DELINEATING OPERATIONS FOR VISUALIZATION AND ANALYSIS OF SPACE-TIME DATA IN GIS

L. Qian, M. Wachowicz, D. Peuquet, A. MacEachren

Department of Geography
The Pennsylvania State University
University Park, PA 16802 USA

ABSTRACT

A foundation upon which both database operations and visualization operations can be defined in a unified way is needed to provide a more versatile and human-oriented means for interfacing with GIS. This paper proposes a taxonomy of operations using a set-based information model that is application and data model independent. Such operations integrate both GIS functionality and visualization tasks at the operational level (i.e. subdivision of conceptual goals into a set of distinct actions to be applied to data in order to achieve the goals).

1 INTRODUCTION

GIS have become essential tools for gaining understanding of our complex and dynamic world. Historically, using GIS required a significant amount of training in order to become familiarized with the substantial number of unique and distinct commands contained within a particular GIS. Commands were often individually implemented as users of a particular package required additional functionality, and were typically implemented as fairly direct function calls to independent functions. As perhaps an extreme example, by 1990, Arc/Info had incorporated over 2,000 separate commands. In contrast to GIS, an integrated and standardized language for Database Management Systems was in common use by the early 1980's. Besides being far easier to learn and understand, the integrated elements and explicit syntactic rules of a formalized language has the advantage of providing a more flexible tool for expressing ad-hoc commands to the system. This be came an important issue as commercial GIS became used for an increasingly wide range of (sometimes unanticipated) applications.

Dana Tomlin provided a simple and elegant, natural language-like command language for his MAP analysis package in 1983 [Tom83], the basic structure of which has been adopted in a number of GIS products including Arc/Info GRID and GRASS Mapcalc. There have also been many efforts over the past decade to extend the standard DBMS language, SQL, to incorporate spatial operations, for example [Ege89]. Although they are a step in the right direction, both extended SQL and the Map Algebra approach have some serious drawbacks. Tomlin's Map Algebra in its present form [Tom90] is restricted to use with a "ridded, or at least a tessellated, data model. It also takes training and practice to understand many of the operations (e.g., "spread", drain") and how to combine them to achieve a desired operation as may be expressed in everyday language, such as determining a least-cost route. SQL with spatial-operation extensions exhibit the same characteristics, but in this case all operations are based upon data organized in tables, where the relational data model offers little help in expressing complex GIS queries. This is not surprise since SQL also was not initially designed to handle spatial data, and has even been argued that the SQL framework may be inappropriate for an interactive query language for a GIS [Ege92]. Neither approach was designed to deal with temporal operations, although a number of (non-spatial) temporal SQL extensions have been proposed ([Sar90], [TCGe93]).

Albrecht [Alb96] found that people learn to conceptualize their GIS use in terms defined by the software, and using a world view reflected in a specific data model or data models used by the GIS. This is observed in conversations of GIS users, such as a forester who may point across a valley to a stand of trees and speak of "that polygon". What happens is that the user is forced to view the world, and the problem he or she wishes to address, using the specific world view presented by the software. This is perhaps the root cause for the "unfriendliness" of current GIS and the need for a significant training process before users feel "comfortable" using a specific system. The user subsequently begins to adopt this view generally, regardless of whether or not it is the most effective for dealing with a particular task.

A more versatile and human-oriented means for interfacing with the visual display and the database must be developed that can handle a diverse range of data from different sources, and that seems "natural" and easy to use when humans (who are domain, rather than GIS, specialists) perform complex tasks. In the current paper, we discuss a framework for deriving a small set of data model and application independent operations as a basis for such a modern GIS interface. This is a report upon one aspect of a larger effort, called the Apoala Project, to develop and assess a prototype Temporal Geographic Information System with integrated spatio-

temporal visualization capabilities. The prototype includes a multi-representational database model, called the TRIAD model, that combines object-based, location-based, and time-based representations of environmental data in a tightly coupled manner linked to a multi-dimensional visualization capability which facilitates exploratory and task-based spatiotemporal analysis [Peu94], [PQ96].

2 DERIVATION OF A SET OF UNIVERSAL SPACE/TIME OPERATIONS

To avoid the constraint exhibited in previous efforts, the emphasis in the current research is to approach the definition of universal space/time operations at multiple levels. As is true in the design of any information system (whether or not it is directed to geographic data) design should be approached at multiple levels. In the context of geographic visualization system design, Howard and MacEachren [HM96] proposed three levels: conceptual (determination of what to do with a system from user perspective), operational (subdivision of conceptual-level goals into a set of distinct actions to be applied to data in order to achieve the goals), and implementational (implementation of operations in a specific hardware, software, and application environment). These correspond to the three levels proposed by Peuquet [Peu84] as multiple levels of abstraction from "reality"; data, or information model to the data structure and the file structure (Figure 1).

