Some research trends in KR&DB (position paper)

Paolo Bresciani
IRST
via Sommarive, 18
I-38050 Trento Povo, Italy
brescian@irst.itc.it

Abstract. In recent years, applications based on database tools and those based on Description Logic systems are progressively converging towards integrated technologies that try to overcome the limits of each single discipline. Therefore, a great call for an integrated view of Description Logics and Database is emerging.

In this position paper, some research topic in the database area for which Description Logics seem very promising, due to their richer formalism and inference mechanisms, are mentioned. In particular we list some of them in the field of database conceptual modeling, and in the field of intelligent information access.

Moreover, our past and present efforts in devising some technologies for integrating the access to knowledge bases and databases are recalled, together with a couple of application examples realized, or under development, at our Institute. Also some other relevant application fields are mentioned.

1 Introduction

Traditional software applications – for example, bank accounts management software, airline flights booking systems, etc. – that need the storage and management of large amount of data, rely on Database (DB) techniques and systems, while Artificial Intelligence (AI) applications – like, for example, expert systems applications, natural language dialogue systems, decision support systems – that need the managing and reasoning about more structured data, rely on the Knowledge Representation (KR) languages and systems.

In fact, the main difference between knowledge representation management systems (KRMS), and database management systems (DBMS) is that the latter are oriented to the efficient management of large amounts of data while the former seek to give a more structured representation of the universe of discourse in which data are placed. In fact, one of the most important features of KR is the ability of providing the *classification* of complex object descriptions, in order to organize them into *taxonomies*. This hierarchical organization of object descriptions largely contributes to formulate more accurate models of real domains, and, therefore, is one of the building blocks of the sophisticated reasoning tasks that characterizes AI applications.

Databases, instead, are suited to manage data efficiently, with little concern about their dimension, but the formalism for organizing them in a structured way is quite absent, as well as the capability to infer new information from that already existing.

On the other hand, several similarities can be found between knowledge representation and database modeling [Borgida,1991]. The basic one is that both databases and knowledge bases seek to capture in (simplified) symbolic models the relevant knowledge about portions of the world and their behaviors. In this way, this knowledge can be reproduced – both statically and dynamically – into computer applications, in order to give some benefits to the users, as, for example, the ability of examining the informative content of the real world – even considering possible evolutions – without involving real access or modification of it, but only acting on the model.

In fact, in recent years, traditional software applications are moving progressively towards a wider and wider use of AI techniques and KR tools that can help to add more accurate model descriptions and more flexibility, for example, to their Computer-Human Interfaces. In particular, a greater and greater call to include "intelligent" aspects in Information Systems (IS) is being developed. Primarily, it must be considered, here, the task of enhancing the ability of accessing IS resources both in terms of a wider and easier use – even by non-expert users – and in terms of a deeper and more accurate and effective usability of the IS. This involves the realization of more easy-to-use interfaces – based on advanced multi-modality and/or natural language interaction – and intelligent tools that integrate and coordinate the access to distributed and heterogeneous sources of information, and aggregate, interpret, and adequately present the results of such accesses.

In this context, the combination of effective Computer-Human Interfaces and intelligent information retrieval is the base for the realization of Intelligent Information Systems (IIS) based on iterative information bases navigation. AI, and more specifically KR, play a primary role in this process, providing important tools – in both the mentioned aspects – for building Intelligent Information Systems [Brodie,1988].

Of course, also from the point of view of IS design, and in particular in conceptual data modeling, KR tools and formalisms are particularly appealing, due to the similarities between these two disciplines [Borgida,1991]. For example frame description languages (FDL) can be used to capture the semantic of standard conceptual models, like, e.g., entity-relationship [Bergamaschi and Sartori,1992].

Indeed, the importance of KR has been regarded as fundamental for the construction of good IIS for more than ten years (see, e.g., [Tou *et al.*,1982]). But only recently basic progresses have been made by establishing the theoretical foundations of a Description Logic (DL) based approach to databases

[Buchheit *et al.*,1994]. This turns out to be of relevent importance because DL are likely to offer promising formalisms for solving several problems concerning data modeling and access, like, for example, the definition of rich schema languages, the schema integration, the inter-view relationship handling, and the intelligent query management. In fact, description logics can be considered as an unifying formalism, since they allow the logical reconstruction and the extension of representational tools such as *frames*, object-oriented data models, semantic data models, semantic networks. Thus, the ideal IIS should incorporate the characteristics of both DL and DB.

