Advantages of Classification/Subsumption for Query Processing in Semantic Databases

Sunit K. Gala, Shamkant B. Navathe and Howard W. Beck
Database Systems Research and Development Center
University of Florida, Gainesville, FL 32611
1989

1 Introduction

The inadequacies of the three classical data models in database systems, have spawned a large number of proposed semantic or object-oriented data models. Initially, these models were proposed for designers to describe a database at a conceptual level. This conceptual schema was then translated into a relational, network or hierarchical schema. Now semantic data models are increasingly being used as database systems directly. Thus, interest in strategies for querying such systems is emerging. The query languages being proposed for these semantic data models are usually based on some algebra and tend to have an SQL or QUEL-like syntax. Queries expressed in these languages cannot be described by the underlying data models, and hence the DDL and DML are distinct features of the system. We term this as the operational approach. The DML specifies manipulations to be performed on the database, which are then internally translated into a sequence of algebraic operations. It does not exploit the structural relationships or domain constraints which are specified in the schema of the semantic data model. The problems associated with this approach are:

1. the query language syntax has no bearing on the underlying data model,

2. the structural and semantic constraints of the model are not exploited to aid in query interpretation, query reuse or query reformulation,
3. being a fragmented approach, it is difficult to treat data objects, queries and views homogeneously through a single language,

4. view definitions in these languages lead to serious update problems,

5. even though the query language may be declarative, the user still has to think of how (in terms of algebraic operations) and from where (logical access paths, plus exact names of attributes and objects) to get the desired data before specifying the query.

We believe that the **DDL and DML dichotomy** in these database systems, coupled with a lack of reflection of the model semantics in the query languages or DMLs, is at the core of the above problems. This has been somewhat alleviated in the relational model by marrying it with logic as exemplified by the so-called deductive databases. But we are unaware of any such marriage between a semantic data model and logic. Thus, we would like to see a more homogeneous specification and behavior of objects, queries and views. This necessarily means collapsing the DDL and DML into a single coherent language. Further, this language must be formally specified so as to reflect the semantics of the data model. **Ideally, this data model should be a correct, complete and tractable model of computation, and yet be expressive enough to be useful.**

With this objective in mind, we propose that one way to enforce database integrity and also process queries is based on deductive reasoning about object definitions. Classification and subsumption functions are used for such reasoning about structural relationships among objects. The notions of classification and subsumption have been formally developed in a series of frame-based knowledge representation languages which explore the computational complexity of subsumption. These include FL and FL-, BACK, KL-ONE, NIKL and KANDOR. The disappointing conclusion of these studies is that subsumption becomes intractable unless the constructs of the language are carefully and narrowly constrained. However, even these constrained languages can be made as expressive as most semantic data models [BGN89] [BB89]. Semantic data models have not been concerned with these issues because they either rely on operational query languages, or just serve as conceptual modeling tools which do not support queries at all. We know of none which support terminological reasoning or address the tractability requirement, the only exceptions being CANDIDE [BGN89] and CLASSIC [BB89].
(note that subsumption in CANDIDE was later found to be co-NP-hard, and the tractability proof for CLASSIC is yet to be published).

Of these frame-based languages, FL-, KANDOR, and BACK explore the limit of expressibility and tractability, but out of these restricted languages only FL- has a subsumption function which can be executed in polynomial time. We have developed CANDIDE based on extensions to KANDOR in order to facilitate their applicability to data modeling. CANDIDE illustrates that querying by classification is a viable technique for querying semantic data models. In particular, we wish to emphasize the following:

1. The DDL and DML become a single language. Thus, the semantics of the database schema is automatically exploited for query processing.

2. Since query and view objects are treated as new class definitions, they are represented as object definitions and behave in the same way as data objects.

3. One can potentially take any model that supports structural definitions and use classification for computing queries. This brings two benefits - the first is that the class/object/entity definitions acquire a formal denotation via the subsumption function, and the second is that no new language has to be developed for querying the database.

