ON THE DESIGN OF GEOGRAPHIC INFORMATION SYSTEM PROCEDURES

By

J. Armando Guevara, Ph.D.
Software Engineer

Environmental Systems Research Institute, Inc.
380 New York St.
Redlands, California 92373

Telephone: (714) 793-2853 Ext. 208

AUTO/CARTO 9
April 2-7, 1989 Baltimore, MD

ABSTRACT

This paper identifies the building blocks that have played a major role in the design and implementation of current geographic information system procedures. It then examines and proposes the following six continuity concepts as unifying elements of an evolutionary GIS:

1. Functional continuity: the ability for a GIS to have a transparent functional flow of control.
2. Data base continuity: the ability of a GIS to manage giant amounts of data on a distributed system as one logical data base and have multi-user access.
3. Data structure continuity: the coexistence of vector, lattice, and raster data structures under one data model.
4. Knowledge continuity: the utilization of artificial intelligence techniques to create data base model usage schemas and create application procedures.
5. Human interface continuity: what makes a good GIS interface.
6. Data transfer continuity: the ability of a GIS to exist and transfer data independent of the hardware platform.

INTRODUCTION

A geographic information system (GIS) is composed of a set of building blocks termed geographic information system procedures. A geographic information system procedure is an abstract algorithmic function of a GIS that allows one to select, process, and update elements from a spatial data structure (SDS) and/or spatial data base (Guevara, 1983). Based on this, a GIS can be defined as a model composed of a set of objects (the spatial data structures) and a set of operations (GISP) that perform transformations and/or queries on the spatial objects. Unlike any other information system, GIS has the particular characteristic that its operations are mainly of spatial nature, thus a GIS is part of a major group of systems called Spatial Information Systems (SIS). The elements modeled by a SIS are generally imbedded in two-dimensional space and in some instances in two-and-a-half and three-dimensional space. In addition to the elements it manipulates being spatially located, the elements themselves possess a set of attributes that can be qualitatively or quantitatively defined. These attributes can not only give a description of the spatial elements, but can also become time components (changes in time of the spatial data base). Given this particular nature, GISP must be able to both query/transform the spatial elements and the attributes associated with those elements.

GISP are categorized to be the primitive operators of a GIS. In this sense GISP can be:

a)	SELECTORS	- Capture, select spatial data.
b)	RECOGNIZERS	- Structure/search spatial data.
c)	PROCESSORS	- Process spatial data.
d)	TRANSFORMERS	- Output spatial data.

This paper identifies the major building blocks that have played a key role in the design and implementation of GISP:

1. The use of geometry as the mechanism to digitally model the location of spatial elements.

2. The study of computational geometry; a better understanding of the digital representation of geometric algorithms and their numerical aberrations.

3. The use of topology to digitally model the relationship among data elements.

4. The local processing concept: the managing of large amounts of spatial data under limited RAM.

5. The fuzzy intersection concept: used to make the polygon overlay problem tractable at the implementation level.

6. The geometric simplification concept: used to simplify the geometric complexity of GISP objects.

7. Data base management systems: the relational model.

In retrospect, these concepts have served their purpose and are now the cornerstones of many implemented GIS. What is required now is to evaluate what is needed next in this growing and demanding technology. This paper then examines and proposes the following six continuity concepts as elements of a unified and evolutionary GIS:

1. Functional continuity: the ability for a GIS to have a transparent functional flow of control.

2. Data base continuity: the ability of a GIS to manage giant amounts of data on a distributed system as one logical data base and have multi-user access.

3. Data structure continuity: the coexistence of vector, lattice, and raster data structures under one data model.

4. Knowledge continuity: the utilization of artificial intelligence techniques to create data base model usage schemas and create applications procedures.

5. Human interface continuity: what makes a good GIS interface.

6. Data transfer continuity: the ability of a GIS to exist and transfer data independent of the hardware platform.

BACKGROUND HISTORY

It has been almost eight years since the first introduction of the ARC/INFO system. This system introduced for the first time a widely distributed and used operational GIS. Conceptual aspects of GIS have been around now for over 25 years. The gap between theory and practice began to be broken in the mid 70's and the technology has really taken off in the latter part of the 80's. If we classify the major breakthroughs that created this bridge we have:

a. The formal study of geometric algorithms via computational geometry (the study of spatial searching schemas, the study of spatial data structures) (Shamos et. al. 1976).

b. The use of topology to establish the spatial context for the geometric elements being digitally represented.

c. The organization of the spatial elements in a Data Base Management System.

d. The advent of high-power-low-cost computers.