On the other hand, some prototypical applications based on AI, and in particular on KR, are becoming more and more realistic, and begin to be considered "true" applications. Often, scalability is one of the major problem to be faced in the hard process of transformation from KR-based prototypes to real applications, because typical KRMS have serious difficulty in managing very large knowledge bases. Instead, DBMS can manage large amounts of data efficiently, but, as mentioned, they lack a good formalism for classifying and reasoning about them. Once again, it is clear that in the most ambitious AI-based applications, where KR and DB alone cannot succeed, an integrated use of both of them can reach the goal. Thus, KR-based applications and, more generally, AI-based applications can be widely enhanced by AI/DB interfaces [Pastor *et al.*,1992; McKay *et al.*,1990].

Concluding, it is not surprising that a great reciprocal interest is rising in both DB and KR areas about all the possible relations and cross-fertilizations of KR and DB (see, e.g., [Baader *et al.*,1994; 1995]).

2 Achieved Results

As Knowledge Bases (KB) developers, we meet the practical problem of integrating KB and DB since 1992.

First, we faced the problem of using external databases to extend knowledge bases when we were developing a large natural language dialogue system prototype [Bresciani,1992], domain and linguistic model of which were represented using LOOM [MacGregor,1991] knowledge bases.

In such kind of applications it is very important that the database can be queried from the Description Logic Management System (DLMS) in a way completely transparent to the user or the Natural Language (NL) application – with respect to how they query the KB – in order to allow to see a unique large Information Base (IB). Of course, this call for a semantically well founded linking between the knowledge base and the database. It can be obtained by *coupling* DLMS and DBMS [Borgida and Brachman,1993]: primitive concepts and relations in a KB are made to correspond respectively to unary and binary tables in a DB. Two possible way to couple DLMS and DBMS can be proposed: *loose coupling* (that requires a pre-loading of the data from the DB into the KB), and *tight coupling* (that implements a *on demand* access to the DB).

We implemented a system [Bresciani,1994; 1995b] based on tight coupling, where compound conjunctive queries involving unary and binary predicates can be done without wasting memory space in the DLMS in order to keep descriptions of DBMS data; moreover, answers are given on the basis of the current state of the KB and the DB, without needing periodical updating of the KB with new data from the DB.

The system is currently used in an enhanced version of the prototype described in [Bresciani,1992], that, using a KB extended by a DB, is capable of dealing with thousands of individuals.

Another IRST project in the field of NL interfaces, where such tools will be used (see [Bagnasco *et al.*,1996] for the description of a preliminary prototype) is being to be started at IRST. The scenario of the project include a user (a Public Administration clerk) that uses an advanced multi-modal interface – based on graphical aspects and natural language dialogue – to access several heterogeneous databases, in order to answer and give advice to citizens seeking for bureaucratic records concerning their social insurances. The clerk can use the interface without knowing the details of the structure of the many different databases involved. Also in this case the realization of the natural language interface implies extended use of knowledge bases, and, thus, of our integrated approach to KB and DB access.

3 Research Topics

In the present section some possible research topics will be listed, considering both applicative aspects and more technical/fundamental subjects.

3.1 Applications

In the following, some KR, DB, and IIS applications fields in which KR-DB integration can be profitably applied will be mentioned (see also [Bresciani,1995a]):

- *Large KB management.* In realistic applications, KB not only can be complex, but can also involve a large number of individuals, difficult to manage – when not impossible – with the existing DLMS. A large portion of data about them can be managed better by a DBMS.
- *Integrated KB and DB.* As noted in [Borgida and Brachman,1993], KB based on DL are often used in applications where they need access to large amounts of data stored in already existing databases. It is the case, for example, of the TAMIC system [Bagnasco *et al.*,1996].
- *Knowledge Acquisition.* As observed in [Bresciani,1994; 1992] the task of acquiring knowledge for a real knowledge based application often includes a great amount of raw data collecting. For this subtask a DB and its robust front-end can be more adequate than usual KB interfaces.
- *Data Archaeology.* In *data archaeology* the main task is the search and extraction of previously ignored knowledge from several and possibly large databases, by means of an interactive and iterative process of analysis and refinement [Brachman *et al.*,1992]. In this case, the use of intelligent front-ends based on DL can be helpful [Devanbu,1993]. The use of friendly and powerful interfaces is even more important when data are widely and easily available to several kinds of users, but scarcely inter-organized and spread over several sites as in the case of WWW [Kirk *et al.*,1994].
- *Distributed and Heterogeneous Databases.* In this case the problem is to plan the optimal access to many, distributed, and possibly heterogeneous databases [Hsu and Knoblock,1993; Levy *et al.*,1994]. DL is crucial on both the site descriptions meta-level, as noted in [Levy *et*

al.,1994], and on the data level, as noted in [Hsu and Knoblock,1993].

3.2 DL for Conceptual Modeling and Information Access

Other interesting application areas could be listed, but it seems here more appropriate to spend a little space to list some more general research topics in the database area for which DL seem very promising, due to their richer formalism and inference mechanisms.