4. Additionally, it has been our experience that the notions of subsumption and classification lend themselves naturally to a variety of application domains as diverse as natural language processing, heterogeneous systems integration, schema integration, schema evolution over populated databases, machine learning and family resemblance problem, to mention a few.

2 Classification and Subsumption for Query Computation

A significant departure from traditional database querying techniques is that we treat a query just as any other object described in the database schema. An object definition is prescriptive in the sense that :
1. It provides the minimal class description for any object to be considered a member of the corresponding type set, i.e., an object instance can have additional attributes beyond what is prescribed in each parent class, thus relaxing the fixed arity constraint.

2. Typing information is attached to the object class in terms of its superclasses, subclasses, and domain restrictions on its attribute labels (this is similarly true for object instances).

3. Since the attributes are labeled, we can relax a traditional constraint such as fixed position (ordering of attributes).

4. Object class definitions can provide necessary and sufficient conditions for class membership which can be used to deduce additional relationships among objects not specified by the user.

This interpretation must be based on the semantics of the subsumption relationship. A class F subsumes a class G if and only if every instance of G is also an instance of F, i.e., F is a superclass of G. This subsumption relationship is computed on the basis of whether the attribute constraints for class F logically imply the attribute constraints for class G. The classification operation can compute the missing relationships by controlled application of the subsumption function, and completely specify the class taxonomy. We have used this same classification process to compute the results of a query in CANDIDE. Classification can be viewed as the process of correctly locating a new object in an existing taxonomy. The correct location is immediately below the most specific classes which subsume the new class and immediately above the most general classes subsumed by this new class. A query object specification is classified against the complete taxonomy. The required object instances are obtained from the union of the set of instances of all the superclasses of the query object, subject to additional attribute constraints of the query object, plus the union of all instances of each subclass of the query object.

It is known that structural knowledge representation schemes bring in a syntactic structure to a well chosen subset of logic. Any statement in this sublanguage would then be a well-formed-formula (wff). If this wff satisfies certain constraints then it is a theorem in this subset of logic. Further, if these constraints are specified by the semantics of subsumption, then such
subsets of FOL are called terminological logics. Classification can also be viewed as the deduction of the most immediate super- and subclasses of a given class. However, when querying a large database by classification, the most important result is the set of instances belonging to the query class. Thus, identification of these instances must be highly optimized.

The syntactic structure of this logic has a well-defined BNF grammar which can be given an extensional or denotational semantics which is equivalent to the well-chosen subset of logic. We feel that these semantics can lead to an algebra which can retrieve instances very efficiently. This would be imperative in a very large database. Thus, it seems that some research in this direction would be required if we were to implement large database systems based on terminological logics.

Additionally, it seems that by giving a denotational semantics to the syntactic structure of our language, we can view classification as semantic type inferencing not different from that in programming languages. If we think of class/concept definitions as type declarations and instances/individuals as variables, classification can be thought of as a type inferencing mechanism. This seems to suggest that the terminological logic can provide persistent types in a programming language. But, of particular interest is that this fact seems to suggest a different way of implementing the A-box. Some inspiration in this direction can be drawn from LOGIN (a strongly typed logic programming language) and F-Logic (a strongly typed object-oriented logic programming language). Of course, the reason to augment a query language with a database programming language is to allow computation of arbitrary transactions which the query language itself is not capable of handling.

3 Comparison with Previous Query Processing Techniques

Querying by classification is the process of specifying a query object using the same definitional facility as data objects, and then searching for objects which are structurally and semantically related to this query object. Query processing is based on deductive inferencing about object structures rather than a procedural specification of operations. Query specification is entirely declarative in that the user need not provide any information on how the
query is to be executed. The user concentrates on describing the desired information. Inferencing techniques for matching are defined formally by the subsumption function which determines whether one object class is a subclass of another. Since subsumption is a form of terminological reasoning, the user can describe a query in terms which may be different from the exact terms under which the desired information is stored, so long as the meaning is similar. This contrasts with the SQL-type queries in which names of relations and attributes must be precisely specified. Dealing with concepts also implies that the database objects are not only descriptors for physical data, but must also represent the meaning of terms used to describe the data. Although semantic data models are capable of expressing these meanings, this aspect has not been exploited for query processing.

