KNOWLEDGE-BASED APPROACHES IN GIS

T R SMITH AND YE JIANG

This chapter provides a framework for understanding the application of knowledge-based techniques (KBT) in GIS. It is intended neither as a review of such techniques nor as a survey of their application. It is argued that full first-order logic is a proper theory on which to base such techniques. In terms of understanding the application of KBT, it is contended that expressive and computational power and computational efficiency are far less important than the ease with which applications may be built and with which users may interact with GIS. Current applications of KBT typically involve the use of rules in relation to the main functional components of GIS, with loose coupling between the rule base and the spatial database. The full value of KBT to GIS is likely to be realized only when they are applied in a systematic manner on the basis of formal logic, and when current semantic and optimization issues are settled. Such value will reside in the ability of knowledge-based GIS to model complex spatio-temporal phenomena.

INTRODUCTION

The application of knowledge-based techniques (KBT) in current GIS may appear as little more than the *ad hoc* application of techniques developed initially in the area of artificial intelligence. The goal of the present chapter is to provide a framework in which these applications may be viewed in a more valuable and systematic manner. The significance of a suitable framework for understanding applications of KBT lies in its ability to focus research on important issues.

Throughout the chapter the following view is taken:

1. Database systems (DBS), when implemented in terms of a specific domain of application, provide a model of the domain. A major function of a DBS is to make explicit various properties of the model.
2. GIS are best viewed as DBS that provide models of spatio-temporal domains.

While GIS are increasingly characterized by large volumes of spatial data and while data storage and retrieval are critical issues requiring much research, it may be convincingly argued that analytical and modelling capabilities are a discriminating factor for GIS. In particular, much of the information in a GIS is typically stored in implicit form, especially in the case of raster-based GIS. Hence, it is assumed that a major requirement of GIS is an ability to deduce relatively complex properties of domain models from information that is stored in the data and knowledge bases of the system. This viewpoint is important for understanding the application of KBT in GIS, since a KBT may be viewed as a set of tools that facilitate the construction of computational models of relatively complex domains and provide mechanisms for deriving properties of the domains.

This viewpoint may be refined and a KBT considered as a set of techniques that have been developed for representing 'knowledge' about some domain and for supporting procedures for deriving inferences about the domain from some 'knowledge base' (KB). Logics of various forms are increasingly being chosen as the theoretical basis for such techniques, and a great deal of current research in the highly relevant area of deductive databases (e.g. see Przymusinski 1989) is currently employing both subsets and extensions of full first order logic to provide such a basis. Other techniques that are frequently used, such as those relating to semantic networks, frames and production systems (e.g. see Barr and Feigenbaum 1982a, 1982b) may be usefully viewed as special cases of approaches based on first order logic (Nilsson 1980).

There has been a great deal of recent interest in the application of KBT in the area of DBS, particularly in relation to the use of representational languages involving rules. In particular, concepts such as logic programming, deductive databases, expert DBS and knowledge-based management systems have been developed (Mylopoulos 1986). Several factors must be considered in order to understand the application of KBT to DBS in general and to GIS in particular, including:

- the expressive and computational power of the representational language(s) in the DBS;
- the computational efficiency of the system; and
- the ease with which applications may be written and with which users interact with the system.

It is contended that, while expressive and computational power and computational efficiency are of critical significance, the ease with which applications may be written and the ease with which users may interact with the system are probably the key factors in explaining the application of KBT in DBS in general, and in GIS in particular.

This contention is supported by first arguing that knowledge is not fundamentally different from data, and that any distinction is best viewed in terms of the explicitness of the form of representation. In terms of this viewpoint, data are represented explicitly as the DB extension, while knowledge involves a more implicit and, therefore, a generally more compact, representation as the database intention.

Equivalently, the difference may be expressed in terms of the expressive and computational power of the representational language. This argument is convincing when data and knowledge are modelled in terms of first-order logic. It may be noted that KBT, as interpreted above, add nothing new to conventional techniques in terms of obtaining the full expressive and inferential power of the full first-order logic, while implementations of KBT have yet to be made computationally efficient. It is concluded that KBT are mainly of use in facilitating the building of applications and in facilitating the use of GIS in modelling application domains.

In the sections of the chapter following the presentation of these arguments, several architectures are described for DBS that facilitate the application of KBT. In particular, the discussion focuses on loosely coupled systems (Stonebraker and Hearst 1989) in which rule bases (RBs) are the dominant form of knowledge representation. Examples are then provided of the application of KBT in GIS in terms of various functional components of GIS, including acquisition, storage, access, analysis and interfaces.

THE THREE FACTORS

Before proceeding, it is of value to discuss briefly the three sets of factors listed above, as well as the nature of how change typically occurs in software systems. The function of a GIS is to answer queries. In this context, a query is simply a mapping from the database to some relation defined on the database. Hence the query language of the system is of fundamental significance. This is particularly the case in the context of the present chapter, since full first-order logic is viewed as the appropriate basis for interpreting the applicability of KBT. In relation to logic-based DBS, the query language may be viewed as a single language for expressing queries, data, integrity constraints, views, programs and specifications (Lloyd 1987).