FUNCTIONAL COMPONENTS OF GIS DESIGN

The most important concepts introduced in the functional design of a GIS have been that of the Local Processing Concept (Chrisman, 1976) and Fuzzy Overlay (White, 1978; Guevara, 1983). The Local Processing Concept made tractable at the implementation level the processing of large amounts of data given a limited amount of RAM by using the spatial properties of the data of location and orientation. The Fuzzy Overlay concept allowed for the implementation of the most powerful function of a GIS, that of data integration.

As knowledge was gained on the behavior of spatial algorithms, a functional categorization emerged that did away with functional continuity. Functional continuity refers to the ability in a GIS to be able to access any data set (or portion thereof in a seamless data base) and apply operators without the system losing track of the data environment and history of the operations performed. Functional continuity would allow access to all GISP within one process environment. Although this has a tremendous power, it is not without its user interface complications.

The functional categorization introduced gave way to the basic architecture of a GIS: data input, data base management, analysis, and output. Within each category, different means have been created to handle the user interface. Menu and command driven functions have become the main ways of interaction. The functions have a proper protocol to internally deal with the processes. Some are action driven while others are environment driven.

Action driven functions produce an immediate feedback to the user (e.g., draw a map). Environment driven functions have a cumulative effect that ends in an action driven function. Action driven functions are easy to explain and use. Environment driven functions pose a variety of user interface problems.

A functionally continuous GIS would be mostly environment driven. Such a system would require of knowledge environment and function tracking procedures. Recognizing the importance of user interaction with a GIS is the concern of the Human Interface Continuity Principle and is key to the life appreciation or depreciation of the system (how easy or complicated it is to learn and use).

DATA MODEL COMPONENTS OF GIS DESIGN

The most important concepts introduced in GIS that allowed one to digitally model spatial relations was that of topology (Corbett, 1975) and the relational data base model (Codd, 1970). It is interesting to find that it really has not been until recently that the power of this notion has been widely accepted at the implementation level of GIS. ODYSSEY was the first system to implement it (Dutton, 1978), then ARC/INFO (Aronson et al. 1983).

The various data structures introduced to handle geographic data (Morton sequences, Peano curves, quad trees, R-trees, B-trees, etc.) and their general definition and/or implementation (vector, raster, lattice) were to guide the definition of the GIS data model in the sense of only dealing with one particular data structure at a time. To achieve a continuous data model in the true perspective of not just spatial continuity but data integration also, the design of a GIS system must take into account the integration and management of all these data types (data structures). This would allow a GIS system to handle planimetric data, surface data, and imagery data.

In the early 70's the relational data model was introduced along with mechanisms to express relations between stored items (a relational algebra). Because the ubiquitous geographic matrix (rows, location, columns, description) fitted so naturally the relational model, this made the transition quite natural and simple to implement. A one-to-one relationship between the geometry and the descriptive data could be implicitly accomplished.

SPATIAL DATA MODELS

The ultimate task of a GIS is to model some aspect of a spatial reality. The model should include enough information that would allow its user to obtain answers to queries and infer situations that otherwise would not be possible. We can identify two types of models:

a. a generic functional model

b. a specific derived model.

A generic functional model (GFM) is a model made of basic spatial primitives: points, lines, and areas. The model holds descriptive data about the primitives, but does not know about existing relationships. The model is functionally driven (i.e., any further inference about the data aside from primitive location and basic description is obtained via spatial operators). The GFM is an open model that requires only very basic knowledge about the spatial primitives being stored.

A specific derived model (SDM) is a model derived from established relationships among the spatial primitives, and a linkage is created among compounded spatial primitives and their descriptive data. The SDM requires a well-understood knowledge of how the GIS is going to be used and what it is going to model.

The relational approach to spatial data handling falls under the GFM category, while the object-oriented approach falls under the SDM. It is important to understand that these two models are not mutually exclusive (i.e., a GFM can be used to support a SDM). However, the absence of an underlying GFM in a SDM raises flexibility and performance issues.

