answers to a query. The first kind of derivation uses facts from the information sources to derive facts for the domain relations that are not stored in the KB. These derivations use the rules that provide the semantic mapping between the information sources and the domain model. The second kind of derivation uses facts of the domain model relations to derive additional facts for the domain model relations, and in particular to compute answers to the query.

Since the cost of answering queries is dominated by the cost of accessing external information sources, the *key* to optimizing the evaluation of a query is to determine a *minimal* set of portions of information sources that are relevant to answering the query. We formally define relevance in stages as follows.

Recall that in Horn rule knowledge bases, a derivation can be viewed as a tree whose root is the goal node corresponding to the query Q. A goal node G is either the leaf of the tree (corresponding to a fact explicitly stored in the KB) or has a child rule node r whose children are themselves goal nodes $G_1, \ldots, G_n$ used to derive G using rule r. We say that a derivation d *uses* a fact $R(\bar{a})$ if $R(\bar{a})$ is one of the goal nodes in the tree.[2]

Definition 1: *A fact $E(\bar{a})$, where E is a relation in the domain model, is* relevant *to a query $Q(\bar{X})$ if there is some extension of the domain model relations that is consistent with the integrity constraints in the KB such that $E(\bar{a})$ is used in a derivation of some answer to the query $Q(\bar{b})$.* □

The above definition formalizes the notion of relevance for the second kind of derivation performed in the process of answering a query (i.e.,, using facts of the domain model relations in order to compute the answer to the query). In a similar fashion we can define relevance for the first kind of derivation. In doing so, E in the above definition is a relation representing an external information source relation and Q is the domain model relation, and we consider all possible extensions on the external sources that satisfy the integrity constraints in the KB. For more details on definitions of relevance, see (Levy and Sagiv 1993a).

Based on these definitions, we can define portions of external information sources relevant to a query as follows. A portion of an external information source relation R is denoted by a pair (R, C), which represents the set of all facts of relation R that satisfy the constraint formula C. As before, C is a DNF formula involving concepts from the description logic and interpreted binary predicates.

Definition 2: (Relevance): *A portion (R, C) of an external source relation R is* relevant *to a query $Q(\bar{X})$ if for every fact $R(\bar{a})$, $\bar{a}$ satisfies C if and only if there is some fact $E(\bar{b})$ of a domain relation E such that $R(\bar{a})$ is relevant to $E(\bar{b})$ and $E(\bar{b})$ is relevant to the query $Q(\bar{X})$.* □

Note that the above definition of relevance depends *only* on the descriptions of the information sources, i.e., on the rules and integrity constraints in the knowledge base, and not on the actual contents of the information sources. This definition is motivated by the need to determine relevance *without* actually accessing the external information sources, which would undermine the optimization effort.

The query processor in **IM** uses the query-tree algorithm presented in (Levy and Sagiv 1992) for determining the relevant portions of external information sources. Finding relevant portions of the external source relations proceeds in two steps. The first step determines which portions of the domain relations are relevant to the query, and the second step determines which portions of the source relations are relevant to the portions of the domain relations deemed relevant to the query.

The following theorem shows that it is possible to precisely determine the portions of the source relations that are relevant to a given query.[3]

Theorem 1: *Let $Q(\bar{X})$ be a query. For each external information source relation R_i, the query processor of* **IM** *determines precisely the constraint C_i such that (R_i, C_i) is the portion of R_i that is relevant to $Q(\bar{X})$.*

Furthermore, assuming bounded arity of predicates in the knowledge base, relevant portions of source relations are determined in time polynomial *in the size of the knowledge base.* □

The proof of the first part of the theorem follows from the observation that the constraints in our rep-

[2] Note that a fact can have multiple derivation trees, corresponding to different ways of deriving the fact.

[3] Clearly, for some source relations the relevant portion can be empty, indicating that the source relation does not contain any relevant information.

resentation language satisfy the abstract properties required by the query-tree algorithm in order to guarantee completeness. The proof of the second part of the theorem uses the result that, for the constraints in our representation language, determining whether one conjunctive formula subsumes another conjunctive formula is in polynomial time.