Given this viewpoint, the three factors listed above may be seen as evaluative criteria relating to a single entity, namely the query language of the system. The expressive and computational power of the language relates to the class of functions (and hence queries) that can be expressed and computed in terms of the query language. Although query languages may be devised that are able to express non-computable functions, the only languages of practical interest are those that can compute Turing-computable functions. The essential point is that query languages may vary dramatically in their ability to express (and compute) different classes of queries, ranging from highly restricted languages on relations to languages having the power to express all Turing-computable queries. The efficiency associated with a query language essentially relates to the computational efficiency of the procedures that support the answering of queries expressed in the language. The ease with which applications may be written and the ease of user interaction are clearly related to both cognitive factors and the 'naturalness' with which a given query language matches human representations of some domain of interest. A major goal of the theory of query languages is to provide an understanding of how to design query languages that are natural to use, expressive and efficient in practice.

The current status of KBT in DBS (and in GIS in particular) may be partially understood in terms of how change typically occurs in software systems, since GIS are themselves in part software systems. Ullman (1986a) is followed in listing a typical sequence of events in which:

- a need is perceived;
- *ad hoc* approaches to programming solutions are found;
- the programming tricks are understood; and
- a second generation of researchers automate the programming process with the use of a high level programming language without tricks.

It would appear that the application of KBT in the area of GIS is currently in the stages of need perception and *ad hoc* approaches to programming solutions. This argument, coupled with the idea that the application of KBT is largely concerned with ease of building applications and the ease of system use, appears to explain the currently *ad hoc* nature of their application in GIS.

DATA AND KNOWLEDGE

In order to understand the significance of applications of KBT in GIS, the distinction between data and knowledge must be examined. A variety of such distinctions have been made, including:

1. When dealing with data some automated process can be relied upon to collect the material, while when dealing with knowledge expertise is required to collect the material (Smith and Smith 1977).
2. Data reflect the current state of the world at the level of instances while knowledge deals with abstractions and entity types (Wiederhold 1986).
3. Knowledge is represented in terms of rules.
4. A KB involves a richer semantics for interpretation than does a DB and contains knowledge about something. DBs are more concerned with efficient storage and retrieval (Brodie and Mylopoulos 1986).
5. Knowledge is used chiefly as an attribute of programming systems that support some form of declarative language, which is typically some form of logic (Ullman 1986b).
6. A KB supports recursive queries (Naqvi 1986).

While each of these viewpoints provides some insight, they are ambiguous.

An alternative viewpoint is that there is no essential distinction between data and knowledge. A more refined version of this viewpoint, and one adopted here, argues that: (1) there is a gradation between the expressiveness of representational languages in terms of the classes of statements that can be made concerning some domain; (2) facts are characterized by a restricted class of statements; and (3) knowledge is characterized by a larger class of statements. This view may be exemplified by taking the representational language to be some subset of full first-order logic or an extension of such logic. In order to provide some power to the argument, the adequacy of logic for both representing and making inferences about the data and knowledge in a GIS is briefly discussed.

From a syntactic point of view, a first-order theory consists of an alphabet (namely the constants, variables, function symbols, predicate symbols, connectives, quantifiers and punctuation symbols), a set of axioms and a set of inference rules. The deduction of the theorems of a theory may be characterized in terms of those formulae that are logical consequences of the axioms of the theory using the rules of inference. In relation to this framework, an answer to some query to a DBS may be interpreted in terms of the logical consequences of a set of axioms and a deduction may be viewed in terms of the computation of some function. In applications of first-order logic to database theory (see Lloyd 1987), a standard approach is to use formulae that have the form of rules:

$$A \leftarrow A_1,\ldots,A_n$$

in which the As are predicates or relations. In particular, facts are special rules having the form:

$$A \leftarrow$$

and queries take the form:

$$\leftarrow A_1,\ldots,A_n$$

If it is permitted to use function symbols in the arguments of the predicates and recursion in the rules (i.e. the same predicate symbol can occur on both sides of the implication sign), then any computable function can be expressed in terms of a set of such rules, given a suitable encoding (see Lloyd 1987), which in turn may require the use of negation. It is, therefore, clear that such a representation is conceptually completely adequate for any GIS and any query to a GIS, although there are many semantic and practical issues that currently require resolution before any such implementation is truly feasible. Proper subsets of full first-order logic can be implemented for DB applications with greater ease, but there is a cost in terms of expressive and inferential power. For example, standard (relational) DBs may be viewed as a finite collection of ground atomic sentences ('facts'), each being represented in terms of a single n-place predicate symbol, a set of n individual terms with no variables and a limited inferential capability.

Given the preceding discussion, it is not unreasonable to view 'knowledge' as information that is generally representable in terms of the general formulae of full first-order logic, while facts may be viewed as a proper subset of ground state atomic sentences. This point of view is consistent with the distinctions (2), (3), (5) and (6) above, and is, in fact, the approach that is taken in both logic programming and deductive databases. The significance of the preceding discussion is that it focuses attention on issues relating to expressive power and ease of expression. For example, the use of the (extended Horn-clause) rules described above appears to facilitate greatly the representation of certain classes of knowledge. Hence KBT may be viewed, in general, as techniques that are concerned with the ease with which information that is more than facts, as defined above, may be expressed and used in a deductive manner.

