Tsou, M.H. and Buttenfield, B.P.; 1997 A Direct Manipulation Interface for Geographical Information Processing. In: Kraak, M.J. and Melenaar, M. (eds.) *Advances in GIS Research II*: 905-915. First appeared in *Proceedings of the 7th International Symposium on Spatial Data Handling*, Delft, The Netherlands, August 1996: 13B.37 - 13B.47.

A Direct Manipulation Interface for Geographical Information Processing

Ming-Hsiang Tsou and Barbara P. Buttenfield
Department of Geography, University of Colorado, Campus Box 260
Boulder, Colorado 80309-0260, USA
email: tsou@ucsu.colorado.edu, babs@colorado.edu

ABSTRACT

Today the processing and analysis of geographical information is complicated by an increasing volume of information. We present a system to directly manipulate geographical data by using object-oriented approaches and graphic user interface (GUI) design. The study concentrates on vector data and overlay operations. A case study has been conducted by using this system for potential site selection in the Ellington, Connecticut area. The GUI design of the system uses icons to represent the geographical data and their operations. Object-oriented approaches are adopted in establishing a knowledge-based GIS system. This study suggests that the next generation of GIS user interfaces should provide an intelligent agent to assist users to search, query and operate on data. Keywords: direct manipulation, object-oriented approach, graphical user interface.

INTRODUCTION

With the rapid growth and development of computer technology and data gathering techniques, GIS have become more powerful, tremendous and complicated. A dramatic increase in the volume of geographical information challenges efficient management and analysis of this information. Processing and analysis of spatial data will be more complicated and difficult in the future. GIS operations have become incomprehensible for some users. However, many GIS procedures need to be modified and parameterized by their users, to cope with the complexity of the real world. We need a more intuitive way to help users work with GIS operators and data.

Recently, the graphic user interface (GUI) has emerged throughout GIS software. Case studies show that GUI users complete over one third more work per unit time compared with users of command-line interfaces. (Graham, 1994, p.32) Our approach adopts heuristic methods to improve the interaction between users and systems. However, most GIS software limits GUI commands to display of geographical information rather than on processing. As yet, there is no appropriate GUI designed for the purpose of processing or analysis of geographical data. The reason is that the nature of geographical phenomena is complicated, and the analysis of geographical phenomena is difficult to formalize into fixed functions. Introducing object-oriented approaches for spatial and geographical analysis operations may provide a natural and heuristic design at the conceptual level (Milne, Milton, and Smith, 1993). Since the processing methods developed by object-oriented approaches can provide reusable and extensible models, future developers of GIS can improve models developed previously by other geographers or scientists. Thus, the processing of geographical information will be more sophisticated and powerful.

Object-oriented concepts originate from object-oriented languages. The original object language was "Simula" developed in Norway in the 1960s (Taylor, 1992). The language was designed to simulate real world interactions and resolve complex problems. Object-oriented methods have been applied in a variety of fields such as business modeling, for example to establish the structure of an organization and the information flows within (Graham, 1994, p.318). In an object-oriented approach, data and operations are combined. Damage of important data due to unintentional changes by programming mistakes can be avoided. Spatial data need frequent updates and transformation to many different formats. One might say they need to be protected from external intrusion. Another advantage of an object-oriented approach is the reuse of object components. Object-oriented methods encourage people to retain established application models and improve upon them. Thus, geographical knowledge can be accumulated and multiplied.

Early studies of object-oriented methods in GIS applied object-oriented programming to specific applications, such as "object-oriented locational analysis" (Armstrong, 1989). Many geographers advocated the use of object-oriented approaches for spatial data management (Egenhofer and Frank, 1988) and data modeling (Worboys *et al* 1990). In cartographic research, object orientation has been applied to automate "... knowledge-based symbol selection ... for the visualization of univariate spatial statistical information." (Zhan and Buttenfield, 1995, p.293)

This project adopts a direct manipulation approach to implement iconic processing and analysis. We apply object-orientation to represent spatial data and GIS operations. According to past experience, we limit the project scope for the successful development of a knowledge-based system (Zhan and Buttenfield, 1995). This research uses a small testbed located in Ellington, Connecticut, and provided with ARC/INFO. We limit the scope of GIS operations to vector data and overlay analysis. This system has been implemented for a case study to locate a residential site. Although the object-oriented geographical processing is not comprehensive in this case study, the complete definition of various spatial relations and GIS operations is the long-term goal.