The GFM has the following characteristics:

a. It should allow for dynamic relationship construction via spatial operators.

b. Compounding of spatial primitives should be done efficiently without restrictions or constraints. The compounding would still yield a (more complex) GFM.

Internally, the GFM follows a similar structure to that described in Guptill (1986) with the exception that the lowest level of the model relationships are not explicitly stored.

The SDM has the following characteristics:

a. Relationships between spatial primitives are pre-established in the model based on behavioral, procedural, and transactional facts. These facts make the SDM schema.

b. Mutations on the spatial primitives should be done efficiently. Mutations such as aggregation (compounding), disaggregation (uncompounding) would still yield a (more complex or simpler) SDM.

The SDM would be the basic model for object-oriented transactions as those described in Kjerne (1986) and fundamented in Cox (1986) and Bertrand (1988).

The GFM and SDM should allow for the following types of data base queries:

a. Spatial Context: Given an unambiguous geometric definition, extract from the data base all elements selected by the geometric definition.

b. Spatial Conditional Context: Given an unambiguous geometric definition and a condition expressed in terms of the stored descriptive data, extract from the data base all elements selected by the geometric definition and that suffice the descriptive condition given.

c. Descriptive Context: Given a descriptive data element, extract from the data base all elements that match the one given.

d. Descriptive Conditional Context: Given a descriptive data element and a condition expressed in terms of the given element, extract from the data base all elements selected that suffice the descriptive condition given.

The conjunction of a GFM and a SDM would give the user the ability to perform spatial operations at various levels of complexity and integration. GFM and SDM bring the ability for a GIS to be flexible and schema independent.

Finally, both the GFM and the SDM should maintain the data base continuity concept (i.e., preserve the notion of a continuous physical space underlying the data model).

TOWARD AN ADAPTABLE SPATIAL PROCESSING ARCHITECTURE

A modern GIS is expected to be able to integrate a different variety of data sources; these data sources will be used in many ways and also under a wide range of support decision making. The nature of separate user views of the same data base accompanies a series of (sometimes conflicting) demands to the GIS designer that must somehow be met to guarantee the usefulness and longevity of the system. In synthesis, a GIS is a multidisciplinary tool that must allow for interdisciplinary support. Specialized spatial information systems are not multidisciplinary tools, thus are very restrictive in regards to what can be done with them.

An Adaptable Spatial Processing Architecture (ASPA) is what is needed to meet the demands of both multidisciplinary and specialized applications. ASPA fundamentals are based on a GFM that has a set of functional (GISP) primitives clearly defined that allow the automatic construction of a SDM. ASPA has to be designed based on the six continuity criterions given above. In this respect, ASPA would be an expert monitor based on a high level language consisting of spatial operators that have definable hierarchical constructs. These spatial operators can be organized following a programmable schema that would allow to generate the SDM. ASPA would work in conjunction with a data base management system (DBMS). The DBMS would respond to both spatial and non-spatial operators. The heart of ASPA and the DBMS would be GFM.

The spatial operators and spatial data structures that ASPA is built upon are based on the five basic software engineering principles of modularity, encapsulation, localization, uniformity, and confirmability (Jensen et al., 1979) applied through the concept of abstraction at the design level of the SDM (Guevara, 1981).

Levels of Abstraction

Levels of abstraction were first defined by Dijkstra (1969). They provide a conceptual framework for achieving a clear and logical design for a system. The entire system is conceived as a hierarchy of levels, the lowest levels being the most specific. Each level supports an important abstraction.

Each level of abstraction is composed of a group of related functions. One or more of these functions may be used by functions belonging to other levels; these are the external functions. There may also be internal functions which are used only within the level to perform certain tasks common to all work being performed by the level and which cannot be referenced from other levels of abstraction.

Levels of abstraction, which will constitute the partitions of the system, subsystem or procedure, are accompanied by rules governing the interrelations between them. There are two important rules governing levels of abstraction. The first concerns resources: each level has resources which it owns exclusively and which other levels are not permitted to access. The second involves the hierarchy: lower levels are not aware of the existence of higher levels and therefore may not refer to them in any way. Higher levels may appeal to the external functions of lower levels to perform tasks and also appeal to them to obtain information contained in the resources of the lower levels (Liskov 1972).

The abstraction of a procedure begins at the level of specification and the details that clarify the abstraction are added at the implementation level (Parnas 1972).

