The Knowledge Engineering Review, Vol. 10:1, 1995, 73–76

Reasoning about structured objects: Knowledge representation meets databases

FRANZ BAADER[1], MARTIN BUCHHEIT[2], MANFRED A. JEUSFELD[1] and WERNER NUTT[2]

[1] *RWTH Aachen, Ahornstr. 55, 52056 Aachen, Germany*
[2] *DFKI, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany*

1 Introduction

Structured objects are items with defined properties that are to be represented in a computer system. Research in Knowledge Representation (KR) and in Database Design (DB) has produced languages for describing structured objects. Although different in the particular means for defining properties, both areas share the goal of representing a part of the world in a structured way. Moreover, the rise of object-centred formalisms in the last decade has significantly influenced the convergence of languages.

This observation was the main motivation for the first workshop KRDB'94, held in conjunction with the German National Conference on Artificial Intelligence KI'94. The goal of the workshop was to provide a platform where experts of both areas can meet and cross-fertilize their work. In particular, we hoped to obtain a collection of viewpoints on structured objects and reasoning services required by applications. The call for papers resulted in submissions from France, Italy, Switzerland and Germany. With respect to the number of presentations, it was the largest of the 18 workshops at the KI'94.

2 Structured objects

It is not surprising that the perception of structured objects differs between DB and KR experts. Database systems require algorithms for efficient query evaluation on large databases containing relatively simple objects. Knowledge representation languages, on the other hand, emphasize expressiveness and complex inference mechanisms, but knowledge bases are usually relatively small. First, we summarize the DB view of structured objects, and then turn to the KR view.

2.1 Database objects

Even if one accepts that object-oriented databases are still restricted to a small market niche, the conceptual modelling of databases is dominated by object-centred languages. The presentations at the workshop supported one common viewpoint: objects are grouped into classes (types, concepts, sets), and classes specify properties (attributes, slots, components) of their objects. By including behavioural aspects (methods, operations) and storage alternatives (set vs. list, value vs. reference), these properties form the core of the data model of an object-oriented database.

2.2 Reasoning services in KR

Most contributions to the workshop concentrated on class description languages in the tradition of KL-ONE, also known as concept languages or description logics. In such a language one can build complex descriptions of classes (concepts) from atomic classes and attributes with the help of certain constructors (such as conjunction, disjunction, negation, restricted types of quantification, cardinality restrictions, etc.). Dialects are distinguished by the permitted constructors. Formally, such languages can be seen as (usually decidable) fragments of a first-order logic with unary and binary predicates only.

Concept languages are used for the intentional representation of an application domain. Fundamental reasoning services are testing satisfiability of a given knowledge base and determining subsumption between two classes, i.e. subclass–superclass relationships induced by the description. The structure of the objects belonging to the classes is expressed by the use of binary predicates (roles, attributes). There is a well-known trade-off between expressiveness of a concept language and complexity (or even decidability) of the reasoning problems, or as one participant put it: "Interesting languages are intractable, and tractable languages are not interesting."

3 Bridging the gap

The presentations of the workshop mentioned various ways of bridging the gap between the research in both areas.

Connections between conceptual database models and concept languages have been investigated since the late 1980s. As one result, reasoning techniques from knowledge representation became applicable to database tasks. For example, the problem of whether there exists a database for a given entity relationship diagram can be solved by encoding the diagram into a set of statements in a concept language and then checking the encoding for satisfiability. This idea has recently been extended to other conceptual database models like semantic data models and object-oriented data models. The latter cannot be represented completely because methods have no counterpart in concept languages.

Another area of cross-fertilization is the determination of view types in object-oriented databases. Loosely speaking, views are predefined queries to a database. They are part of the database schema. To determine whether an answer object of the view satisfies the type restrictions of a method, one can classify the view definition against the type system. This task can be solved by using subsumption algorithms for suitable concept languages.

Concept languages have been proposed for semantic query optimization for object-oriented databases. Here, the predicative specification of a query is separated into two conjuncts, a "structural part" that can be expressed in a concept language, and the remaining "dirty" part. The structural parts of two queries can be checked for subsumption, and this information can be used for optimization purposes.

Conversely, implementations of KR systems can profit from database technology. A common approach is to build hybrid systems with a database component where the instances of classes are stored. Under the additional assumption that all possible information about the instances is stored—the so-called Closed World Assumption, which is, however, not always justified in KR—the query mechanism of the database system can be used to compute inferences in the KR system.

A topic of common interest is the use of indexing techniques in knowledge and databases. Semantic indices based on class descriptions can be used to increase the performance of query answering, i.e. finding the set of individuals belonging to the class defined by the query. When a query is submitted then the most specific subsuming indices are used for optimization.

4 Invited talks

The talks of the two invited speakers gave excellent introductions into the topic. Both presented successful combinations of KR and DB techniques.

Maurizio Lenzerini from the University of Rome I, "La Sapienza", pointed out how inferences with concept languages can be applied to database tasks that require schema level reasoning.

The first application he mentioned was the analysis of conceptual database models specified in some object-oriented or entity-relationship like language. He presented a concept language that is rich enough to capture very expressive data models containing, for instance, extended is-a-relations (like Student is a subclass of all Persons who are not Professors) and cardinality restrictions (like a Student takes at most five Courses). To check whether a schema is free of contradictions, one translates it into concept language descriptions, and tests them for satisfiability. Since a database can only contain a finite number of instances, it is particularly interesting to test whether a description is satisfiable by a finite model. For such a task, standard techniques from logic are not applicable. However, there is an algorithm that is based on linear programming. Although the worst-case complexity of the satisfiability problem in the rich language is exponential, the algorithm runs in polynomial time when used for deciding the consistency of database schemas obeying some natural restrictions.