SYSTEM DESIGN AND IMPLEMENTATION

Vector Data Object

In order to transform geographical data to objects, we need to analyze their data structures. Every geographical data object has a specific geometric type such as polygon, line or point and a specific identifying name. For example, public wells are an instance of the "point data" class. Roads are an instance of the "line data" class. And landuse is an instance of the "polygon data" class. In addition, the complete object also needs geometric and attribute information. Data objects will be represented as icons indicating their data type (Figure 1).

The class of "vector objects" is defined as the composite of type, name, geometric data, attribute data and available operations. There are three subclasses of "vector objects" -- "point objects", "line objects", and "polygon objects". These subclasses inherit a data structure from their superclass "vector objects" and add new operations. The differences are illustrated in the boxes labeled "type" and "available operations". Different types of data require different operations. For example, a "line object" may have a specific "networking analysis" function in its available operations which "point object" and "polygon object" do not have. The advantage of the definition is to generalize the set of all possible GIS functions from hundreds of commands into organized sets of functions appropriate to particular types of data. Object encapsulation allows us to focus on the specific GIS operations associated with a specific type of data without concern for other non-relevant operations. Each data object will have an associated icon to distinguish it in the interface.

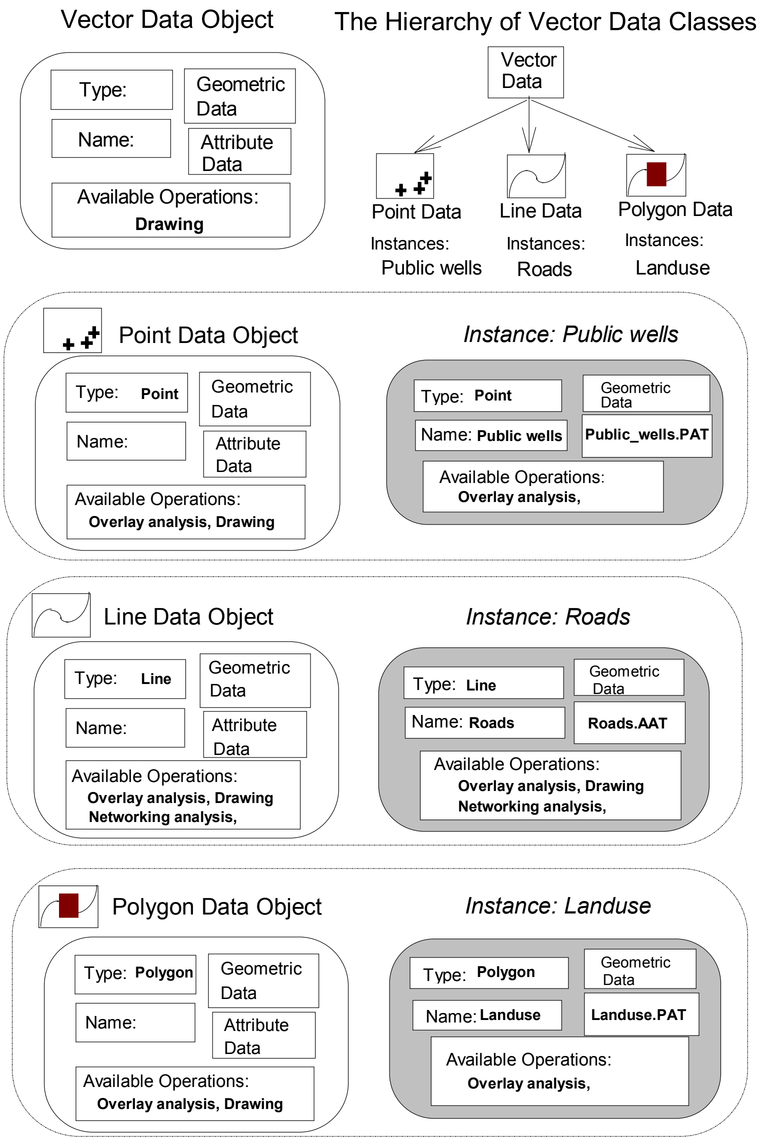

Figure 1. Vector Data Objects

Overlay Operation Objects

It is possible to represent operations as objects. To convert a GIS operation to an object format is more difficult than a data-to-object conversion because an operation is not an item in the same sense as a data item. Operations are composed of algorithms and procedures. In this research, every overlay operation will be represented as an object which consists of specific algorithms, procedures, and associated operation parameters. The elements of an operation object include a function name, the algorithm code and parameters. The object data structure will be filled at execution time by pointers to input and output data objects. Eight operations will be included in this case study. Each has an icon whose image indicates graphically the meaning of the operation. The operations are executed by direct manipulation, by selecting the icon identifying the object.

