Report on the 5th International Workshop on Knowledge Representation Meets Databases (KRDB'98)

Alex Borgida
Rutgers University
USA
borgida@cs.rutgers.edu

Vinay K. Chaudhri
SRI International
USA
chaudhri@ai.sri.com

Martin Staudt
Swiss Life
Switzerland
martin.staudt@swisslife.ch

Introduction

Both databases and knowledge bases are used to represent the relevant parts of an application domain, and to allow convenient access to the stored information. Research in knowledge representation (KR) originally concentrated on expressive formalisms with sophisticated reasoning services, usually under the assumption that the size of the knowledge base (KB) was relatively small. In contrast, database (DB) research was concerned with efficiently storing, retrieving, and sharing large amounts of simple data, but the languages for describing schema information were rather simple, and reasoning about the schema played only a minor role.

This distinction between the requirements and problems in KR and DB is vanishing rapidly. On the one hand, a modern KR system (KRS) must be able to handle large data sets if it is to be employed in realistic applications. This means that techniques developed in the DB area can and should be employed. On the other hand, the information stored in DBs is becoming more complex and comes from heterogeneous sources, thus requiring more intelligent construction and retrieval techniques, especially the use of meta-data, which is really knowledge about data.

The series of KRDB workshops, started in 1994 and intended to foster synergy between the DB and KR fields continued this year on May 31st in Seattle, Washington, USA, with the 5th KRDB workshop, this time with special emphasis on *Innovative Application Programming and Query Interfaces.*

The workshop format was unlike any of the previous KRDB workshops. The workshop consisted of two contributed paper presentations (on Interfaces and Knowledge-based Tools), two invited talks (on Application Programming Interfaces and Reasoning for Query Answering), and two panel sessions (on Evaluating Query Interfaces and Semistructured Data). In addition, there was an introductory session, where participants could "make their point" and shortly relate themselves and their work to the general KRDB topics. The contributed paper presentations were selected from the submitted papers that were best-recommended by the program committee. The authors of other highest-ranked accepted papers were asked to serve as respondents to invited talks or panel participants. The complete list of accepted papers appears in the bibliography of this report ([1] to [19]), and the proceedings are available electronically.[1] We summarize below the principal points of the five sessions.

Interfaces and Knowledge-based Tools

Creating interfaces to heterogeneous information systems is difficult because of syntactic and semantic *heterogeneity* among the component databases, the *dynamic nature of the data sources*, and the problem of *locating the relevant* information sources.

A common approach to building an interface for a heterogeneous system is to define a unifying ontology or "super view". A promising solution to accomplish this is to use *software tools* supported by knowledge bases — an approach that has been used on other information systems (IS) related tasks, such as building and maintaining data-intensive information systems. The papers by Karunaratna et al. [12] and Peterson et al. [2] do just this, although in surprisingly different and novel ways.

Karunaratna et al. propose building an environment for managing loosely coupled federated DBs that uses a knowledge base containing information about the participating DBs, their schemas and semantics, existing views made by others, and so forth. This KB supports tools for browsing, locating relevant data sources, and helping to integrate them. The KB stores, using a semantic network, information in four layers, corresponding to actual databases, their meta-data, universe of discourse (UofD) concepts, and global views generated so far. One particularly interesting tool uses heuristic information pulled from implementation properties of component databases (e.g., foreign keys) to set up

[1] See **http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-10/**. KRDB home page is **http://sunsite.informatik.rwth-aachen.de/Societies/KRDB/** and has links to information on past and future KRDB workshops.

automatically candidate semantic relationships (such as synonymy) between concepts (and meta-data objects), and to give these relationships different numeric weights representing estimated strengths. The human user can then alter these links and values in the KB. Other tools help locate and merge schema substructures, and maintain dependencies to alert users when one of their sources changes. Finally, they provide a GUI that can be used to extract meta-data from the affiliated DBs and to browse and edit the result.

Peterson et al. address the issue of heterogeneity in an entirely different and new manner: rather than struggling to integrate *a posteriori* the (implicit) ontologies of independently developed DBs, they propose to supply tools to derive the database schema from a single universal ontology — namely, the Cyc knowledge base developed over more than a decade — first at MCC and then at Cycorp [GL94]. In their solution, a database designer selects a subset of the Cyc KB that best models the UofD for the particular application, and then build the database schema and its interface using that subset. This approach would ensure that multiple DBs built in this way automatically articulate with each other on the overlapping areas, thereby avoiding the standard semantic mismatch problems. To support this new paradigm, the authors have implemented a tool that starts from a subset of Cyc KB, and "pulls" in the other relevant parts of the Cyc KB.