KBT IN NON-SPATIAL DBS

Non-spatial DBS have received far more attention from researchers than have GIS and the application of KBT in such systems has received correspondingly greater attention. Since developments in such systems are strongly influential with respect to current research efforts in GIS, and since GIS may be viewed as a special case of general DBS, there is a need to identify the relative importance of the three factors listed above that are important for the adoption of KBT in non-spatial GIS. It is assumed that the same relative importance applies to GIS, particularly since GIS typically model more complex domains than standard DBS. The discussion is developed by focusing first on standard relational DBS, then on extensions to relational systems and, finally, on logic-based and object-oriented approaches.

It is of interest to note that foundations of the theory of deductive databases may be found in the seminal paper by Codd (1970), in which a formal basis for relational databases was first outlined. A relational database is a collection of individual facts equipped with the capability to manipulate efficiently (update) its contents and to answer queries about it. Typically, relational algebra or relational calculus, which is first-order logic interpreted for relations, is used to implement these functions. Concerning the expressiveness of the representational language in standard relational systems, the tuples ('facts') of relational tables are, as noted above, equivalent to ground atomic sentences. The standard relational query language is not capable of representing recursive queries, such as finding a full set of nested, political regions that contain a given point. The lack of expressive power of the query language is partially related to the fact that the relational calculus is insufficiently powerful to compute transitive closures. In restricted domains of application, however, in which the computational limitations are not a problem, it is relatively easy to write applications, since user interaction in such domains is greatly enhanced by the separation of the 'what' and the 'how' of the querying process in terms of a declarative query language and query optimization techniques. Constraining the expressive and computational power of the query language has led to great computational efficiency, as a result of the use of data independence and optimization techniques.

One solution to the problem of lack of expressive power is to embed the query language in a host language that supports such functionally with iteration. Even with the support of a host language that possesses full computational power, however, standard relational systems are not easy to apply in non-standard domains of application that include, for example, spatial and statistical data (e.g. see Korth and Silberschatz 1986). Applications are hard to build and the systems are difficult to use, since it is difficult to model complex domains involving space and time as well as nested relationships in terms of relational tables. In particular, the expression of queries concerning complex (i.e. nested) objects is difficult, while the number of joins required to answer such queries raises major efficiency issues.

In order to overcome such limitations, extensions have been made to the relational model. Some extensions have involved extending the query language to support some form of iteration or recursion operation. For example, the query language of INGRES (QUEL) has been extended to QUEL* by adding an operation that executes a sequence of QUEL commands until the DB no longer changes. While QUEL* is therefore computationally complete (Varvel and Shapiro 1989), problems remain with respect to the ease with which applications may be written, the ease of user interaction and efficiency. Because it has proven difficult to deal with non-standard applications with such extensions, it may be concluded that linguistic expressive and computational power are not the key elements in explaining the introduction of KBT into non-spatial DBS.

Other extensions to the relational model that have been introduced for coping with non-standard applications have included the use of abstract data types (ADT) and rule-based techniques. POSTGRES (Stonebraker and Hanson 1988), for example, is an extension to INGRES that has been designed with applications to spatial data in mind, and involves the use of ADT, procedural data types and the use of rules. However, there are still major problems relating to ease of application building, ease of use and efficiency. Major alternatives to extensions of the relational model include (Ullman 1988):

- logic-based approaches;
- object-oriented approaches.

It is noteworthy that these approaches share common threads in their heritage, particularly in terms of KBT developed in artificial intelligence research. Object orientation, for example, traces part of its heritage to the research on semantic networks and frames that was central to many KBT, while predicate calculus was used early as a knowledge representation technique in artificial intelligence research, serving in fact as a unifying language for semantic networks and frames. Furthermore, it is becoming increasingly clear that logic-based approaches essentially include relational DBS as a proper subset.

In relation to the logic-based approach (see Przymusinski 1989), researchers came to realize by the mid-1970s that the capabilities of relational databases were quite limited by their inability to handle deductive and incomplete information. Deductive reasoning is of value in deducing new information from facts and deductive rules which can be included in a database. The need to deal with incomplete information is particularly evident in the case of disjunctive and negative information. Relational databases have no capabilities for storing and handling general deductive rules or for dealing with incomplete information. Deductive databases, however, can store and manipulate deductive rules of reasoning as well as data and are able to answer queries based on logical derivation coupled with some mechanism for handling incomplete information (e.g. Gallaire, Minker and Nicolas 1984; Minker 1988b). Logic programming was based on Kowalski's principle of separating logic and control (Kowalski 1974, 1987). Because of the formal development of logic programming in the late 1970s and early 1980s (Lloyd 1987), it has become clear that logic programming and deductive databases are closely related (Minker 1988a). They are based on the common idea of representing knowledge in terms of logic, and, in particular, of providing computers with a logical specification of the knowledge that is independent of any particular implementation, context free and easy to manipulate.

Logic-based approaches may be viewed in terms of a declarative language involving rules. In terms of limited implementations, such as PROLOG, the use of rules and the uniform representation of 'facts' and rules makes for ease of user application, although the language is not fully declarative in the sense that some rule ordering is required of the user. It is of interest to note that the use of rules has long been thought to be a 'natural' form of representation for human users (Nilsson 1980). The success of this approach depends on assigning precise meaning or semantics to any logic program in order to provide its declarative specification, which can then be compared to the output of some computation. This does not mean that such a computation must be based on some logical proof procedure, but only that logic is the final arbiter of correctness. Finding a suitable declarative semantics of deductive databases and logic programs is a critical research problem, while other major research issues relate to computational efficiency.