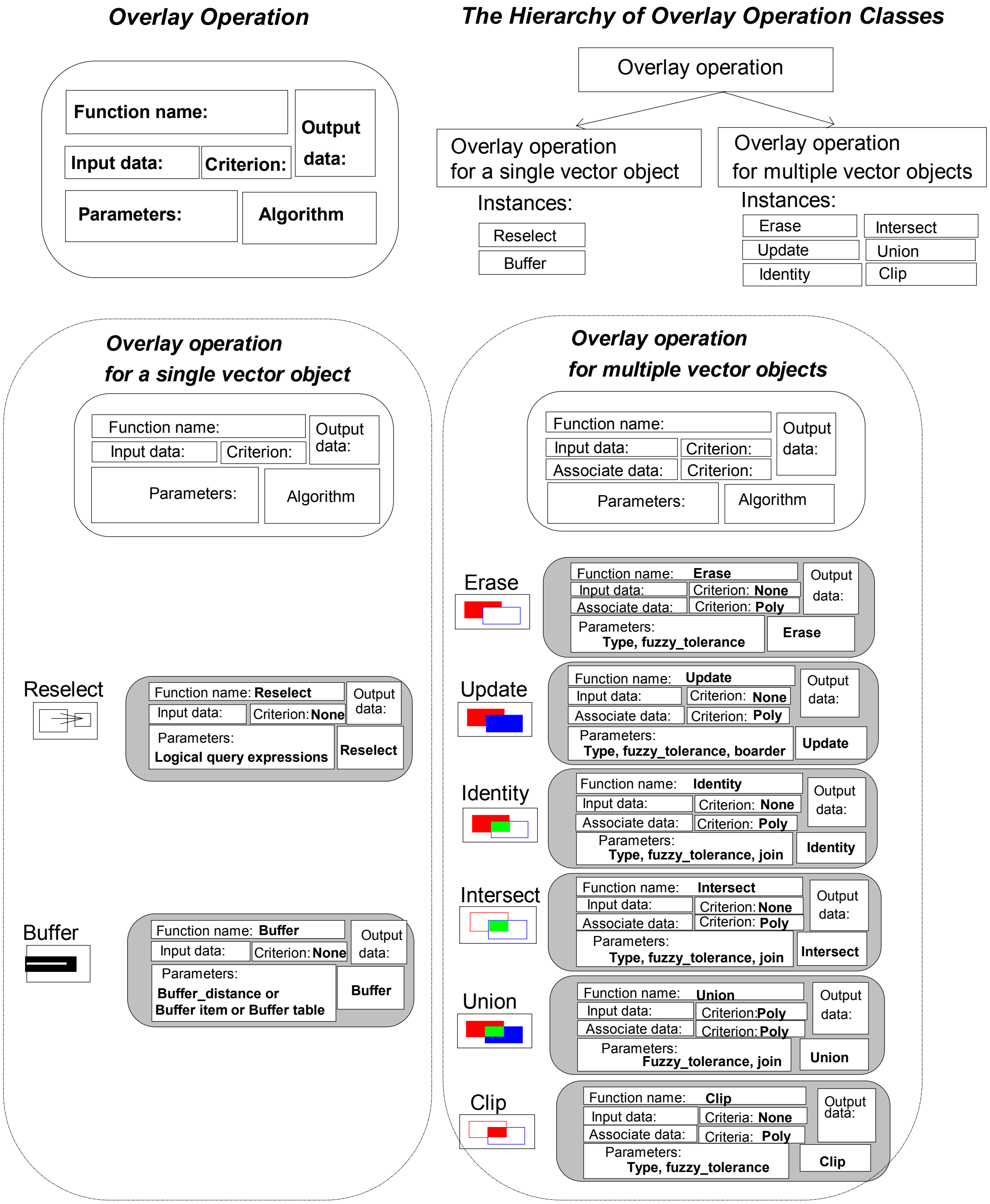

Figure 2. Overlay Operation Objects

In Figure 2 "overlay operations" form a class defined as the composite of "function name", "algorithm", "input data and criteria", "output data", and "parameters". Overlay operations" can be divided into two subclasses, "overlay operations for a single vector object" and "overlay operations for multiple vector objects". "Reselect" and "buffer" operations are instances of the first class as these operations are performed on a single data object. The second subclass adds "associate data" into its data structure because this class needs to process two data objects at the same time. Its instances include "erase", "update", "identity", "intersect", "union" and "clip" operations. This Figure also illustrates the hierarchy of object operations on single and multiple vector objects.

The Display of Data Flow

Major data processing procedures include data input, processing and output. In the interface, icons representing data and operations link between icons represent the data flow (Figure 3). Notice that the geometry contained in the data object (point, line or area) is integrated into the icon image, just as operations icons show simplified representations of the type of encapsulated overlay operator. Two types of data flow identify operators with single and multiple data inputs (data entry paths). Direct manipulation allows users to select data and perform operations by combining data icons with icons of operation objects. Arrows indicate the flow of data from one object to another. Since objects represent both files and operations, the data flow paths indicate data input from a file, processing by a particular operation, and output to another file. Data objects are encapsulated with specific operations, and the interface 'assists' the user by avoiding unlisted (inappropriate) operations. Visualizing the data flow can help users systematically organize their data and develop efficient data flows during a GIS analysis.