To be assured that the information contained in the DBs that are spawned remains compatible, it is necessary to capture in the DB not just the basic conceptual schema (classes, attributes) but also the full semantics (constraints, derivations) in the original domain theory, expressed in Cyc as axioms. However, the language used by Cyc (Cyc-L) is an extension of First Order Logic, which makes it unsuitable as a foundation for database applications because of the complexity of performing bulk updates and queries on large numbers of individuals and the absence of the Closed World Assumption. To resolve this, the paper offers a wide variety of techniques for *approximating* Cyc-L formulas using Horn rules, and then implementing these in the XSB deductive database.

The above systems illustrate two approaches to the building of knowledge-based interfaces to information systems: derive and maintain a knowledge base from the existing databases, or use an existing knowledge base to derive the information system. In either case, knowledge-based tools are central to the solution.

Application Programming Interfaces

Among many other commonalities, KRSs and DBMSs both tout availability of a *declarative* (as opposed to a procedural) interface as one of their greatest advances over the state of the art, say, 20 years ago. A declarative interface is most useful for human users, but may be an impediment to achieving the greatest possible efficiency in situations where the IS is being used as an embedded server to other applications. Some of the significant systems issues that arise here include the access to bulk collections of information being returned as a result of queries or being entered into the IS, as well as the treatment of exceptions, which indicate not only user errors resulting in an inconsistent IS state but also the inability of the system to perform operations (e.g., resource limits being exceeded or restriction on operations, such as null values in relation keys).

The invited presentation by Richard Fikes unveiled the Open Knowledge Base Connectivity (OKBC) proposal for accessing a large class of "frame-based" KRSs.[2] The goal of this proposal is to provide a uniform collection of function calls with which applications can be built without knowing the details of the specific KRS that is being used as a source of information.

A significant contribution of the OKBC proposal, and its precursor Generic Frame Protocol (GFP), is in identifying a set of *concepts* that underlies a significant collection of object-centered KRSs ranging from Cyc [GL94] to Loom [MAC91]. These concepts include notions such as object, individual, class/concept. OKBC also defines a declarative constraint language specified using facets (ternary relations).

A significant problem faced by the OKBC proposal is the wide variability among the various KRSs both in terms of the notions supported (e.g., treating slots as first-class citizens) and in terms of the inferences they perform (e.g., forward chaining). For the former, OKBC introduces a list of **behaviors** that are dimensions along which a system can place itself, and for the latter it identifies three modes of retrieval: directly/as-told facts, inheritance, and any other means of inference. (These reflect the importance of inheritance as a standard reasoning mechanism for frame representation systems.)

Pat Martin and other database discussants pointed out that the OKBC enterprise faces a much more heterogeneous world than Open Database Connectivity (ODBC) — the equivalent API interface to relational databases (RDBMSs): in the RDBMS case there is a single, well-understood data model, which is much simpler, so there is no need to achieve consensus on a set of basic notions. It was remarked that even in relational systems, the treatment of null values (one area where there is room for "inference") appears to be a problem for ODBC; discussants pointed out that most ODBC accesses are for quite simple queries. For subtle queries, no support is given by ODBC, and an application must communicate to a DBMS directly.

A second area of difficulty for relational APIs was raised by J. Freire [11], who pointed out the existence of a communication mismatch between the RDBMSs

[2] A detailed paper on this topic will appear as [CFFKR98].

and its user program. For example, answer relations are normally accessed tuple-at-a-time by using standard cursors, and this is far too inefficient. Notions like bulk access to tuples through arrays, and client-side caching of tuples are solutions that are not uniformly supported, but should be. OKBC addresses at least some of these issues through support for iterators that can be told to bring in more than one object from the knowledge base.