Although the object-oriented approach is not yet clearly defined, it may be viewed as a programming system that combines the data manipulation and host languages to support nested objects, encapsulation and object identity. Hence object orientation may be viewed in terms of a language with capabilities for defining ADTs. While object-oriented approaches, in the general sense, have full expressive and computational power, their greatest strength probably lies in the naturalness with which applications may be written and the ease with which users may interact with such systems, since the objects and relations of the object-oriented model may be chosen to reflect the user's model of the domain of application. There is, as yet, no consensus about the definition of an object-oriented DBS (see Kim 1990) and in particular about their relationship to relational and logic-based systems. There are also major concerns about computational efficiency at the present time.

Since questions of expressive and computational power are not fundamentally at issue in any of the approaches, and since computational efficiency is a major issue in all approaches relating to non-standard domains, and in particular for the logic-based approaches that are so highly related to KBT, it is concluded that the ease of building applications and the ease of system use are probably the key factors in explaining the introduction of KBT into such DBS. Hence, the serious use of KBT in GIS is also apparently dictated by similar considerations.

INCORPORATION OF KBT INTO DBS

The introduction of KBT into DB technology has typically taken the form of expert system (ES) shells, production system (PS) languages and logic programming languages (Brodie *et al.* 1988), although there are other forms, such as the use of constraints (Morgenstern *et al.* 1988) for describing regularities in an application domain. The basic choice criterion, apart from familiarity, appears to relate to the ease of capture of the semantics of a given domain. These approaches are all characterized by rules, which some researchers regard as facilitating the construction of applications and the ease of user interaction (Nilsson 1980). It is quite possible that such systems represent stages in the evolution of DBS towards full, logic-based systems employing well-formed formulae taking the form of rules. Hence, most of the attention is focused on rule systems.

There are various current architectures that permit the integration of rule-based systems into DBS in general and GIS in particular (Stonebraker and Hearst 1989). The following are possible:

1. To enhance either a rule-based system with limited DBMS capabilities, such as data access, concurrency control or security, or enhance a DBMS with rule-based system capabilities, such as knowledge acquisition and representation techniques and reasoning.

2. To employ 'loose coupling', in which an application is written using an ES shell. This shell typically supports application logic, presentation services and navigation rules and manages the rule base. The shell is then

extended to support calls on an external data manager, and hence is like any other DB user.

3. To employ 'tight coupling' to facilitate communication between the rule-based system and the DBMS by building one system as a shell about the other. Hence, either the rule-based system works as a shell around the DBMS or the DBMS can work as a shell around the rule-based system.
4. To build a fully integrated system, such as a true deductive DBS.

The loose coupling approach is the norm at present in both spatial and non-spatial DB applications, while tight coupling and full integration are difficult to achieve, although POSTGRES is one example of a tightly coupled system. Problems with the loose coupling approach include the following (Naqvi and Tsur 1989; Stonebraker and Hearst 1989):

1. There is semantic mismatch between the language of the front end, which is typically procedural, and the language at the back end, which is typically declarative. This leads to efficiency problems because global optimization is no longer possible.
2. There is a mismatch in the granularity of the data objects between the front end and the back end, since the front end typically deals with a single tuple while the back end deals with a set of tuples.
3. The design of a loosely coupled system can be limited by shortcomings at either end. For example, if the back end cannot handle recursive queries, the front end is limited.
4. The rule base is main-memory resident, hence, for example, the rule base can be lost if the address space of the shell goes away, while rules cannot easily be saved.
5. Dynamic data in which there are changes to values relating to facts create problems. In particular, there is a need to maintain consistency in a cache of DB objects.
6. Non-partitionable applications, in which the whole fact base needs to be accessed before the completion of an inference, lead to poor performance.

Tightly coupled systems are being designed to overcome these problems. In one approach based on logic programming, for example, the objects declared and manipulated by the system are the same as the objects stored and manipulated in the database, while there is no front end/back end distinction (Naqvi and Tsur 1989). In relation to tight coupling, it is noted that a DBMS can be extended to manage an RB; can deal with dynamic data by awakening rules as required by changes in the data; and can handle large spatial DBS.

Attention is now focused on loosely coupled systems, since this architecture has become the standard in spatial DB applications and is of significance in current GIS. The rule processing capabilities of such systems serve several functions, including the following:

1. The provision of the services of a traditional ES.
2. The triggering of external actions in response to changes in the data (including data analysis).
3. The provision of state and transition constraints, including referential and semantic integrity constraints.

The first set of services can be implemented in terms of a standard inference engine, such as a forward or backward chaining, operating on a set of rules. In terms of the second set of services, the predicate of a rule may, for example, contain an analytical procedure that produces output when certain data conditions are mentioned. The third set of services include, for example, the enforcement of semantic integrity constraints which are not generally provided by current DBMS.

A key factor determining the efficiency of DBS, which incorporates the three sets of services, is the mechanism for selecting and firing the appropriate rules. Three mechanisms are currently available, including: (1) indexing the predicates in the LHS of the rules; (2) sequencing over rules and computing logical intersections; and (3) the use of database locks. No completely satisfactory solution has yet been found, and rule selection is an area of active research.