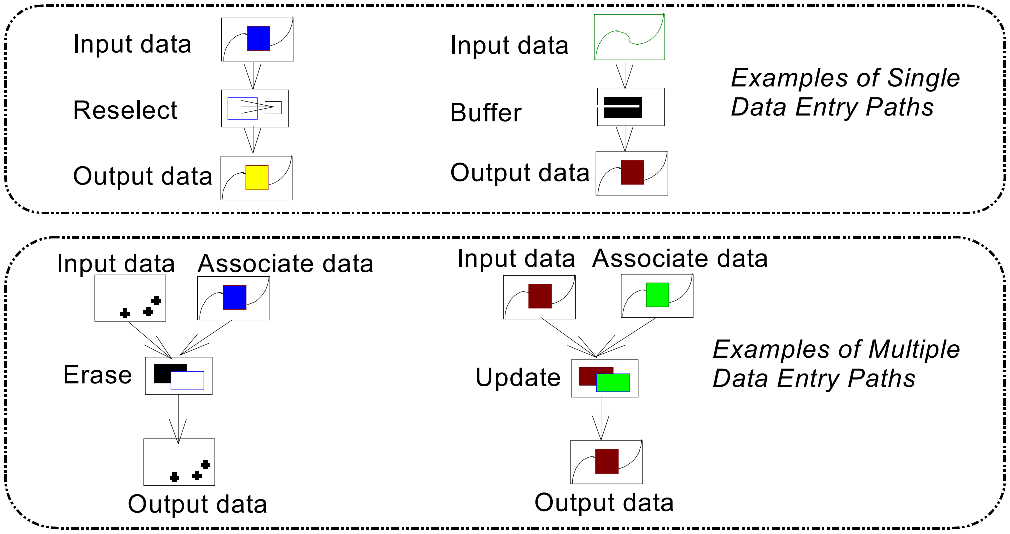

Figure 3. Data Flow Examples

Graphic User Interface Design

The direct manipulation design provides an efficient interface for user-system communication. The icons in this system must be simple, easy to operate and to recognize. Following established principles, the GUI includes a control panel and a display window. A GUI control panel has been designed by using the "form menu" provided in ARC/INFO's AML library (Figure 4).

Control panel functions must meet the requirements of system flow control. These functions include selecting and deselecting data objects and submitting operation requests such as "overlay" or "drawing". A data import function lists the available data objects (files) following an external database directory path. If users select an item from the list (a data object), its data file will be automatically imported to the system and appear as an icon in the display window. Four buttons ("select data", "deselect data", "overlay", "drawing" and "data import list") are implemented in the interface. In addition, two system buttons are included, to "restart" and "exit".

The display window consists of a processing box, an operator's box and a map display box. The processing box displays the data flow described above, and provides a visualization of the GIS analysis. The operator's box displays the icons of overlay operations, highlighting those that are encapsulated in data objects whose icons are selected. To execute the overlay, users select an operator and associated parameters. Its icon appears in the processing box; the operation is launched.

