1998 * Tsou, M. and Buttenfield, B.P. An Agent-based, Global User Interface for Distributed Geographic Information Services. **Proceedings** 7th International Symposium on Spatial Data Handling, Vancouver, British Columbia, July, 1998: 603-612.

An Agent-based, Global User Interface for Distributed Geographic Information Services

Ming-Hsiang Tsou and Barbara P. Buttenfield
Department of Geography, University of Colorado, Boulder, CO 80309 USA
E-mail: tsou@ucsu.colorado.edu, babs@colorado.edu Phone: +1-303-492-3684

Abstract

This paper establishes an architecture for global user interfaces by defining appropriate user tasks and designing an agent-based interface mechanism for distributed geographic information services. User tasks include query, display, data integration, and GIS processing functions. The use of intelligent agents is justified in the context of increasing geospatial data volume, increasing complexity of GIS modeling, and the diversity of GIS software resources available on local and distributed networks. The architecture is defined within a context of distributed component technology. Internet-based GIS use can be facilitated by a global user interface, which is platform-independent, modularized, self-describing, and able to access multiple, heterogeneous, geographic information services distributed on the Internet.

Keywords: user interface, intelligent agent, distributed component technology

1. Introduction

The popular use of the Internet and the dramatic progress of network technology have altered the development of Geographic Information Systems (GIS). More and more GIS applications focus on the capability of "mapping on the Web". To deliver distributed geographic information in an Internet-based, distributed network environment, a new paradigm for GIS architecture needs to be established and adopted. The architecture would operate as a collective of modules, each drawn from a potentially different GIS vendor (across the Internet). The user interface for this architecture would filter requests for GIS information and operations to select the most appropriate module. The interface would also access and convey results of each module's processing in sequence. This requires a radical re-thinking of the user interface (Figure 1).

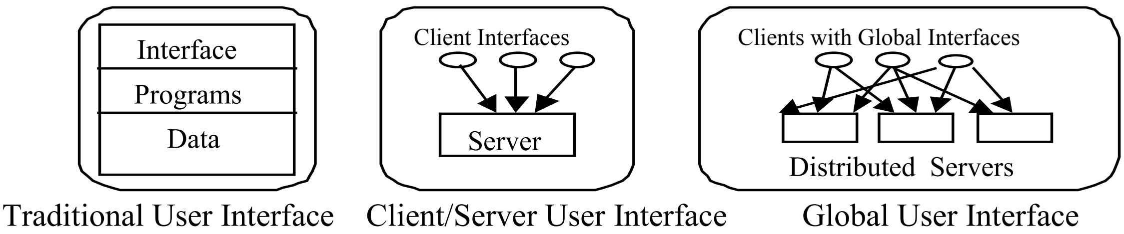

Figure 1. Three paradigms of user interface design

- *Traditional user interfaces* are widely used in current GIS applications and operate within a closed, centralized environment. The user interface is attached to an application from which it cannot be separated. Each user interface is platform-dependent and application-dependent.

- *Client/Server user interfaces* are adopted in generic client/server systems. The interfaces are separated from data and programs to allow distributed clients remote access to the server. The client-side user interfaces are usually platform-independent. However, each client can only access one specified server. Different geographic information servers come with different client-side interfaces, which can not be shared.
- *Global user interfaces* are also platform-independent, but are more advanced than generic client/server interfaces. The most significant difference is that global user interfaces are not constrained for use on a single server. The interfaces can access multiple servers and heterogeneous systems, without the constraints of traditional client/server relationships.

Currently the most well-known examples of global user interfaces are the browsers of the World Wide Web. A Web browser can access different information servers on the Internet. Techniques such as JAVA applets and ActiveX controls have been developed to provide a fully platform-independent interfaces for Internet applications. ***Distributed component technologies*** such as the Common Object Request Broker Architecture (CORBA), the Distributed Component Object Model (DCOM), and JavaBean, allow clients to access heterogeneous servers(Orfali and Harkey, 1997). Distributed component technology provides the essential feature underlying global user interfaces. Distributed data objects and applications can freely interact and inter-operate on the Internet. The trend is to provide a comprehensively open distributed computing environment, where the "network" is the "computer".