GIS AND KBT

GIS technology is less mature in both conceptional and implementational terms than non-spatial DBS technology. Correspondingly, the introduction of KBT into GIS is at a less advanced stage and, for the most part, may be classified in terms of either system enhancement (a GIS with rule-based capabilities, or an ES with spatial data handling capabilities) or of loose coupling of rule-based applications and GIS. So far, the concepts of tightly coupled systems or knowledge base management systems have not been exploited, while rule-processing capabilities have typically been limited to the provision of the services of a traditional ES.

The most important approaches to the application of KBT in GIS may be discussed and characterized as:

- acquisition;
- storage;
- access;
- analysis and processing;
- interfaces.

Before exemplifying the application of KBT to each of these components, the way the applications relate to the three factors determining the general adoption of KBT is summarized. First, it is clear that KBT, defined in terms of languages supporting knowledge representation and deduction, do not add any new expressive or computational power to languages currently used in GIS. Second, applications of KBT have not been focused on increasing the computational efficiency of GIS. In relation to acquisition, KBT enhance the building of applications in terms of automating procedures for acquiring knowledge. In relation to storage, they have been used to ease the construction of, and access to, very small spatial databases. In relation to access, KBT facilitate the use of GIS by such mechanisms as the use of metadata and query optimization. In relation to analysis, the construction of applications is enhanced by the use of rule-based ES that are loosely coupled to a GIS database. Such systems also provide easy-to-use interfaces to GIS. Finally, in relation to interfaces, KBT facilitate the construction of natural language interfaces, which themselves facilitate the use of GIS.

Acquisition

Automating procedures for data and knowledge acquisition eases the task of system construction. One area of applicability of KBT to this task involves the general concept of 'data dredging' (Naqvi and Tsur 1989), which essentially involves finding and storing regularities in data for various purposes. In particular, inductive learning may be viewed as a special case of data dredging. One goal of data dredging is to reduce the difficulty of constructing knowledge and databases, particularly the acquisition of facts, rules and constraints that increase both the efficiency of the system (in terms of learning DB integrity constraints) and the domain modelling capabilities of the system. In particular, such enhancements make explicit relevant parts of the DB intention; they generate rules for performing inference and constraints for query optimization; and they serve to define new objects and modify data and rules.

Two particular strategies for data dredging of interest in the current context are inductive learning and explanation-based learning. Inductive learning is the derivation of abstractions and generalizations from particular instances. Four basic rules underlie most inductive learning procedures, including the method of agreement, the method of difference, the method of residues and the method of concomitant variation. The procedures also involve criteria for evaluating rules. Major limitations include the number of examples required, the lack of efficient algorithms and the problem of noisy data. Inductive learning has been used, for example, in non-GIS applications to learn rules for ES and DB integrity constraints (Hoff, Michalski and Stepp 1983), while in GIS applications, it has been used in KBGIS-II (Smith *et al.* 1987) to generate generalized descriptions of complex spatial objects, which are then placed in framelike structures.

In explanation-based learning (Mitchell, Keller and Kedar-Cabelli 1986), the approach is to analyse a single example in terms of a specific application domain and produce and justify a valid generalization of the example. The inputs to such a procedure include a goal concept, a training example, domain theory and operational criteria.

There do not appear to be current applications of this procedure in the field of GIS.

It should be noted the data dredging essentially expands the information base of the system. Key issues for research are the lack of efficient procedures and problems of maintenance.

Storage

KBT currently find little application to issues relating to the storage of large volumes of spatial data, presumably because of efficiency issues. Semantic networks, frames and rule bases have, however, been employed to represent both data instances and data abstractions in GIS. Since these are not particularly efficient storage structures, their use must be predicated upon the ease of construction and the ease of use.

Access

Access to data in GIS, particularly in the case of large DBs, is a major area of application for KBT. There is currently great interest in the use of metadata (which provides a model of a DB in terms of structure and content) as knowledge that can be employed in facilitating content-based search. KBT languages are increasingly being used to represent metadata.

A key application of KBT with respect to access involves query optimization. The use of high-level, declarative query languages generally requires the use of optimization procedures to enhance system efficiency, since users do not supply information for DB navigation. The success of the relational DBS is in large part due to query optimization techniques, while a major bottleneck for the development of logic-based DBS is the lack of good optimization techniques. Query optimization is of potentially great significance for large-scale GIS dealing with complex (i.e. multicomponent and nested) spatial objects with many constraints between the sub-objects and in which the objects are often implicitly represented in a number of data layers.

A major approach to query optimization involves the application of knowledge, typically in the form of rules or integrity constraints, relating to equivalence relations, containment relationships, expected value ranges, sorting orders, functional dependencies (Hammer and Zdonik 1980) and special transformations (Siegel 1989). Such knowledge is typically used to transform a query for more efficient processing. A key requirement in this process is to maintain semantic equivalence between the original and transformed queries.

Static query optimization employs both domain-independent knowledge and domain-dependent (semantic) knowledge to produce a semantically equivalent query and an efficient sequence of operations that provide lower execution costs. Particular transformations are effected through constraint introduction, constraint removal (Siegel 1989) and constraint replacement. These transformations are implemented in terms of a variety of approaches, including theorem proving (Chakravarthy 1985), graph theoretic (Jarke 1984) or heuristic approaches (Siegel 1989). POSTGRES (Stonebraker and Hansen 1988), for example, uses rules to implement such transformations.