The map display box displays the geographical contents of selected data objects (data files) when the "drawing" request is given. The map display of geographical data can help users see the data features in operation results. The display window uses an ARCPLOT canvas. Comprehensive graphic functions in ARCPLOT display icons for data objects, operations and the map features.

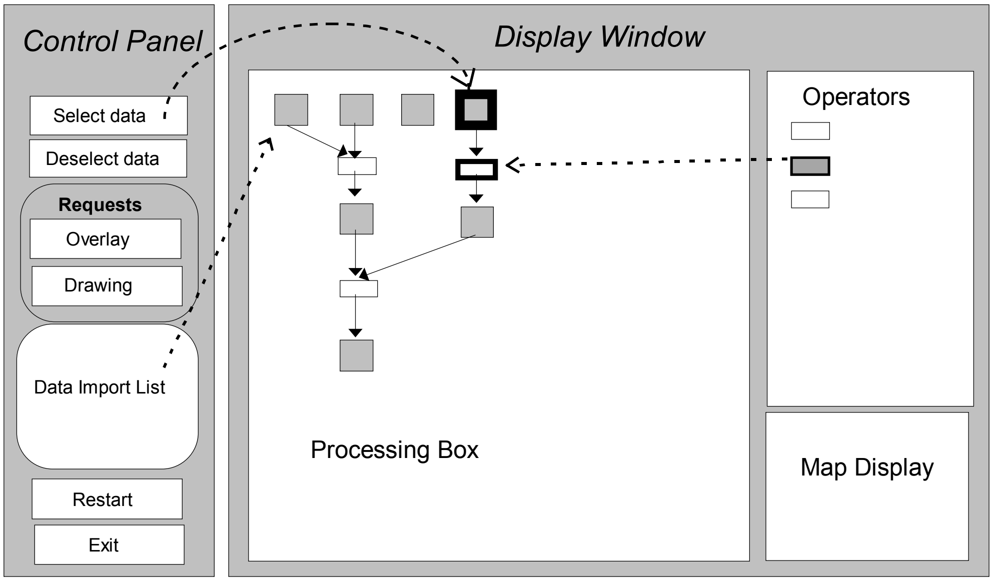

Figure 4. The Deployment of the Graphic User Interface

The deployment of the interface in Figure 4 illustrates how the control panel and display window appear in a typical situation. A user has selected four items from the data import list (four gray icons at the top of the display window). The user has clicked on the "select data" button in the control panel to select one of the four data object icons (gray-filled icon highlighted in black). The user clicked on the "overlay" button to select one of the related operators in the operator's box, and an operation icon appears in the display window (white-filled box highlighted in black). The operation generates a new data object whose icon appears in the processing box (just below the two highlighted icons). This is a single data entry operation. In a previous task, the user has performed a multiple data entry operation, shown by the cluster of icons on the left side of the display window. The control of the system relies almost totally on mouse-driven rather than keyboard input. Using the direct manipulation interface, users can learn and operate the system more easily.