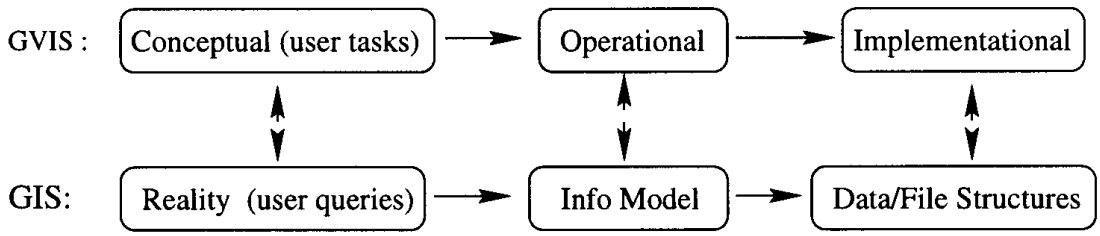

Figure 1: Multilevel GVIS operations and GIS abstractions

The operational level (operators on the information model) is our concern here. Within this level, an initial distinction that can be made is between operations as conceived of by system users (what Knapp calls "mental" actions - e.g., "if the precipitation value appears to be out of line, check the elevation at that point," [Kna95, page 361] and operations implemented in software needed to carry out the conceptualized mental task. Within the context of the implementational level, it may be useful-and indeed necessary-to make a further distinction among those operations that act directly upon the stored data in the database and those that manipulate the data visualization on the display. These distinctions may not, and often should not, be apparent to the user (i.e., at the conceptual level). It is our assertion that there also need not be a distinction among the set of operations for visualization and database tasks at the operational level. In other words, the essential logical processes involved are equivalent. The only difference is whether the operation is applied to the database or to the graphical display.

Both visualization tasks and database queries are carried out by first separating a generally-described conceptual task into a sequence of specific and discrete operations. These discrete operations can then be translated into (visualization or database) programmable algorithms. The overall process is one of increasing specificity. Operations have historically been defined at the implementational (algorithm) level in GIS until recently [Alb96, Ope95, Wor94]. Although this may provide a clear solution for a particular system's design context and given a specific data model, it can obfuscate what the common operations really are. It is the middle, operational level, that promises to provide a taxonomy of clearly distinct and distinguishable operations that are also independent of application domain and of implementational context.

3 THE SET-BASED INFORMATION MODEL

To provide a foundation upon which both general database operations and visualization operations can be defined in a unified way, we need to have a generic and flexible information model. The principal aim of information models is meeting the information needs of users on the basis of stored data. A user requests information in the form of a query (conceptual level) and by interpreting the data output in response to the query the user receives the needed information. The query is transformed into an ensemble of operations

(operational level) which are formulated in terms of the information model. Such operations are then transformed into algorithms (implementational level) producing the answer to the query.

What we employ for this purpose is a set-based information model. We use this particular model because the set is a well-defined and very powerful mathematic concept and construct. Using sets (with the extension of allowing duplicate elements in a set), we can avoid much of the restrictions imposed by the pure relational model on a DBMS [Gil94] or GIS. At the same time sets serve well in abstracting most geographic visualization tasks/operations as will be shown later.

3.1 Basic Constructs: Sets, Elements and Operations

A set can be defined as a collection of elements (of same type). Elements of a set can themselves be sets, thus allowing a hierarchical information model. Each element of a set, as defined within the current context, has a series of attributes that describe its state. Operations are procedures that act upon upon one, all or a group of individual elements of a given set, or upon elements of more than one set, or upon sets themselves.

Elements in a set all have the same type (sharing a common set of attributes). We define three types of sets based on the most distinctive attributes their elements have, given the TRIAD model:

Set Type L: A set of this type consists of individual elements that have one or more attributes describing their *locations.* Examples of such elements include regularly or irregularly spaced sample points. Or vector-type geometries. It is essential to note that *location* here is not constrained to geographic location. *Location is* broadly defined, and can be a location in the temporal dimension or even a position in some color space.

Set Type O: A set of this type consists of individual elements which have one or more attributes describing its non-locational properties. Such elements can be, for example, photographs of wildlife species, acts of legislation, etc.

Set Type C: A set of this type consists of individual elements that are events which have attributes describing the *changes* over time that have occurred to a particular element or elements of a set of Type L or O. Although an element of this set type can possess several attributes, the essential property for an element of type C is a timestamp designating the time of change. This type of set is necessary if we are to directly portray the dynamics of elements in either Set Type L or Set Type O. Set Type C is really a derivative of Set type L in that the essential properties include both a *location* and some other attribute that describes the change that occurs.