DL for Conceptual Modeling

The use of DL based schema language can lead, for example, to the following advantages in conceptual modeling:

- *Ontological organization.* It is possible to capture important basic facets of data semantics, including the structure of complex entities and ontological dimensions such as time, space, and events.
- *Consistency checking.* The consistency of a schema and/or of some specific classes w.r.t. the whole schema can be checked.
- *Data entry.* The user is supported in the phase of populating the data base, according to the knowledge of the schema and satisfying the integrity constraints. The system could not only check the consistency of the data base itself, but also make some deductive inferences asserting new facts regarding the data.
- *Views organization.* DLMS can automatically organize views into a hierarchy – that is a non-trivial task, when there are many complex views – easing their retrieval and reuse.
- *Schema refinement.* The redundancy of a schema can be reduced, by using equivalent descriptions to minimize the number and/or the complexity of the definitions.
- *Inter-schema organization.* DL can be used to define common terminologies describing the schemata of different databases, easing the task of managing multi-database [Goñi *et al.*,1995].

DL for Information Access

The availability of large amounts of data that are heterogeneous in structure and origin is often a serious obstacle to their usability. An excess of information can be equivalent to an absence of information. It is therefore necessary to organize data into intelligible and easily accessible structures, and return answers at various levels of detail to support analytic and decisional activities.

DL can be used as a tool for representing queries. In this way the system is able to *reason* about them, allowing the following advantages:

- *Query validation.* Incoherent queries are detected, and the user is informed about their ill-formed request.
- *Query generalization.* When a query, even if it is consistent, return an empty answer, it is reasonable to generalize it until a non empty answer is obtained. The description lattice is the obvious structure to be used to support the user in the interactive and iterative task of searching for such generalizations.
- *Query refinement.* Like for query generalization, queries can be specified through a iterative refinement process supported by the description lattice for the queries. This process is useful for data exploration tasks, as needed, for example in data archaeology.
- *Query organization.* Data exploration may involve a great amount of queries. The queries hierarchy ease the retrieval of already submitted similar or equivalent queries, in order to support an efficient query caching mechanism [Goñi *et al.*,1995]. This is relevant if the queries need a substantial amount of time to be processed, as it happen for remote and/or multi-database systems.
- *Intensional query processing.* Users may explore and discover new generic facts without querying the whole database, but by giving an explicit meaning to the queries through classification. The system has the ability of answering a query with synthetic concepts representing the general characteristics of the data that satisfy it, as opposed to answering with long sequences of detailed data. Moreover, if the query is classified in a taxonomy of descriptions and queries already computed and indexing the answers, then it can be processed with respect to the indexed objects only, rather than with respect to the whole data base.
- *Query optimization.* Given schemata and a set of views and already processed and cached queries, a query can be optimized by computing an equivalent more efficient one. This can be obtained by using the cached results together with an estimation – computed using the semantic indexing – of the cardinality of the answer to the query and its sub-queries [Schmiedel,1994], as well as by substituting some single terms and complex descriptions with some more specific ones [Beneventano *et al.*,1993].

4 Conclusions

In the present short survey, some issues on the relationship between knowledge representation and databases have been presented. We shortly recalled our previous work in the field, specially concerning the integration of access to KB and DB, and its application to some applications in the field of NL interfaces. Some other application fields in which integration between KR and DB is profitable have been listed. From the point of view of research topics, we addressed some subjects, in the DB field, in which – in our opinion – DL formalisms have good chances to lead to important progresses.

Acknowledgment: our thanks must be addressed to Enrico Franconi, for the many cues given to the writing of this paper.

References

[Baader *et al.*, 1994] F. Baader, M. Buchheit, M. A. Jeusfeld, and W. Nutt, editors. *Working Notes of the KI'94 Workshop: KRDB'94 - Reasoning about Structured Objects: Knowledge Representation meets Databases*, number D-94-11 in DFKI Documents, Saarbrücken, Germany, September 1994. DFKI GmbH.

[Baader *et al.*, 1995] F. Baader, M. Buchheit, M. A. Jeusfeld, and W. Nutt, editors. *KRDB'95: Reasoning about structured objects: Knowledge representation meets databases*, number D-95-12 in DFKI-Research-Report, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany, 1995. DFKI GmbH.

[Bagnasco *et al.*, 1996] Clara Bagnasco, Paolo Bresciani, Bernardo Magnini, and Carlo Strapparava. Natural language interpretation for public administration database querying in the TAMIC demonstrator. Technical Report 9601-09, IRST, Povo, TN, Italy, January 1996. To appear in the Proceedings of the NLDB'96 workshop, Amsterdam, June 26-28, 1996.

[Beneventano *et al.*, 1993] Domenico Beneventano, Sonia Bergamaschi, Stefano Lodi, and Claudio Sartori. Using subsumption in semantic query optimization. In A. Napoli, editor, *IJCAI Workshop on Object-Based Representation Systems*, pages 704–709, Chambery, France, August 1993.