A CASE STUDY

The case study applies the direct manipulation interface to locate a potential housing site. The testbed is "Ellington area". The Ellington database has comprehensive vector geographical data. The criteria for “locating a potential housing site” are listed in Table 1.

Table 1. The Criteria for Locating a Potential Housing Site

1. The site must be more than 100 meters away from wetland areas.
2. The site must be more than 200 meters away from steel tower power lines.
3. The site must be above the 500 year flood plain.
4. The site must be within 200 meters of a major road.
5. The slope of the site must be less than ten percent.
6. The landuse of the site must be "residential urban area".

Because the potential site must be distant from any of the first three criteria, these three areas can be spatially identified by a "union" operation combining three different data files. The other criteria can be met by a series of "intersect" operations. After that, the areas combined from the "union" operation should be excluded from the areas defined by the "intersect" operation. The "erase" operation accomplishes this. Figure 5 presents the GIS procedure in this case study. Users can visualize the flow of data processing and identify the interaction of operations and data from their respective icons. In effect, the Figure illustrates the chronology of processing steps to create the final output file. It is a graphical depiction of lineage. To our knowledge, this is the first iconic representation of lineage whose icons are also direct manipulation tools embedding operators that perform the analysis. In theory, one should be able to exchange this illustration along with the object encapsulations to allow another user to replicate the overlay process identically, including not only the sequence of operators but the parameters associated with each operator as well.

DISCUSSION

A comprehensive GIS software system usually includes hundreds of commands and functions. Users need to identify which commands are appropriate for their specific data and application. The interface proposed here uses direct manipulation to simplify operations, and provide a visual record of a complex GIS task. If users focus on a specific data object, the system can minimize hundreds of available commands to a smaller number of appropriate operations. In addition, the system can check the parameters of the relevant operations. Users can concentrate on their data and application instead of getting sidetracked in finding the right commands.

The interface developed in this research uses object-oriented approaches to formalize the logical expression: *[Select data objects] ---> [Display the available operations] ---> [Select the* *operation] -->[Execute the operation]*. The iconic approach is more logical than a command-line interface because people can visualize the geographical phenomena “first” and then consider appropriate operations. Therefore, users can execute GIS operations more naturally and intuitively.