The above three set types provide a flexible and generic basis for delineating a taxonomy of visualization and database operations within the set-based information model. Such operations are categorized into a hierarchy of **set operations, element operations, and set-element operations.** The system is deemed to be open, therefore it can be expanded with new operations. The following section describes each of these three operational categories.

4 UNIVERSAL OPERATIONS BASED ON THE SET MODEL

4.1 Set Operations

Set operations are those that always operate at the set level, and always involve all elements of the participating set(s).

Union, Difference, and Intersection: These three common set operations require two sets as operands and have a new set as a result. These operations can be applied only to pairs of sets which are of the same type, L, O or C.

The *Union* operation merges the elements belonging to either one of the operand sets. The mathematic expression of *Union* operation $\cup$ can be expressed as:

$$A \cup B = \{x | x \in A \ or \ x \in B\}$$

where A and B are two sets of same type.

The *Difference* operation distinguishes the elements that belong to the first operand set but not to the second operand set. Mathematically,

$$A - B = \{x | x \in A \ and \ x \notin B\}$$

The *Intersection* operation provides a resulting set which has all elements belonging to both operand sets. Mathematically,

$$A \cap B = \{x | x \in A \ and \ x \in B\}$$

Boolean set operations, comparison visualization operations with multiple images, hue/color use to distinguish different surfaces in a volume rendering, are some examples of these such set operations.

Select The *Select* operation accepts some requirements/predicates expressed in terms of elements' attributes, and pick up all those elements that satisfy the requirements. It examines every element in the participating sets. The select statement in SQL, visual operations by each an object is distinctly identified, are examples of the Select operation. The formula for *Select:*

$$\sigma_C(S) = \{x | x \in S \ and \ x \ has \ properties \ satisfying \ criteria \ C\}$$

where S can be one or more sets. If S contains multiple sets, $x \in S$ means x should be in at least one of the sets of S.

Associate The *Associate* operation relates pairs of elements, either belonging to the same set or multiple different sets. Some examples are successive magnification of images associating detail to the overall image, and the join operator in SQL. The formal definition of *Associate is:*

$$\Phi_C(S) = \{(x, y) | x \in S \ and \ y \in S\}$$

where (x, y) is meaningful under criteria C and S can be one or more sets.

Group The *Group* operation assembles the elements of a set into groups, with each group serving as a single element in the new resulting set. Clustering and the SQL GROUP-BY clause are some examples of this operation. Group can be formally defined as below:

$$\Omega_C(A) = \{\{x_1, x_2, \ldots, x_k\} | x_i \in A\}$$

where x_i has properties that satisfy criteria C, for all $1 \leq i \leq k$, and A is one set.

The overall purpose of all set operations described above is to introduce a meaningful framework that can be applied to arbitrary sets of type L, O. and C. They are universal operations that in principle allow a "high-level" natural coding of widely used visualization and GIS operations. The usage of these set operations will be determined by the type of task to be solved as well as the efficiency of their support at the implementation level.

Note that in the above formulas, the terms "satisfy", "meaningful" and "criteria" haven't been formally defined. The further clarification, if at all necessary, will be part of our future work. But the existence of such vagueness does not impair the genericity of the operations since the definitions of them merely provide a framework, where the details are up to the implementational level operations to cope with.

4.2 Element Operations

Element operations operate upon individual elements of a set, and/or the relationship between elements of one or several sets. Four main operations have been envisaged as following: Signify, Measure/Topology, Transformation, and Composing.

Signify The *Signify* operation determines how elements should be displayed, reported, or visualized. Some examples are visualization operations that create isosurfaces, glyphs, time sequences, and 3-D images.

Measure/Topology The *Measure/Topology* operation ascertains the spatial and temporal relationships among elements. Neighborhood, connectivity, and distance/direction operations are some examples.

Transformation The *Transformation* operation changes the configuration of an element. For example the locational representation of an element can be "transformed" from one coordinate system to another. The display of it can be transformed between various colors schemes and so on.

Compose The *Compose* operation combines several elements into a single complex element. Composite objects and nested relations operations are some examples.

The main advantage of using element operations is that all such operations do not depend on the order in which the elements of the set are taken neither on the types of sets an element belong to. This gives the flexibility to operate the elements in an " independent" way, without any consideration to the semantics of the set types which have been encoded as type L, O. and C.

4.3 Set-Element Operations

Set-Element operations operate on the relationships between the elements and the sets to which they belong. Basically, two main operations have been identified as being *Membership* and *Super-Sets.* They are applicable to operand sets of any type (type L, type 0, or type C).

Membership The *Membership* operation obtains the status of an element as being a member of a set. It is a unary operation on a setelement condition whose value is a set consisting of only one element which has the value of the operand [Can you simplify this sentence?]