Dynamic optimization is an alternative to static optimization and enforces constraints during the search procedure. Such optimization has been employed in KBGIS-II (Smith *et al.* 1987), for example, where the basic problem is to retrieve complex spatial objects. Forward checking is used dynamically to enforce domain constraints. During the search, the values for any variable are examined sequentially and constraints are explicitly computed in order to check whether the value selected from the domain satisfies the constraints on the variable. Spatial constraint propagation is used to replace the explicit checking of constraints during backtracking by geometrical search within constrained areas of the database. Two forms of dynamic update occur during the search procedure: domain-dependent rules are used to produce semantically equivalent queries about sub-objects, while sub-object search may be reordered according to criteria relating to sub-object existence (as determined by search) and sub-object complexity and frequency (as determined in a knowledge base). Frames, semantic networks and rules are the structures used to represent such knowledge.

Analysis and Processing

Most applications of KBT relating to analysis and processing have involved rule-based approaches and

ES technology. Such applications typically use ES that are, at most, loosely coupled to the spatial DB component of a GIS.

Application of PS and ES techniques in domains involving geographical data has been widespread. Davis and Clark (1989), for example, compiled a bibliography of over 200 articles, written between 1976 and 1989, describing ES applications in the areas of natural resource management that included agriculture, geographical data handling, forestry, environmental law, environmental planning, water resources, and wildlife and vegetation modelling. It is of interest to note that apart from natural language applications, few of the articles describe artificial intelligence techniques other than ES and many of the ES were not coupled to the spatial data-handling capabilities of a GIS.

Robinson, Frank and Karimi (1987) provide a survey of 20 systems involving ES that have been built in order to support various GIS operations in the area of resource management. Many of these systems are loosely coupled to the data handling capabilities of some GIS and many involve the use of ES shell languages. In several applications, the ES may be viewed as an applications-oriented 'interface' to the GIS. For example, ASPENEX (Morse 1987) is an ES providing interface services and interfacing to a GIS (MOSS). The system provides rules on aspen management, while MOSS provides information on the characteristics of aspen stands. Special software provides communication between the ES and the GIS. Among other applications of ES technology are automated interpretation of aerial photography (ACRONYM, Brooks 1983), change-detection in LANDSAT images (FES, Goldberg, Alvo and Karam 1984), automated terrain feature extraction and decision making in economic and urban systems (URBYS, Tanic 1986; GEDDEX, Chandra and Goran 1986).

There have been several applications of ES modules in GIS that have been based upon ES that were originally developed for non-GIS applications. Katz, for example, has essentially emulated the PROSPECTOR system in MAPS, which is a raster-based GIS. In particular, MAPS incorporates Bayesian, fuzzy and certainty factor techniques, which constitute essential elements in PROSPECTOR, in order to process applications such as mineral exploration.

Some ES modules have been constructed using logic programming techniques. For example, Franklin and Wu (1987) have formulated the polygon overlay in PROLOG, while Webster (1989) has shown in detail how point-in-polygon queries may be expressed in the predicate calculus and answered by resolution theorem proving in a PROLOG environment. Yan (1988) has provided a general exposition of some of the elements of the theory of logic programming in the context of GIS. In practical terms, such an approach is of immediate value for systems with little spatial data, but the current lack of development of such a theory in the context of large spatial data sets and complex spatial objects, as well as current problems relating to efficiency, presently inhibit the development of large scale GIS employing such technology.

Interfaces

While some of the ES discussed above may be viewed as providing an interface to a GIS (e.g. ASPENEX), their orientation has related to ease of application building by separating domain modelling and analysis considerations from DB considerations. There are applications of KBT that correspond to more traditional interface concepts. For example, LOBSTER (Frank 1984) is a query language for GIS serving as an interface to an object-oriented, network spatial DBMS (PANDA). The interface is logic based and is implemented in PROLOG syntax. The interface to KBGIS-II (Smith *et al.* 1987) is a declarative query language based on the predicate calculus.

The most obvious applicability of KBT in this area, however, relates to natural language (NL) interfaces. An NL interface offers users efficient access to a GIS, since details of the system are hidden from the user and the user is not constrained by the disparity between the simplicity of natural language constructs for spatial objects and relations and their complex, low-level representations in GIS (Hendrix *et al.* 1978).

The success of NL interfaces in non-spatial approaches is in part a result of their limitation to relational databases and SQL (e.g. see Bates, Moser and Stallard 1984). Such interfaces have typically involved the following:

1. Conceptional models of underlying database architecture and contents, and automated

translations of a deep meaning representation of a user's input into SQL.

2. Extensible systems enabling non-language specialists to apply the language system to different application domains within the relational database architecture.
3. Syntactic and semantic modules providing a wide linguistic coverage of natural language.
4. Framelike representation language systems that support a robust model of a user's view of some domain.
5. Automated processes for 'meta-describing' a particular database organization.

Of great importance is the consensus among NL researchers that domain knowledge and its representation is fundamental to the building of a 'working' language system. Success in the area of NL interfaces for GIS, as in most other areas of AI, depends on limiting the domain of application and having a good representation of this domain in the computer. Hence KBT, particularly framelike and semantic network representation schemes, have played an important role in such interfaces because of their ease of use for representing domains of application. An NL interface requires specific domain knowledge of various types including:

1. Generic knowledge about spatial domains, such as the information about spatial relationships analysed by linguists.
2. Knowledge about specific domains of application.
3. Knowledge about the GIS to which the NL interface is connected, such as knowledge about how the data are stored and the operators available in the GIS.