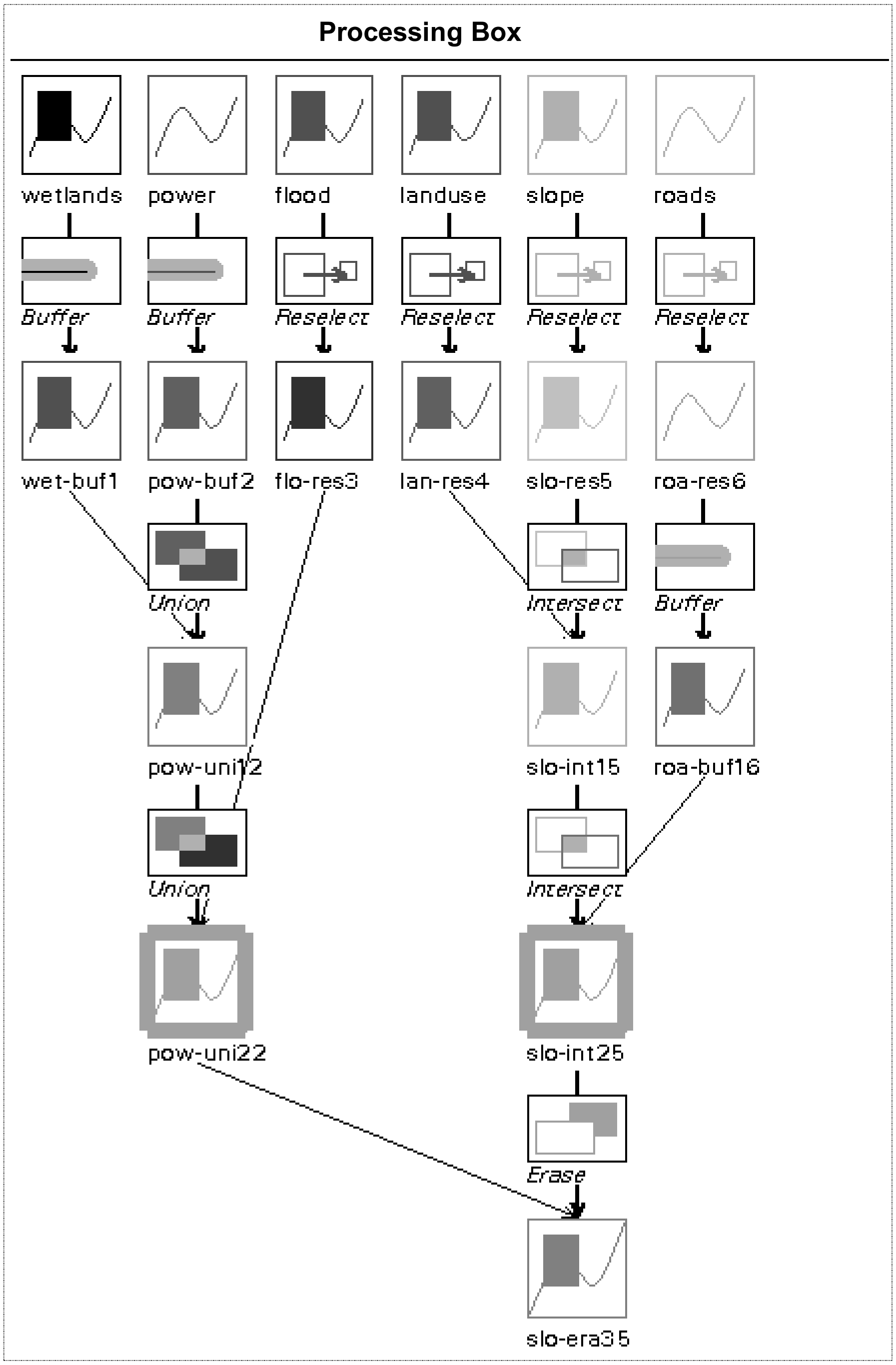

Figure 5. Data Flow of the Case

It is hoped that iconic direct manipulation interface design will improve data sharing and communication among geographic researchers and scientists. By saving an image of the icons after a GIS task is complete, researchers can share their models or procedures with other people easily. Exchange of an iconic representation of the model or procedure should not require transfer of source code, AMLs or other programs, but only exchange of the object encapsulations. Recipient users can replicate, revise or tailor the model to meet their specific application.

SUMMARY

Recently the GIS community has begun to pay much attention to human-computer interaction (HCI) in the development of GIS user interfaces (Medyckyj-Scott and Hearnshaw, 1993). Vendors have begun to develop GUI to improve the competitiveness of their products. For example, ESRI's ARC/INFO system introduced ARCTOOLs, a menu-driven interface as an easy-and-friendly user interface. MGE, Intergraph's Modular GIS Environment, uses the menus and icon-based interface of Windows. ERDAS' Imagine adopts a graphical user interface. Another example of GIS software which concentrates on the visualization of information processing and analysis is Geolineus, which is developed by GeoDesigns, Inc. (Lanter, 1994). This study extends Lanter's tools, implementing direct manipulation into the icons. All of these systems are intended to make GIS operations easy and self-explanatory for novice users and non-programmers.