Example 3: Consider a query that asks for information about travel agents in Miami, FL (area code 305) who sell tickets from Newark to Santiago for under $1000. The query can be formulated as a Horn rule defining the relation Q, as follows:

$$quote(Ag, Al, \text{'Newark, NJ'}, \text{'Santiago, Chile'}, C, D) \\ \wedge dir(Ag, Ac, TelNo) \wedge Ac = 305 \\ \wedge C < 1000 \Rightarrow Q(Ag, TelNo, C).$$

None of the directory information sources that provide phone numbers for area codes other than 305 would be considered relevant. For example, no portion of the Manhattan, NY directory information source which provides telephone numbers in the 212 area code would be relevant since the conjunction of the constraints $Ac = 212$ (from the description of the Manhattan source) and $Ac = 305$ (from the query) is *unsatisfiable.* □

Extensions

Often there are many useful relationships between the information sources, e.g., containment and disjointness, that can help reduce the set of information sources that need to be accessed in order to answer a query. The results of (Levy and Sagiv 1995) can be used to compute the minimal set of sources when such relationships are described in the knowledge base. Further, the results of (Levy *et al.* 1993) can be used to determine relevant portions of external source relations if queries were extended to include negation. Finally, the results of (Levy *et al.* 1995) can be used to compute the set of relevant sources in cases where the relations representing the information sources appear in the consequents of the rules.

Completeness and Redundant Sources

In practice, information may reside redundantly in many information sources. Accessing all the information sources that are relevant to a query could thus involve retrieving information redundantly. Using the descriptions of information sources discussed in the previous section, there was no way to infer that subsequent queries to other information sources would be *redundant.* Etzioni et al. (Etzioni *et al.* 1994) describe a method for specifying that a source has *locally complete* knowledge about some aspect of the domain. Such knowledge can be used to exclude redundant information sources. For example, if an information source provides *all* the telephone numbers of the 212 area code, and a certain 212 number was not found there, then there is no need to query any other information source that may have 212 numbers.

To express information of this form, we allow local closed world (LCW) statements of the following form in the **IM** knowledge base:

$$E(\bar{X}) \wedge C(\bar{X}) \Rightarrow R_1(\bar{X}_1) \wedge \ldots \wedge R_n(\bar{X}_n)$$

where E is a domain model relation, $C(\bar{X})$ is a constraint formula and the R_i's are information source relations. This statement expresses the fact that the conjunction of the R_i's gives *complete* information about the facts of E that satisfy C. In particular, if $n = 1$, then the source relation R_1 alone contains the complete information.

These statements are used by the query processor to prune information sources as follows. Recall that the first step in answering the query is to determine, for each domain model relation, which portion of it is relevant (Definition 1). Suppose we have determined that, for the relation E_i, the relevant facts are those that satisfy the constraint formula C_i. The next step is to find a set of information sources that can provide these facts. We first consider the LCW statements to find the maximal subset C_i' of C_i that can be obtained *completely.* Then we find a minimal set of sources that provides the tuples satisfying C_i'. We then use the query-tree to find sources that provide tuples satisfying $C_i - C_i'$, as described in the previous section.

Example 4: Suppose our query involves phone numbers from the NYC area (i.e., area codes 718 and 212). We have four information sources available: $S1$, which provides *all* the numbers in the 718 area code, $S2$, which provides *all* the residential numbers in the 718 area code, $S3$, which provides *all* the business numbers in the 718 area code, and $S4$, which provides numbers in the 212 and 718 area codes (but does not have complete information).

Our query processor will first derive that it has sources to compute *all* the 718 area code, and will determine that $S1, S2$ and $S3$ are sufficient (and therefore $S4$ is redundant). It then determines that $\{S1\}$ alone provides all the numbers in the 718 area code (and therefore $S2$ and $S3$ are redundant). Finally, it will try to find sources that have numbers in the 212 area code, and find $S4$. When it queries $S4$, it will ask only for 212 numbers, because the 718 numbers that $S4$ has are redundant with respect to $S1$. □

A natural question that arises in the presence of completeness (or LCW) statements (considered in (Etzioni *et al.* 1994)) is whether the answer to the query represents complete knowledge about the world, i.e., whether we can infer, from the completeness of the information sources, completeness of the query. For instance, in our example, we do not have complete information about numbers in the NYC area, even though we have complete information about the 718 area code. Levy (Levy 1994) shows that the problem of inferring

completeness for queries is closely related to the problem of determining independence of queries from updates (Levy and Sagiv 1993b). One consequence of this close connection is the following:

Theorem 2: *Suppose that for any rule in the knowledge base, the antecedent contains at most one occurrence of every domain relation. Then, the answer to a query is complete if and only if we have complete information about every portion C_i of a domain relation E_i that was deemed relevant.* □

In contrast, (Etzioni *et al.* 1994) showed that the *if* direction holds in general, without the restriction on the number of predicate occurrences. An important corollary of the above theorem is that, under the conditions of the theorem, the query processor of **IM** finds the *minimal* number of source relations needed to answer the query. This means that, although other source relations may be relevant, there is no subset of the chosen source relations that would produce the same answer to the query (i.e., every proper subset may result in missing some answers). In (Levy 1994), a sound and complete inference procedure for determining completeness of queries is described for the general case.

It should be emphasized that our treatment of local closed world information generalizes that of (Etzioni *et al.* 1994) by considering the semantics of expressive constraint formulas in the rules and completeness statements. In contrast, (Etzioni *et al.* 1994) reasons with conjunctions of constraints of the form $X = c$, where X is a variable and c is a constant. Finally, the close connection with the problem of independence of queries from updates sheds light on the complexity of the problem of inferring completeness (see (Levy 1994) for details).

Interacting with IM

The **IM** system implements a WWW client with a multimedia hypertext interface (similar to Mosaic) coupled with a knowledge base for organizing and querying Internet information sources. In a typical **IM** session, the user interacts with **IM** via three window panes: a hypertext browser, a knowledge base browser, and a command/query interaction pane. The bulk of this paper dealt with queries in **IM**. We now briefly describe the user interactions with the hypertext and knowledge base browsers in **IM**. A close coupling between the knowledge base and the hypertext browser permits users to move seamlessly between hypertext navigation, structured browsing of the information space and organization of new information sources. Specifically, the coupling enables the user to easily transfer information from the browser to the knowledge base, and allows information on the Internet to be used to answer queries posed to the knowledge base.

The **IM** interface supports refinement and extension of the knowledge base as users discover new information and their view of the information space evolves. Information sources are added to the knowledge base by simply picking them up from the hypertext browser and throwing them into the knowledge base browser at an appropriate place in the topic hierarchy. At this point, the **IM** system fills in most of the "surface" characteristics of the information source automatically (e.g., physical characteristics such as modification time, type, etc). For more complex characteristics, a representation language allows one to express more sophisticated relationships, as described previously. A given **IM** knowledge base therefore represents a unique view of the information space, tailored to the interests of an individual or group. The knowledge base is persistent across **IM** sessions.

The knowledge base serves as both a repository for descriptions of information sources, and as a medium for browsing and querying them. Many retrieval operations may be expressed by browsing the knowledge base and using simple gestures on knowledge base objects to find information sources of interest (e.g. "find documents with at least one topic under OODB"). A result of this retrieval operation is that the system can position the user on sites on the Internet relevant to the query for subsequent browsing of *unstructured* information sources. For retrieval actions that cannot be expressed as browsing operations, a query language allows one to express more complex queries, as described previously.

Related Work

The **IM** approach to retrieving and organizing information from disparate sources is related to the approach taken by systems such as SIMS (Arens *et al.* 1994) and Carnot (Collet *et al.* 1991), to work done in software agents (e.g., (Cohen *et al.* 1994)), as well as to work done in multidatabase systems.

The work most closely related to ours is the SIMS project for integrating multiple information sources (Arens *et al.* 1994). The representation of the domain in their system is based on the description logic system LOOM (MacGregor 1987). Answering a query in SIMS proceeds in two steps: finding the relevant sources and accessing them. In SIMS, both components of the problem are posed as search problems, whereas, in our approach, the query processor tries to use the representations of the sources as much as possible before accessing any external information sources. Unlike the approach of SIMS, our algorithms are guaranteed to give only relevant sources. Our domain representation language is more expressive than that used in SIMS. For example, SIMS does not allow arbitrary n-ary relations, and therefore, it has to map each external relation to a concept in LOOM. This can be done only when the relation has a primary key. Although one can always conceptually add another such attribute to a relation, modeling a relation in such a way is unnatural. Furthermore, it limits the kinds of

relationships that can be expressed between sites (in particular, it is not possible to express the fact that one relation is a join of two others).