In the GIS community at present, many popular GIS on-line services access the Web via HTML format and CGI programs. Examples of current products include Xerox Map Server (Putz, 1994) and GRASSLinks (Huse, 1995). Recent research projects such as the Alexandria Digital Library Project (Buttenfield and Goodchild, 1996) have moved from HTML to JAVA technology to explore and compare comprehensive services for on-line spatial queries and metadata browsing. On the other hand, distributed component technology has not been widely adopted for on-line GIS services, in spite of many proposals focusing on this issue, such as OpenGIS (Buehler and Lance, 1996), ISO/TC211 (Ostensen, 1995), and component-oriented GIS (Li and Zhang, 1997).

The GIS community needs to adopt these new computer technologies, to broaden the computing framework for distributed geographic information services. We seem to be entering the "aviation" era from the "automobile" era. For the user, the interface is the mechanism to guide the vehicle. We need to develop a new type of interface, a "control stick" for airplanes instead of a "steering wheel". A global user interface will access data distributed in multiple clearinghouses, and access heterogeneous processing modules that may be distributed by multiple vendors or brokers. This paper establishes an architecture for global user interfaces. We identify a minimum set of four user tasks, define mechanisms by which the interface conveys information and thus accomplishes the tasks, and define architecture components for each task. At the end of the paper, we discuss how the global user interface might operate.

2. User Tasks and Interface Mechanisms

The design of a user interface involves interdisciplinary collaboration between computer science, experimental psychology, graphic design, and information science (Shneiderman, 1998). From the cognitive perspective, user interface design should be driven by two fundamental elements-- ***user tasks*** and ***interface mechanisms***. Task-centered design identifies real, complete, and representative user tasks through various methods (scenario-building, cognitive walk-throughs, e.g.), and subject-tests these to validate interface design in actual use (Lewis and Rieman, 1993). Interface mechanisms bridge the communication between users and systems. Examples of interface mechanisms include menus, command-lines, direct manipulation icons, graphical metaphors and virtual reality (VR). The appropriateness of any interface mechanism depends on the set of user tasks and plays an essential role for the success of user interface design.

Some GIS research has addressed these issues, identifying general tasks for GIS (Davies, 1996), tasks involving data manipulation (Gould, 1993), and GIS operators formalized in map algebra (Tomlin, 1990; Kirby and Pazner, 1990). However, these studies focused on traditional GIS architecture, which may not be suitable for distributed network environments.

2.1 Tasks for distributed geographic information services

According to several researchers (Plewe,1997; Li and Zhang, 1997), the set of representative user tasks for distributed geographic information services should include:

- Spatial and text-based query;
- Map display;
- Data download, pre-processing and integration with local databases; and
- On-line GIS processing and spatial analysis.

These tasks are identical to traditional GIS tasks, except for being performed in distributed environments. Two reasons for distributing geographic information services come readily to mind. First, increasing size of geospatial data sets impedes GIS task completion. Data are time-consuming to download, and processing large data sets may not always be possible on smaller workstations. One can imagine a distributed processing arrangement whereby one could send encapsulated GIS processing objects to a large clearinghouse. Data would be processed at the server, and results encapsulated within the processing object to be returned to the client. Mechanisms for this type of processing form a related research interest of the authors (Tsou and Buttenfield, 1996; Buttenfield and Tsou, 1997).

A second advantage of distributed task processing relates to package specialization. Most GIS software has acuity for specific processing tasks. Some but not all packages can handle differential segmentation (breaking up linear features on the basis of a particular attribute); others are adept at merging field data with entity data; still others provide excellent address matching as a primary function. The complexity of modeling tasks undertaken by most GIS analysts demands a working knowledge of several GIS packages. In a truly distributed geographic processing environment, a global user interface would find the most appropriate GIS module available on the distributed network, and identify the most data source most appropriate to the user's articulated task. The idea is not so far-fetched as skeptics might argue. It is nearly within the realm of existing methods of object-orientation and distributed networking.