Super-Sets The *Super-Sets* operation determines whether one set is a sub-set of another set. It is a binary operation having a resulting set with all elements of the operand sets. Inheritance operations in object-relational databases [Sto96] are some examples of this operation.

5 PROPERTIES OF THE OPERATIONS FOR THE SETBASED INFORMATION MODEL

Having described the operations in the previous sections, we can now summarize three main properties which correspond to the characteristics of the overall operational level in our set-based information model. They are:

Universality: Due to the generic set-based model, our definition of the operations are universal and applicable to a variety of entities in both database and visualization contexts. The more general the operations are, the simpler their implementations are, and to a greater extent the principal goal of modeling is achieved, since the mental process of users is supported by the operations within a query.

Data Independence: The issue of data independence is primordial to defining operations that not rely on applications, storage structures, and access methods. Data independence can be regarded as a property interrelated with key problems in parallel and distributed data processing [QP97].

Support for integrated GIS/VIS: The operations as defined in this paper interface both GIS and geographic visualization systems in a unified way. As such they offer a great potential for fully integrating the (currently separate) visualization functionalities and the processing/analysis functionalities in a modern GIS.

6 CONCLUSION

In this paper we delineated a taxonomy of universal operations based on a set-based information model. It should be noted that none of the operations described is our invention; they have been around in many different contexts. The purpose of the current effort, however, is to define a complete set of operations that are common to both GIS database analysis operations and also for visualization tasks. Our future work is to further refine this taxonomy and implement a prototype system in which all of the universal operations will be mapped into various algorithms and functions. The mapping is deemed to be one to many, that means for each universal operation an integrated graphic query and visualization will be supported.

7 ACKNOWLEDGMENTS

This research was sponsored by the U.S. Environmental Protection Agency (EPA), under Grant R825195-01-0, and by the National Science Foundation under Grant FAW NSF 90-27.

REFERENCES

[Alb96] J. Albrecht. Universal Analytical GIS Operations. A Task-Oriented Systematization of Data Structure - Independent GIS Functionality Leading Towards a Geographic Modeling Language. ISPA Mitteilungen, 1996.

[Ege89] Max J. Egenhofer. A spatial sql dialect, 1989.

[Ege92] Max J. Egenhofer. Why not sql! International Journal of Geographical Information Systems, 6(2):71-85, 1992.

[Gil94] Mikhail M. Gilula. The Set Model for Database and Information Systems. ACM Press, Wokingham, England, 1994.

[HM96] David Howard and Alan M. MacEachren. Interface design for geographic visualization: Tools for representing reliability. Cartography and GIS, 23(2):59-77, 1996.

[Kna95] Loey Knapp. A task analysis approach to the visualization of geographic data. In Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems, pages 355-371. Kluwer Academic Publishers, Netherlands, 1995.

[Ope95] S. Openshaw. Two exploratory space-time attribute pattern analysers relevant to gist In Spatial Analysis and GIS, pages 83-104, 1995.

[Peu84] Donna J. Peuquet. A conceptual framework and comparison of spatial data models. Cartographica, 21:66 -113, 1984.

[Peu94] Donna J. Peuquet. It's about time: A conceptual framework for the representation of temporal dynamics in geographic information systems. Annals of the Association of American Geographers, 84(3):441-461, 1994.

[PQ96] Donna J. Peuquet and Liujian Qian. An integrated database design for temporal gist In Seventh International Symposium on Spatial Data Handling, pages 2-1 to 2-11, Delft, Holland, 1996. Taylor Francis.

[QP97] Liujian Qian and Donna J. Peuquet. A systematic strategy for high performance gist In Auto-Carto 13, volume 5, pages 145-154, Seattle, 1997. ACSM/ASPRS.

[Sar90] Nandlal Sarda. Extensions to SQL for historical databases. IEEE Transactions on Knowledge and Data Engineering, 2(2):220230, 1990.

[Sto96] M. Stonebraker. Object-Relational DBMS. Morgan Kaufmann Publishers, 1996.

[TCGe93] Abdullah Uz Tansel, James Clifford, Shashi Gadia, and etc. Temporal Databases. Database Systems and Applications. Benjamin/Cummings Publishing Co., Redwood City, CA, 1993.

[Tom83] C. Dana Tomlin. Digital Cartographic Modeling Techniques in Environmental Planning. Yale University, Graduate School, New Haven, CT, 1983.

[Tom90] C. Dana Tomlin. Geographic Information Systems and Cartographic Modeling. Prentice Hall, Englewood Cliffs, NJ, 1990.

[Wor94] Michael Worboys. Unifying the spatial and temporal components of geographical information. SDH'94, 1994.