[Bergamaschi and Sartori, 1992] Sonia Bergamaschi and Claudio Sartori. On taxonomic reasoning in conceptual design. *ACM Transactions on Database Systems*, 17(3):385–422, September 1992.

[Borgida and Brachman, 1993] Alex Borgida and Ronald J. Brachman. Loading data into description reasoners. In *Proceeding of ACM SIGMOD '93*, 1993.

[Borgida, 1991] Alexander Borgida. Knowledge representation, semantic modeling: Similarities and differnces. In Hannu Kangassalo, editor, *Entity-Relationship Approach: The Core of Conceptual Modelling*, pages 1–25. North-Holland, 1991. Proceedings of the Ninth International Conference on the Entity-Relationship Approach.

[Brachman *et al.*, 1992] Ronald J. Brachman, Peter G. Selfridge, Loren G. Terveen, Boris Altman, Alex Borgida, Fern Halpern, Thomas Kirk, Alan Lazar, Deborah L. McGuinness, and Lori Alperin Resnick. Integrated support for data archaeology. In *ISMM International Conference on Information and Knowledge Management*, Baltimore, MA, November 1992.

[Bresciani, 1992] Paolo Bresciani. Use of loom for domain representation in a natural language dialogue system. Technical Report 9203-01, IRST, Povo, TN, Italy, March 1992. presented at LOOM Users Workshop, Los Angeles, March 23-24, 1992.

[Bresciani, 1994] Paolo Bresciani. Uniformly querying knowledge bases and data bases. In Baader et al. [1994], pages 58–62.

[Bresciani, 1995a] Paolo Bresciani. The challenge of integrating knowledge representation and databases. In *KARP-95 Second International Symposium on Knowledge Acquisition, Representation and Processing*, pages 16–18. Auburn University, AL, USA, September 1995. Extended version in IRST TR 9412-06, IRST, Povo TN (Italy).

[Bresciani, 1995b] Paolo Bresciani. Querying databases from description logics. In Baader et al. [1995], pages 1–4.

[Brodie, 1988] Michael L. Brodie. Future Intelligent Information Systems: AI and database technologies working together. In *Readings in Artificial Intelligence and Databases*, chapter Epilogue, pages 623–641. Morgan Kaufman, San Matteo, California, 1988.

[Buchheit *et al.*, 1994] Martin Buchheit, Manfred A. Jeusfeld, Werner Nutt, and Martin Staudt. Subsumption between queries to object-oriented databases. *Information Systems*, 19(1):33–54, 1994.

[CIKM '93, 1993] *Proceedings of Second Conference on Information and Knowledge Management (CIKM '93)*, 1993.

[Devanbu, 1993] Premkumar T. Devanbu. Translating description logics to information server queries. In CIKM '93 [1993].

[Goñi *et al.*, 1995] Alfredo Goñi, Arantza Illarramendi, and José Miguel Blanco. Caching in multidatabase system based on DL. In *Proceedings of the 1995 International Workshop on Description Logics*, pages 67–71, Rome, Italy, June 1995.

[Hsu and Knoblock, 1993] Chun-Nan Hsu and Craig A. Knoblock. Reformulating query plans for multidatabase systems. In CIKM '93 [1993].

[Kirk *et al.*, 1994] Thomas Kirk, Alon Y. Levy, and Divesh Srivastava. The information manifold. In *1994 CAIA Workshop on Intelligent Access to Libraries*, 1994.

[Levy *et al.*, 1994] Alon Y. Levy, Yehoshua Sagiv, and Divesh Srivastava. Toward efficient information gathering agents. In *Working Notes of the AAAI Spring Symposium "Software Agents"*, March 1994.

[MacGregor, 1991] R. MacGregor. Inside the LOOM description classifier. *SIGART Bulletin*, 2(3):88–92, 1991.

[McKay *et al.*, 1990] Don P. McKay, Tim W. Finin, and Anthony O'Hare. The intelligent database interface. In *Proc. of AAAI-90*, pages 677–684, Boston, MA, 1990.

[Pastor *et al.*, 1992] Jon A. Pastor, Donald P. McKay, and Timothy W. Finin. View-concepts: Knowledge-based access to databases. In *Proceedings of Second Conference on Information and Knowledge Management (CIKM '93)*, Baltimore, 1992.

[Schmiedel, 1994] Albrecht Schmiedel. Semantic indexing based on description logics. In Baader et al. [1994], pages 41–44.

[Tou *et al.*, 1982] F. Tou, M. Williams, R. Fikes, A. Henderson, and T. Malone. Rabbit: An intelligent database assistant. In *Proc. AAAI'82*, 1982.