In this respect, the lowest level of abstraction is composed of the GFM, a clearly defined set of spatial operators (selectors, processors, recognizers, transformers) and a DBMS. The building blocks of the SDM are then those based on ASPA.

Data and System Independence

A GIS must be data and system independent. Multiple functional mappings should be allowed between the GFM, SDM, and any external data transfer operator. Similarly, the levels of abstractions induced in the GISP should allow the GIS to perform identically on different computers with no user intervention when doing the functional mappings.

CONCLUSION

GIS technology has finally taken off. However, as users become more sophisticated and demanding, we begin to discover how good a GIS has been designed. The notions of continuity presented above are the most important issues that need to be covered for a successful design. In my experience, along with the internal algorithmic robustness and data base consistency and integrity, flexibility and user friendliness (magic words) are today the most relevant points to be considered from the outcome of the design.

We should avoid making the mistake made during the 70's where authors entrenched themselves about whether raster data structures were better than vector structures. None and both were the answer. As we uncover the usefulness of concepts such as objects (object data bases, object oriented software engineering), we must not lose track of the flexibility geographic information systems must have. GIS are multidisciplinary tools. Fixed schemas will hinder GIS usage.

A solution has been explored here, whereby a GIS based on the six continuity principles given is able to support a Generic Functional Model (basic primitives, tool kit) that can generate via an Adaptable Spatial Processing Architecture, a Specific Derived Model (features, objects).

ACKNOWLEDGEMENTS

This work was possible thanks to support from project 9845/190 of the Environmental Systems Research Institute. This paper is part of ESRI's ongoing research program on GIS technology.

REFERENCES

Aronson, Peter and Morehouse, Scott (1983), "The ARC/INFO Map Library: A Design for a Digital Geographic Data Base," in Proceedings of AUTO-CARTO VI, Ottawa, Canada.

Bertrand Meyer (1988), Object-Oriented Software Construction, Prentice Hall International.

Codd, E.F. (1970), "A Relational Model of Data for Large Shared Data Banks," CACM 13, No. 6.

Chrisman, Nicholas (1976), "Local vs. Global: the scope of memory required for geographic information process," Internal Report 76-14, Laboratory of Computer Graphics and Spatial Analysis, Harvard University.

Corbett, J.P. (1975), "Topological Principles in Cartography," Proc. AUTO/CART 2, Reston, VA.

Dutton, Geoffrey (1978), "Navigating in Odyssey," in Harvard Papers on GIS, Vol. 6, Harvard University.

Cox, B.J. (1986), Object Oriented Programming, An Evolutionary Approach, Addison-Wesley.

Dijkstra, E.W. (1969), Structured Programming, Prentice Hall.

Guevara, J. Armando (1983), A Framework of the Analysis of Geographic Information System procedures: The Polygon Overlay Problem, Computational Complexity and Polyline Intersection, Unpublished Ph.D. dissertation, Geographic Information Systems Laboratory, State University of New York at Buffalo.

Guevara, J. Armando (1981), "Cartographic Data Structures: Abstraction," Unpublished paper, Geographic Information Systems Laboratory, State University of New York at Buffalo.

Guptill, S.C. (1986), "A New Design for the U.S. Geological Survey's National Digital Cartographic Data Base", Proceedings, Auto/Carto London, Vol. 2, 10-18.

Jensen, R. and Tonies, C. (1979), Software Engineering, Prentice Hall.

Kjerne, D. (1986), "Modeling Location for Cadastral Maps Using an Object-Oriented Computer Language," Proceedings, URISA, Vol. 1, 174-189.

Liskov, B. (1972), "A Design Methodology for Reliable Software Systems," Proceedings, Fall Joint IEEE Computer Conference.

Morton, G. (1966), "A Computer-Oriented Geodetic Data Base, and a New Technique in File Sequencing", report prepared for IBM Canada Ltd., Toronto.

Parnas, D.L. (1972), "A Technique for Software Module Specification with Examples," Comm. ACM, Vol. 15, No. 5.

Shamos, M.I. and Hoey Dan (1976), "Geometric Intersection Problems." 17th Annual Symposium on Foundations of Computer Science, pp. 208-215.

White, D. (1978), "A Design for polygon overlay", in Harvard Papers On Geographic Information Systems, Vol. 6, Harvard University.