As a second application, Lenzerini proposed to represent so-called inter-schema knowledge between cooperative information systems. Assuming that the systems involved are described in an expressive entity relationship language, the problem is to relate the entities and relationships in one schema to those in another one. This can be done with the help of axioms stating that a complex description in terms of one schema is subsumed by a description in terms of the second schema. For example, one can say that the Students in system A are a subset of the Graduate-Students in system B. After a translation of the schemas into a concept language, inference techniques become available for consistency checks and for deriving consequences of the axioms.

Marc Scholl, from the University of Ulm, argued that the topic of views can link research in databases and knowledge representation. Views are prefabricated queries which are used to define more complex queries, to optimize queries by precomputing the answer to views, to enhance security and integrity of a database, and last, but not least, to integrate database schemas. Object-oriented databases, like the Cocoon system developed in Ulm, provide for schema languages that have many characteristics in common with concept languages. Views in such a system can be seen as new classes defined by the properties of their elements.

To find the right place of a view class in the class hierarchy of a schema, Cocoon uses a procedure similar to subsumption algorithms for concept languages. Views can be specified by an arbitrary algebraic query expression, which is too expressive a language to keep subsumption decidable. The implemented procedure is incomplete for the full problem, but works well for most real cases.

5 Discussion

As described above, there has already been a considerable amount of cross-fertilization between database and knowledge representation experts. Prototype systems like BACK, CLASSIC or KRIS in the area of KR, and CANDIDE, COCOON or ConceptBase in the DB area show the practical value of combined techniques. The workshop participants identified some interesting areas of future research:

- *Database views* There are several proposals to reason about view definitions by representing them in concept languages. How do they relate to traditional approaches? Can they be applied to relational, network, and hierarchical databases?
- *Distributed systems* With computer networks, the lack of information is often replaced by an information overload. Can information needs be directed to information sources in the same way as queries are related to views via subsumption tests?
- *Data mining* What query language is suitable for extracting hidden knowledge from large databases? Are concept languages expressive enough for this purpose?
- *Scaling up* For large information systems, there can be thousands of schema elements and view definitions. How can reasoning about such schemas be ensured to be tractable? Can schemas be modularised? Are there incremental algorithms that keep the effects of local changes small?
- *Embedding* How can the developed techniques be embedded into the software lifecycle? Do they mix well with existing software engineering techniques like structured analysis and abstract data types?

A vivid discussion arose on the topic of terminological cycles, i.e. definitions of concepts that refer to the defined concept. They have been excluded from most concept languages because of semantical ambiguities and algorithmic problems. Interestingly, borrowing the separation into schema and views known from database design, one can get rid of both the semantical and algorithmic problems.

The presentation of a data analysis practitioner surprised the audience. He reported that entity-relationship diagrams are not a suitable language for discussing a database schema with customers because database developers tend to prematurely translate them, at least mentally, to the relational level. Instead, a language only consisting of nodes and links was much more successful in his projects. The diagram is augmented afterwards by constraints. This technique was motivated by the use of semantic networks for knowledge representation in AI.

6 Conclusions

The workshop probably was the first organized occasion in Germany to discuss the subject of structure objects with participants from different disciplines in computer science.

The number of ideas and techniques was impressive, though sometimes limited in its focus. People coming from KR learned that there is a multitude of practical applications that are strongly related to their work. In many cases, properties of an application make methods tractable that are intractable in the general case.

We plan to continue the workshop in 1995, and to establish it as a forum for information exchange. The current trend to distributed systems and worldwide communication creates new challenges for conceptual modelling and reasoning. We believe that the combination of KR and DB is a good answer to these challenges.

The working notes of the workshop are available as technical document D-94-11 from DFKI, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany, or by ftp from ftp.dfki.uni-sb.de/pub/tacos/KRDB94/Long.

References

Baader, F, Buchheit, M, Jeusfeld, MA and Nutt, W (Eds.), 1994. "Reasoning about structured objects—knowledge representation meets databases". *Working notes of the KI'94 Workshop KRDB'94,* Saarbrücken, September 20–22; DFKI-Report No. D-94-11, DFKI Bibliothek, Postfach 2080, D-66123 Saarbrücken, Germany.

Baader, F and Hollunder, B, 1991. "KRIS: Knowledge Representation and Inference System." *SIGART Bulletin* **2** (3).

Beck, HW, Gala, SK and Navathe, SB, 1989. "Classification as query processing technique in the CANDIDE semantic data model." In: *Proc. 5th Int. Conf. on Data Engineering.*

Buchheit, M, Jeusfeld, MA, Nutt, W and Staudt, M, 1994. "Subsumption between queries to object-oriented databases". *Information Systems* **19** (1).

Borgida, A, Brachman, RJ, McGuiness, DL, and Resnick, LA, 1989. "CLASSIC—a structural data model for objects." In: *Proc. ACM SIGMOD.*

Donini, FM, Lenzerini, M, Nardi, D and Nutt, W, 1991. "Tractable concept languages." In: *Proc. 12th Int. Joint Conf. on Artificial Intelligence (IJCAI-91),* Sydney, Australia.

Scholl, MH, Laasch, C and Tresch, M, 1991. "Updatable views on object-oriented databases." In: *Proc. 2nd Int. Conf. on Deductive and Object-Oriented Databases; LNCS 566,* Springer-Verlag.

Staudt, M, Nissen, HW, and Jeusfeld, MA, 1994. "Query by class, rule, and concept". *Journal of Applied Intelligence* **4**.