Querying using classification can be contrasted with the operational approach exemplified by SQL-type query languages. In the operational approach, a query is mapped into a sequence of algebraic operations applied to the objects in the database.

Internally, this is a procedural approach since the plan of query execution involves an exact sequence of operations. Externally, high-level data manipulation languages are used which are more declarative in that the user need not specify these steps. However, the user must still have a thorough understanding of the database schema, and must know the names of objects and attributes. Also, the user must be able to conceptualize the operations in general. For example, the user must specify access paths, or describe which relations need to be joined. A number of data manipulation languages based on the operational approach have been proposed for semantic data models.

ARGON is an information retrieval system built on top of KANDOR, but it still had to process some queries outside the classifier. It was the same case with RABBIT. But due to the extensions in CANDIDE as described above, we are able to handle certain queries which otherwise had to be processed by operations outside the classifier. This section would be incomplete without mentioning CLASSIC [BB89]. We share most of the arguments and concerns brought forth in [BB89]. CLASSIC is much richer since it has new features such as using concepts as partial descriptions of instances, and adding rules and co-reference constraints. It has also introduced many operators other than classification and subsumption, and the result of its query can be extensional or intensional. While all these are desirable features for a DBMS to have, we seem to differ in a few minute details. The distinction between in-
stances and classes should be kept sharp in traditional database applications since it would definitely improve processing efficiency. The idea is to have a kernel on top of which one builds layers of such complexity rather than weaving them into the subsume and classify functions. This would ensure no loss of functionality and at the same time efficiency of traditional processing does not deteriorate.

Finally, the concept matching approach can be contrasted with querying in logic databases. The search by the classifier is directed by the taxonomical relationships among objects, and the terms being unified can be complex objects, but the subsumption function is much less powerful than full first-order logic. Essentially, subsumption can be viewed as a *constrained inferencing technique* compared with logical queries. There has also been some effort on incorporating structure via types and inheritance into logic, as in F-logic and LOGIN. These are essentially programming languages and are too large and unwieldy to be considered as a typical DDL or DML.

## 4 Conclusions and On-going Research

We have presented arguments in favor of using classification as a query processing technique. Further details of how this is accomplished in our model CANDIDE are provided in [BGN89]. Our approach is different from existing ones in that:

1. we provide a uniform treatment of objects, queries and views, and

2. query computation is based on deductive reasoning about the structural relationships between objects.

We have implemented CANDIDE using Prolog and C in the POPLOG environment. Syntax-oriented object editors have been implemented in POP-11. This model is the basis for various on-going projects at the University of Florida. It seems to us that terminological reasoners are powerful and versatile tools which can be exploited in diverse application domains. Some of our projects include:

- an intelligent multi-media information retrieval system which supports natural language queries.
• the FIB (Federated Information Bases) project which involves a federation of loosely coupled databases and software modules which will be integrated using this data model.

• a database programming language to enable computing arbitrarily complex queries against the database which cannot be handled by the classifier (CANDIDE provides the persistent type mechanism for this language).

• making extensions and modifications to the CANDIDE language. It has been our experience that the max restriction can be removed without much loss, but role value maps or 'dot expressions' are quintessential. It is interesting to note that CLASSIC has role value maps and also does not have the max restriction.

• a schema integration tool which takes multiple CANDIDE schemas as input and generates a single CANDIDE schema.

• an algebra for efficient and optimal retrieval of instances. Basically, equivalent algebraic operators should replace many operations done by the classifier in identifying the correct instances. This would lead to a more efficient implementation.

• a broader examination of categorization theory is being made which looks beyond the necessary/sufficient definition notion of determining class membership. Family resemblance, inference mechanisms for natural kinds, and conceptual clustering can be accounted for by techniques which orthogonally augment the standard subsumption function.

Acknowledgements

Parts of this research have been supported by NSF Grant DMC 8415 385 and Florida High Technology and Industry Council Grant 87101421. We also acknowledge the present collaboration with Bellcore on the schema integration tool.
References