KBT may be used to apply such knowledge in a layered architecture for an NL interface. In such an architecture, the user's query is translated into some meaning representation language. The transformed query is then sent to a mapping module that transforms the query into a GIS-specific set of operators that are sufficient to answer the query. The set of operators is then executed by the relevant GIS. The transformations between NL, meaning language representation and operators is accomplished by the application of knowledge in three language knowledge bases, namely a lexicon, a syntactic rule base and a semantic rule base, as well as knowledge bases relating respectively to general world knowledge, specific GIS system knowledge and specific GIS application domain knowledge. A target representation for the NL input is a network of interrelated domain concepts and relations formally represented in the system spatial domain model employing KBT.

Apart from NL interfaces, there are other applications of KBT that relate to display production. For example, ES have been constructed that automate various cartographic procedures for display, such as name placement (AUTONAP, Freeman and Ahn 1984) and map generalization (MAPEX). These topics are discussed elsewhere in this volume (see Freeman 1991 in this volume; also Muller 1991 in this volume).

CONCLUSIONS

It is concluded that the application of KBT in GIS is largely motivated by issues relating to the ease of constructing applications and the ease of system use, rather than by issues relating to expressive and computational power and efficiency. This viewpoint is of value in understanding current and future applications of KBT. It is to be expected that there will be many more applications in all components of GIS. Also, a more systematic approach to such applications is likely, particularly in terms of object-oriented approaches in the short term and logic-based approaches in the long term. The major force driving such applications will be the desire to construct more powerful DB models of complex spatio-temporal phenomena.

REFERENCES

Barr A, Feigenbaum E A (eds.) (1982a) *The Handbook of Artificial Intelligence*, Vol. I. HeurisTech Press and William Kaufmann, Stanford

Barr A, Feigenbaum E A (eds.) (1982b) *The Handbook of Artificial Intelligence*, Vol. II. HeurisTech Press and William Kaufmann, Stanford

Bates M, Moser M G, Stallard D (1984) The IRUS transportable natural language database interface. *Technical Report*. Bolt, Beranek and Newman
Brodie M, Mylopoulos J (1986) Knowledge bases versus databases. In: Brodie M L, Mylopoulos J (eds.) *On Knowledge Base Management Systems: integrating artificial intelligence and database technologies*. Springer-Verlag, New York, pp. 83–6
Brodie M L, Bobrow D, Lesser V, Madnick S, Tsichritzis D, Hewitt C (1988) Future artificial intelligence requirements for intelligent database systems. In: Kerschberg L (ed.) *Proceedings from the Second International Conference on Expert Database Systems, Tysons Corner, Virginia, 25–27 April 1988*. The Benjamin/Cummings Publishing Company, pp. 45–62
Brooks R A (1983) Model-based three-dimensional interpretations of two-dimensional images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-5**: 140–50

Chakravarthy U S (1985) *Semantic Query Optimization in Deductive Databases*. Unpublished PhD thesis, University of Maryland
Chandra N, Goran W (1986) Steps toward a knowledge-based geographical data analysis system. In: Optiz B (ed.) *Geographic Information Systems in Government*. A Deepak Publishing, Hampton
Codd E F (1970) A relational model of data for large shared data banks. *Communications of the ACM* **13** (6): 377–87

Davis J R, Clark J L (1989) A selective bibliography of expert systems in natural resource management. *AI Applications in Natural Resource Management*. Moscow, Idaho

Frank A U (1984) Extending a network database with PROLOG. *Proceedings of the First International Workshop on Expert Database Systems, Kiawah Island, SC, October*, pp. 665–74
Franklin W R, Wu P Y F (1987) A polygon overlay system in PROLOG. *Proceedings of AUTOCARTO 8*. ASPRS/ACSM, Falls Church Virginia, pp. 97–106
Freeman H (1991) Computer name placement. In: Maguire D J, Goodchild M F, Rhind D W (eds.) *Geographical Information Systems: principles and applications*. Longman, London, pp. 445–56, Vol 1
Freeman H, Ahn J (1984) AUTONAP – an expert system for automatic map name placement. *Proceedings of the 1st International Symposium on Spatial Data Handling*. International Geographical Union, Ohio, pp. 544–71

Gallaire H, Minker J, Nicolas J (1984) Logic and databases: a deductive approach. *ACM Computing Surveys* **16**: 153–85
Goldberg M, Alvo M, Karam G (1984) The analysis of LANDSAT imagery using an expert system: forestry applications. *Proceedings of AUTOCARTO 6*. ACSM/ASPRS, Falls Church Virginia, pp. 493–503

Hammer M, Zdonik S B (1980) Knowledge-based query processing. *Journal of the IEEE* **6**: 137–46
Hendrix G, Sacerdoti E, Sagalowicz D, Slocum J (1978) Developing a natural language interface to complex data. *ACM Transactions on Database Systems* **3** (2)
Hoff W A, Michalski R S, Stepp R E (1983) INDUCE 3: a program for learning structural descriptions from examples. *Final Draft Report*. Chicago, Department of Computer Science, Artificial Intelligence Lab, University of Illinois