KRSs usually assume that there are relatively few individuals and, with a few exceptions, do not support query languages — only procedures for accessing local information. Martin et al. address these issues by providing an API to better support querying knowledge bases containing many individuals [13]. Unlike OKBC, Martin's API is intended for a specific data model, Telos.[3] Martin's API supports querying by content of a KB (conjunctive queries involving the (binary) relationships in the Telos model) as well as by navigation, and also supports manipulation of *large data sets*. In an interesting reflective twist on the theme of the workshop (KR and DB), Martin et al. applied their techniques to sizable knowledge bases that represent meta-data, including the World-Wide Web (WWW).

Reasoning for Query Answering

An important contribution of KR research toward improved DB systems consists in providing reasoning mechanisms that enable efficient query answering. Data warehouse (DW) is an interesting application area where data is collected from different sources into one central data store and made available for decision support applications. On the one hand, a DW itself is a materialized and integrated view on a number of heterogeneous data sources, for example, productional online transaction processing (OLTP) systems. On the other hand, a DW can be seen as a collection of (overlapping) views, sometimes called data marts, serving different analysis tasks and targeted at different user groups.

A key inference related to materialized views is computation of subsumption relationships between views purely based on their definition. Over past five years, a number of KR and DB researchers have used the experiences gained in devising subsumption algorithms for KR languages to extend the query containment techniques developed by DB researchers and applied the results to certain other problems that occur, for example, in the DW context.

The invited talk by Alon Levy was dedicated to the DW design problem, that is, selecting a palette of views to be materialized in the DW that guarantees an efficient answering of decision support queries.

[3] We note that making an OKBC interface for Telos presents some serious challenges because Telos [MBJK90] integrates time thoroughly in the model, and reifies all relationships (including class membership).

Similar to designing an appropriate representation of an application domain for building a knowledge-based system, designing an appropriate database (or data warehouse) schema is a key to obtaining efficient performance of a database application. One of the questions addressed by the AI community almost from its inception is that of problem reformulation. That is, how can a system automatically reformulate a representation of the domain in order to yield better performance for some specific tasks. Reformulations have been considered in a variety of problem-solving settings, including automatic programming, constraint satisfaction, design, diagnosis, machine learning, planning, qualitative reasoning, scheduling and theorem proving.

The design of data warehouses poses another instance of the reformulation problem. Levy described the problems of reformulation and of DW design from a unified perspective, exemplifying commonalities and differences between the problems and their treatments in the literature. Several areas of future research were suggested: (1) the use of approximate views in data warehouses, (2) the use of additional domain knowledge in the design of data warehouses, and (3) the use of algorithms for rewriting queries using views for translations between ontologies and for problem reformulation.

As a respondent to Levy's talk, Critchlow emphasized the importance of query reformulation and rewriting [8]. He explained the DataFoundry project at Lawrence Livermore National Laboratory that approaches the problem of handling frequently changing source schemas for data warehouses in scientific domains like genetics. Query reformulation ensures that the query is propagated to the correct data sources. DataFoundry includes the wrappers and mediators that act as intermediaries between the query processing layer and the individual data sources. While this infrastructure is normally assumed to be reliable and static, this is not the case in dynamic scientific environments. The DataFoundry project makes an extensive use of an ontology infrastructure to reduce the impact of change on the warehouse. An Ontolingua-based ontology serves for representing four types of knowledge: abstract domain-specific concepts, database descriptions, mappings between the database and the abstractions, and transformations between different data formats. Using this knowledge, it becomes possible to automatically generate mediators and therefore to significantly reduce the effort for maintaining the warehouse.

In his response to Levy, Stefan Decker pointed out that even problem formulation is difficult, for example, in knowledge acquisition tasks. He proposed to use more domain-specific KR and DB languages supporting conceptual primitives that make modeling tasks easier. For implementation, Decker argues in favor of compiling these primitives to lower-level languages that can be executed by standard techniques from logic programming and deductive databases. As an example. Decker sketched the re-engineering of the deductive and object-oriented data model F-Logic and the temporal representation language Chronolog (based on Linear Temporal Logic). The resulting language can be implemented by an inference engine that is able to handle Horn logic with negation.

Evaluating Query Interfaces

The panel on Evaluating Query Interfaces was led by Umesh Dayal. The participants were Ugur Cetintemel (University of Maryland at College Park), Vinay K. Chaudhri (SRI International), and Ray Liuzzi (Air Force Research Laboratory).