Both multidatabase systems and our approach have as their goal the ability to access multiple autonomous databases through querying. However, there are many differences between the multidatabase system architecture and ours. The approach taken by multidatabase systems is to make the multiple autonomous databases usable without a conceptually unified domain model. Their main reason for this approach is to preserve source autonomy, while supporting global updates. Our architecture only seeks to provide global querying, and mandates that all updates be performed at each source locally. Furthermore, it should be noted that in our approach, the domain model is purely conceptual and does not require each source to support a global schema. In querying a multidatabase system, the user has to be explicitly aware of the existence and the contents of the conceptual schemas in each of the autonomous databases. While the multidatabase approach to querying multiple source relations is feasible when the number of source relations is small, it can be quite cumbersome in the presence of a large number of information sources.

Another project related to ours is the Carnot project (Collet *et al.* 1991), which has similar objectives to multidatabase systems in providing resource integration, global querying and global updates. The representation of the domain in their system is based on the Cyc knowledge base, and articulation axioms establish equivalences between components of the source schemas and components of the domain schema, to enable the task of schema integration. However, articulation axioms do not allow for the specification of the semantic contents of source relations, and hence there is no notion of optimizing queries in Carnot.

Cohen et al. (Cohen *et al.* 1994) also use the language of Horn rules to represent the relationships between the contents of external sources and the domain model. However, they did not consider the issue of query optimization.

Conclusion

The functionality of the current **IM** prototype is complete enough to demonstrate the utility of using knowledge representation to assist with location and organization of information distributed throughout the Internet; we already have a useful tool. The second main contribution of our work is a general treatment of the issue of query optimization in distributed heterogeneous environments.

References

Arens, Yigal; Chee, Chin Y.; Hsu, Chunnan; and Knoblock, Craig A. 1994. Retrieving and integrating data from multiple information sources. *International Journal on Intelligent and Cooperative Information Systems.*

Brachman, R. J.; Borgida, A.; McGuinness, D. L.; Patel-Schneider, P. F.; and Resnick, L. A. 1991. Living with CLASSIC: When and how to use a KL-ONE-like language. In Sowa, John, editor 1991, *Principles of Semantic Networks.* Morgan Kaufmann, San Mateo, CA. 401–456.

Cohen, Philip R.; Cheyer, Adam; Wang, Michelle; and Baeg, Soon Cheol 1994. An open agent architecture. In *Working Notes of the AAAI Spring Symposium on Software Agents.*

Collet, Christine; Huhns, Michael N.; and Shen, Wei-Min 1991. Resource integration using a large knowledge base in Carnot. *IEEE Computer* 55–62.

Etzioni, Oren; Golden, Keith; and Weld, Daniel 1994. Tractable closed world reasoning with updates. In *Proceedings of KR-94.*

Levy, Alon Y. and Rousset, Marie-Christine 1995. CARIN: A representation language integrating horn rules and description logics. Technical report, AT&T Bell Laboratories.

Levy, Alon Y. and Sagiv, Yehoshua 1992. Constraints and redundancy in Datalog. In *Proceedings of the Eleventh ACM Symposium on Principles of Database Systems*, San Diego, CA.

Levy, Alon Y. and Sagiv, Yehoshua 1993a. Exploiting irrelevance reasoning to guide problem solving. In *Proceedings of IJCAI*, Chambery, France.

Levy, Alon Y. and Sagiv, Yehoshua 1993b. Queries independent of updates. In *Proceedings of the International Conference on Very Large Databases.*

Levy, Alon Y. and Sagiv, Yehoshua 1995. Semantic query optimization in Datalog programs. In *Proceedings of the ACM Symposium on Principles of Database Systems.*

Levy, Alon Y.; Mumick, Inderpal Singh; Sagiv, Yehoshua; and Shmueli, Oded 1993. Equivalence, query-reachability and satisfiability in Datalog extensions. In *Proceedings of the ACM Symposium on Principles of Database Systems*, Washington, D.C.

Levy, Alon Y.; Mendelzon, Alberto O.; Sagiv, Yehoshua; and Srivastava, Divesh 1995. Answering queries using views. In *Proceedings of the ACM Symposium on Principles of Database Systems.*

Levy, Alon Y. 1994. Relating LCW to the detection of queries independent of updates. In Preparation.

MacGregor, R. M. 1987. A deductive pattern matcher. In *Proceedings of the Sixth National Conference on Artificial Intelligence.* 403–408.