Limitations of this study can be discussed from several perspectives. First, the system is specific to ARC/INFO. The encapsulated objects are not directly transferable to other GIS software. The programming capability in AML constrains object definitions and their hierarchy. To get around this, we used a series of global variables to represent the meanings of objects and their relationships. The GUI design is also constrained by the capabilities of ARCPLOT. For example, in a fully direct manipulation environment, functions such as "click and drag" would be implemented. At present, these functions can not be performed in ARCPLOT. Second, from a cognitive perspective, the iconic language has not been empirically evaluated, and may be insufficient for users to understand GIS operators and ancillary information. The authors have begun to design experiments to resolve this issue. Also the classification and hierarchic structure of GIS operations is still preliminary. Third, this study only focuses on vector data objects and overlay operations. Raster data objects and other GIS analysis operations such as networking analysis, hydrological analysis and 3D modeling provide opportunities for further research.

In the past, the development of GIS lacked the application of geographic models that were complicated and diverse. It was difficult to implement geographic models by procedural programming languages. In the future, there are many possible directions for the development of user interface design in GIS. However, the development of a GIS user interface still relies highly on the progress of hardware and software technologies. "GIS represents the interface between geography and an external technology, because developments in computers have been the key enabling factor that has made GIS possible" (Taylor and Johnson, 1995, p.51). The rapid progress of computer science will always have a significant influence on the development of GIS.

Direct manipulation programming capability may be available in the next generation of GIS software. Users should be able to implement their geographical models and spatial theories through object-oriented approaches. The next generation of GIS should provide user interfaces as intelligent agents to assist users to search, query, and operate the system. Interface design should consider comprehensive perspectives, including the system operation level and users' recognition and visualization levels. Future research on GIS system design should focus on the following three areas-- establishing spatial analysis theories and geographical models in GIS by using object-oriented approaches, deeply exploring the meaning of visual language and the cognition of users, and designing a comprehensive and intelligent user interface. This study gives a proof-of-concept for a direct manipulation interface and explores relevant research issues.

ACKNOWLEDGMENTS

This paper represents part of NCGIA Research Initiative 8, "Formalizing Cartographic Knowledge", at the National Center for Geographic Information and Analysis, supported by a grant from the U.S. National Science Foundation (SBR-88-10917). Support by NSF and by the University of Colorado is gratefully acknowledged.

REFERENCES

Armstrong, M. P., Densham, P. J. and Bennett, D.A.,1989, Object oriented locational analysis, *Proceedings GIS/LIS '89*, Orlando, Florida: 717-726.

Egenhofer, M. J. and Frank, A. U. 1988a. Object-oriented database: database requirements for GIS. in *Proceedings of the International Geographical Information Systems Symposium: The Research Agenda*, Vol. II: 189-211.

Graham, I., 1994, *Object-oriented Methods.* Workingham, England: Addison-Wesley Publishing Company (2nd Edition).

Lanter, D. P., 1994, A Lineage Metadata Approach to Removing Redundancy and Propagating Updates in a GIS Database, *Cartography and Geographic Information Systems*, Vol. 21(2): 91-98.

Medyckyj-Scott, and Hearnshaw, H. M. Editors., 1993, *Human Factors in Geographical Information Systems*. London: Belhaven Press.

Milne, P. , Milton, S. and Smith, J. L., 1993, Geographical object-oriented databases -- a case study, *International Journal of Geographical Information Systems*, Vol. 7(1): 39-55.

Taylor, D. A., 1992, *Object-Oriented Information Systems: Planning and Implementation*. New York: John Wiley & Sons, Inc.

Taylor, P. J. and Johnson, R. J., 1995, GIS and Geography, In: Pickles, John (Ed.) *Ground Truth: The Social Implications of Geographic Information Systems*, New York: Guilford. Ch. 3: 51-67.

Worboys, M. F., Hearnshaw, H. M., and Maguire, D. J., 1990, Object-oriented data modeling for spatial database. *International Journal of Geographical Information Systems*, Vol. 4(4): 369-383.

Zhan, F. B. and Buttenfield, B. P., 1995, Knowledge-based Symbol Selection for Statistical Information, *International Journal of Geographical Information Systems*, Vol. 9(3): 293-315.