Dayal introduced innovative query interfaces as unconventional DBMS applications that do not fit the OLTP model. Quite often such applications are described using adjectives (as in the current paragraph) and quantitative metrics to characterize them and measure their performance are lacking. The panelists were asked to define their vision of innovative query interfaces and propose techniques for evaluating the performance of such interfaces.

Centintemel's vision of an innovative query interface is that of a query processing system that takes into account the heterogeneity, availability, and usage cost of the source information, returns answers that may not be precise. and is sensitive to the data quality. Such requirements are typical while building a query processing system for the WWW. For such a system. existing performance evaluation metrics and methodologies that focus only on efficiency are insufficient because the quality of answers must also be taken into consideration.

Cetintemel talked about his recent project on *Unsafe Query Optimizations for IR*, in which query evaluation strategy is dynamically changed based on the buffer contents. For evaluating such a system, a new metric called *non-interpolated average precision*, which combines precision and recall into a single number, was defined. Even though his work represents a solid example of evaluating an innovative query interface, significant hurdles were faced in finding realistic workloads.

Chaudhri's vision of an innovative query interface is that of a system that is able to use ontological information, can return intentional answers, and has an ability to explain answers. When asked a question about companies located in California. an innovative query interface may return those companies which are known to be in Palo Alto. The system accomplishes this by using the information that Palo Alto is in California, and if a company is located in the subregion of a region, it is also located in that region. Such information may reside in a knowledge dictionary that could be part of a DBMS in the same way as a data dictionary.

Chaudhri defined several metrics for the performance of such a system. For example, *conceptual magnification* measures the additional number of queries answerable by having a knowledge dictionary over a system that does not have a knowledge dictionary. Designing a good knowledge dictionary and integrating it into a DBMS poses significant research challenges. He described the use of HPKB Upper Ontology, a knowledge dictionary, which is being used in a query processing system that he is building.

Liuzzi's presentation focused primarily on an Internet-based resource for *Evaluating Intelligent Systems* (EIS) that has been funded by Air Force Research Laboratory. The EIS home page, available at `http://eksl-www.cs.umass.edu:80/eis/`, provides extensive information to those interested in following a rigorous empirical methodology for evaluating their systems. For example, the EIS pages include a Field Guide, which provides information on designing experiments and analyzing the data they produce. For each of several basic experiment types, the Field Guide describes techniques for data preparation, data exploration, hypothesis testing, and modeling. The description of each technique includes details about its application, as well as warnings about potential pitfalls, and suggested follow-up procedures.

In summary, two themes emerged from this panel. First, database query processing will evolve to produce better-quality answers, for example, sometimes the answers may not be precise and at other times they may be obtained by using domain-specific knowledge already encoded in the DBMS. Second, we will need to invent rigorous metrics to help us evaluate innovative query interfaces. Such metrics may be obtained by combining the metrics used in DBMS system performance with metrics in information retrieval.

Semistructured Data

The panel on semistructured data was chaired by Daniela Florescu (INRIA, France), and the panelists included Phil Bernstein (Microsoft, USA), Maurizio Lenzerini (University of Rome, Italy), Len Seligman, (MITRE, USA). and Dan Suciu (AT&T Labs, USA),

Semistructured data is emerging as an active area of research. The purpose of the panel was to evaluate the field and identify connections to knowledge representation. Dana Florescu summarized the main characteristics of semistructured data, centering around the appropriate notion of "*schema*" and "*object structure*":

1. The schema is not given in advance, and is often implicit in the data itself.
2. The schema is descriptive, rather than prescriptive, so that data can deviate from it.
3. The schema is partial, so additional information can be easily attached at any place.

4. The schema evolves rapidly – this is particularly natural when it is implicit in the data, as pointed out in the first item.
5. The schema may be large compared to the size of the data.
6. Objects and attributes are not strongly typed.
7. Objects in the same collection may have different representations.

The application areas that handle semistructured data include biology, data integration, digital libraries, multimedia systems, and WWW sites. According to Suciu (organizer of last year's SIGMOD workshop on semistructured data [SUCIU97]), the vast majority of data in the world's computers, whether scientific data, WWW pages or documents, is not stored in databases managed by DBMSs. Suciu claimed that the advent of the XML standard for document markup is a particularly significant step toward making most of the data on the WWW world look like semistructured data.