Jarke M (1984) Semantic query optimization in expert systems and database systems. *Proceedings of First International Conference on Expert Database Systems*, pp. 467–82

Katz S (1988) Emulating the Prospector Expert System with a raster GIS. In: Thomas H F (ed.) *GIS: Integrating Technology and Geoscience Applications*. National Resource Center, Connecticut, pp. 27–8
Korth H F, Silberschatz A (1986) *Database System Concepts*. McGraw-Hill, New York
Kowalski R (1974) Predicate logic as a programming language. *Proceedings of IFIP-74*, pp. 569–574
Kowalski R (1987) Algorithm = logic + control. *Communications of the ACM* **22**: 424–36

Lloyd J W (1987) *Foundations of Logic Programming*. Springer-Verlag, New York

Minker J (1988a) *Foundations of Deductive Databases and Logic Programming*. Morgan Kaufmann, Los Altos
Minker J (1988b) Perspectives in deductive databases. *Journal of Logic Programming* **5** (1): 33–60
Mitchell T M, Keller R M, Kedar-Cabelli S T (1986) Explanation-based generalization: a unifying view. *Machine Learning* **1**: 47–80
Morgenstern M, Borgida A, Lassez C, Maier D, Wiederhold G (1988) Constraint-based systems: knowledge about data. In: Kerschberg L (ed.) *Proceedings of the Second International Conference on Expert Database Systems, Tysons Corner, Virginia 25–27 April 1988*. Benjamin/Cummings Publishing Company, pp. 23–44
Morse B W (1987) Expert interface to a geographic information system. *Proceedings of AUTOCARTO 8*. ACSM/ASPRS, Falls Church Virginia, pp. 535–41
Muller J-C (1991) Generalization of spatial databases. In: Maguire D J, Goodchild M F, Rhind D W (eds.) *Geographical Information Systems: principles and applications*. Longman, London, pp. 457–75, Vol 1
Mylopoulos J (1986) On knowledge base management systems. In: Brodie M L, Mylopoulos J (eds.) *On Knowledge Base Management Systems: integrating artificial intelligence and database technologies*. Springer-Verlag, New York, pp. 3–8

Naqvi S (1986) Discussion. In: Brodie M L, Mylopoulos J (eds.) *On Knowledge Base Management Systems: integrating artificial intelligence and database technologies*. Springer-Verlag, New York, pp. 93
Naqvi S, Tsur S (1989) *A Logical Language for Data and Knowledge Bases*. Computer Science Press, New York

Nilsson N J (1980) *Principles of Artificial Intelligence.* Tioga Publishing Co., Palo Alto

Przymusinski T C (1989) On the declarative and procedural semantics of logic programs. *Journal of Automated Reasoning* **5**: 167–205

Robinson V B, Frank A U, Karimi H A (1987) Expert systems for geographic information systems in resource management. *AI Applications in Natural Resource Management* **1** (1): 47–57

Siegel M D (1989) Automatic rule derivation for semantic query optimization. In: Kerschberg L (ed.) *Proceedings of Second International Conference on Expert Database Systems, Tysons Corner, 25–27 April 1988.* Benjamin/Cummings Publishing Company, pp. 69–98

Smith J M, Smith D C P (1977) Database abstractions: aggregation and generalization. *Association for Computing Machinery Transactions on Database Systems* **2** (2): 105–33

Smith T R, Peuquet D J, Menon S, Agarwal P (1987) KBGIS-II: a knowledge-based geographical information system. *International Journal of Geographical Information Systems* **1** (2): 149–72

Stonebraker M, Hanson E N (1988) The POSTGRES rule manager. *IEEE Transactions on Software Engineering* **14** (7): 897–907

Stonebraker M, Hearst M (1989) Future trends in expert database systems. In: Kerschberg L (ed.) *Expert Database Systems.* The Benjamin/Cummings Publishing Company, Redwood City, pp. 3–20

Tanic E (1986) Urban planning and artificial intelligence: the URBYS system. *Computers, Environment, and Urban Systems* **10** (3–4): 135–46

Ullman J (1986a) Logic and database systems. In: Brodie M L, Mylopoulos J (eds.) *On Knowledge Base Management Systems: integrating artificial intelligence and database technologies.* Springer-Verlag, New York, pp. 121–24

Ullman J (1986b) An approach to processing queries in a logic-based query language. In: Brodie M L, Mylopoulos J (eds.) *On Knowledge Base Management Systems: integrating artificial intelligence and database technologies.* Springer-Verlag, New York, pp. 147–64

Ullman J D (1988) *Principles of Database and Knowledge-Base Systems.* Computer Science Press, Rockville

Varvel D A, Shapiro L (1989) The computational completeness of extended database query languages. *Journal of the IEEE* **15** (5): 632–8

Webster C J (1989) Point-in-polygon processing in PROLOG. *Technical Reports in Geo-Information Systems 17.* Wales and the South West RRL, University of Wales College of Cardiff

Wiederhold G (1986) Knowledge versus data. In: Brodie M L, Mylopoulos J (eds.) *On Knowledge Base Management Systems: integrating artificial intelligence and database technologies.* Springer-Verlag, New York, pp. 77–82

Yan S Y (1988) *A Logic Foundation for Expert Geographic Database Systems.* Melbourne, Australia, Department of Computer Science, University of Melbourne.