Seligman [16] took the position that the decision process for using some "management system" to store semistructured data, as opposed to flat ASCII files, starts with questions such as the kinds of *queries* desired (e.g., how complex do they need to be?), and whether additional DB-like operations are needed (e.g., constraints, views, update control, transactions). In addition to detailing the above points, Seligman considered a scheme for evaluating the benefits of moving to a semistructured data management system. He cited an online university technical report library as a case where there is insufficient gain to move away from a simple text-based system, pointing to a variety of costs in making this move, including the cost of writing "wrappers" that create an interface conforming to the new data model, and then the cost of maintaining the wrappers. Seligman pointed out that in some cases one may even consider moving to a regular DBMS, to gain the full benefits of existing technology. [4]

The "standard" models for semistructured data that seem to have taken hold (among the best known are LOREL and BDFS) involve viewing the information as directed, labeled graphs. Many of the properties of semistructured data (especially items 2, 3, and 6) are characteristic of knowledge representation and reasoning schemes. Moreover, *semantic networks* in KR are representation schemes based on labeled graphs. Lenzerini and his group have pursued intensively the application of *description logics* – which are formalized descendants of semantic networks – to the problem of semistructured data management. In particular, they have shown [LENZ98] that one of their expressive languages can add the ability to express both local and non local constraints in the BDFS model, while maintaining decidable reasoning about the schema. Moreover, they show an example involving a loop in the graph whose proper interpretation requires *greatest fixed-point* semantics for predicates, which is currently offered only by their reasoner.

Finally, Bernstein made a thought-provoking presentation claiming that systems for the management of semistructured data already exist — they are called **"repositories"** — and in fact service a *big* market. A repository is a place for sharable and reusable metadata (often about software applications, but also about data and documents), which is used by tools such as browsers and scripting languages. Repositories are used in areas such as application development environments, databases (data warehouses, information resource management), systems (site configuration and distributed system management), and now WWW/document management. Repositories (see [BSHSZ97]) support an information model that is object-oriented and is *highly extensible* (through meta-meta-models, type interpreters), while supporting semantically rich relationships, and version and configuration management, as well as database-like amenities such as views, SQL queries, security, and even transactions. These features are available today (`http://www.microsoft.com/repository`) and can deal with many aspects of the semistructured data mentioned above, while providing additional benefits, including efficient implementations. Bernstein pointed out that a repository's type systems are like persistent semantic networks, and they could use additional help with reasoning tasks and more declarative behavior. For example, declarative query or constraint languages native to the data model appear to be lacking. This is an opportunity for interesting KR additions, but with the caveat that performance is critical.

In summary, semistructured data, with its semantic network-like features, provides a compelling area of research at the intersection of DB and KR, and where much could be gained by learning from existing repository products and prototypes.

Summary

The following are some themes that were raised more than once in the "live" sessions of the workshop (the papers for the whole workshop are readily available to the reader electronically), and as such might be interesting sign-posts for future KR-DB directions:

Approximate answers were identified as useful ideas in innovative query interfaces and as possible improvements in the design of data warehouses.

Use of domain knowledge continues to be a key technique for DB integration, design, query interface, and good warehouse design.

[4] One of the reporters cannot resist the opportunity to point out that *if there are relatively few deviations* from the schema, existing research on accommodating exceptions in databases can make this a palatable alternative [BORG85].

Declarative languages of various kinds (whether for stating constraints on semi-structured data , querying knowledge bases , characterizing reasoners) continue to be of interest, at the same time as procedural. more navigational, API interfaces are desired.

Characterization of performance for databases with semantic content was a central theme in the panel on innovative query interfaces, and was reinforced by remarks that KR solution for semistructured data would greatly benefit by serious performance claims.

Acknowledgments

The workshop was co-sponsored by ACM SIGMOD, Air Force Research Laboratory (Rome Laboratory), USA, and Swiss Life, Switzerland. The coordinators were supported at their respective institutions by Rome Laboratory contract titled *Reusable Tools for Knowledge Base and Ontology Development*, Contract Number F30602-96-C-0332 (Chaudhri), and NSF Grant IIS-9619979 (Borgida).

We thank the members of the program committee: Craig Anken, Franz Baader, Umesh Dayal, Adam Farquhar, Manfred Jeusfeld, Alon Levy, Alberto Mendelzon, Werner Nutt, and Timos Sellis, who served us so generously. We are also grateful to the panelists who agreed, often on short notice, to participate in the workshop, and who made it a lively event.

Finally, we thank Loni Syltebo for her support in local arrangements, Pamela Drew, the SIGMOD General Chair, and Alberto Mendelzon, the PODS Chair, for helping collocate KRDB'98 with SIGMOD/PODS'98.

References

[1] "Sophia: Providing Basic Knowledge Services with a Common DBMS", N.F. Abernethy, R.B. Altman.

[2] "Knowledge Bus: Generating Application-focused Databases from Large Ontologies", B. Peterson, W.A. Andersen, J. Engel.

[3] "Intelligent Caching for Information Mediators: A KR Based Approach", N. Ashish, C.A. Knoblock, C. Shahabi.

[4] "Hypothetical Database Temporal Reasoning with History Encoding", M. Arenas, L. Bertossi.

[5] "On Extending a Semantic Data Model with Some Aspects", M. Boufaida, Z. Bouriche-Boufaida.

[6] "The Interactive Databases Approach to the User Interface Modeling", E. Bugaenko.

[7] "Bottom-up Integration of Ontologies in a Database Context", J.-C.R. Pazzaglia, S.M. Embury.

[8] "Automatic Generation of a Warehouse Mediators Using an Ontology Engine", T. Critchlow, M. Ganesh, R. Musick.

[9] "On Domain-Specific Declarative Knowledge Representation and Database Languages", S. Decker.

[10] "Reasoning About Web-Site Structure", M. Fernandez, D. Florescu, A. Levy, D. Suciu.

[11] "Practical Problems in Coupling Deductive Engines with Relational Databases", J. Freire.

[12] "Organizing Knowledge of a Federated Database System to Support Multiple View Generation", D.D. Karunaratna, W.A. Gray, N.J. Fiddian.

[13] "A Metadata Repository API", P. Martin, W. Powley, P. Zion.

[14] "Semantic Query Optimization and Rule Graphs", J. Robinson, B. Lowden.

[15] "Deductive Data Mining: Uncertainty Measures for Banding the Search Space", L. Russell.

[16] "Databases for Semistructured Data: How Useful Are They?", L. Seligman, K. Smith, I. Mani, B. Gates.

[17] "Role-based Semantics for Conceptual-Level Queries", D. Silberberg, R.D. Semmel.

[18] "Knowledge-Intensive Query Processing", B. Starr, V. Chaudhri, A. Farquhar, R. Waldinger

[19] "Evaluating Answer Quality/Efficiency Tradeoffs", U. Cetintemel, B.T. Jonsson, M.J. Franklin, C.L. Giles, D. Srivastava

[GL94] R.V. Guha, D. Lenat, "Enabling Agents to work Together", *Communications of the ACM 37(7)*, 1994, pp. 126–142.

[CFFKR98] Vinay K. Chaudhri, Adam Farquhar, Richard Fikes, Peter D. Karp, James P. Rice, "OKBC: A Programmatic Foundation for Knowledge Base Interoperability", *Proc. AAAI'98 Conference*, Madison, WI, July 1998.

[SUCIU97] Special Section on Management of Semistructured Data, D. Suciu editor, *SIGMOD Record 26(4)*, December 1997, pp. 3–44.

[LENZ98] Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, "What can Knowledge Representation do for Semistructured Data?", *Proc. AAAI'98 Conference*, Madison, WI, July 1998.

[BSHSZ97] Philip A. Bernstein, Brian Harry, Paul Sanders, David Shutt, Jason Zander, "The Microsoft Repository", *Proc. VLDB-97*, Athens, Greece, pp. 3-12.

[BORG85] Borgida Alex. "Language Features for Flexible Handling of Exceptions in Information Systems", *ACM Transactions on Database Systems*, 10(4):565–603, 1985.

[MAC91] R. MacGregor, "The Evolving Technology of Classification-based Knowledge Representation Systems", In J. Sowa, editor, *Principles of Semantic Networks*, pp. 385–400. Los Altos, CA, 1991.

[MBJK90] John Mylopoulos, Alex Borgida, Mathias Jarke, Manolis Koubarakis. "Telos: A Language for Representing Knowledge About Information Systems", *ACM Transactions on Information Systems*, 8(4):325–362, October 1990.