2.2 Underlying mechanisms of a global user interface

Two specific problems must be addressed to implement this strategy. The first problem is how to access possibly heterogeneous systems. The second problem is how to the formalization of interoperable geodata objects and GIS operators, which incorporate comprehensive metadata descriptions. This paper utilizes an agent-based interface mechanism to help users distribute GIS tasks, identify geodata and operators, and easily access heterogeneous systems.

Recently, intelligent agents have become a new direction in both user interface design and artificial intelligence (AI) research. The major goal of intelligent agents is to reduce user work and information overload (Maes, 1994). Intelligent agents can provide services in filtering data, searching for information, online tutoring, and so on. By utilizing AI inference engines and learning process components, intelligent agents gather and formalize knowledge from communications between users and systems in network environments. The learning processes are derived from user feedback and the evaluation of previous performance. Learning reinforces the agent's capabilities towards more rational behaviors (Russell and Norvig, 1995). Many intelligent agent applications are under development now, especially in distributed computing environments (Knapik and Johnson, 1998). Two fundamental functions of an intelligent agent are essential to distributed geographic information environments. An intelligent interface agent should be able to perform both functions.

The first role, as an ***information filter***, helps users limit their choices to a reasonable scope according to pre-defined knowledge rules and user's advice and feedback (Figure 2). For example, pre-defined knowledge about the amount of client RAM could limit the choice of source data resolution. The interaction between intelligent agent and user is metaphorically similar to the relationship between car dealers and car buyers. When a buyer tells the car dealer what kind of car they prefer and what price they can afford (task specifications), the car dealer filters information and generates a reasonable list of choices. One expects that with

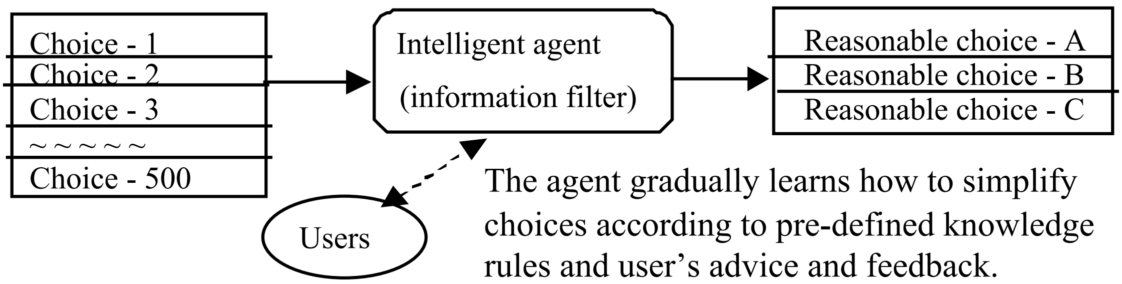

Figure 2. The information filter function of an intelligent agent

time, the interface agent will gradually become more effective in filtering information, and generate more rational choices. The role of information filter is essential to distributed geographic information services which hypothetically could involve a multitude of different GIS operations and data types. Moreover, "experienced" intelligent agents might suggest that novice users modify or revise their requests in order to get a "better" result set of choices.

The second role of an intelligent agent is to act as an ***information interpreter***. Heterogeneous data models and systems can not communicate directly. An information interpreter must be designed to access and convey information. In order to convey information correctly, the interpreter has to acquire some knowledge and methods to guide the translation procedure. It would be unfeasible to encode a set of translators into the interpreter, as one must anticipate that no list can be comprehensive in the longterm. An alternative solution is to design data- and operator-objects which are self-describing, that is, which contain metadata describing their structure, input and output parameters, and computing requirements in an open message-passing format. The encapsulation provides metadata information for the intelligent agent to interpret the heterogeneous information correctly. Figure 3 shows one example of an

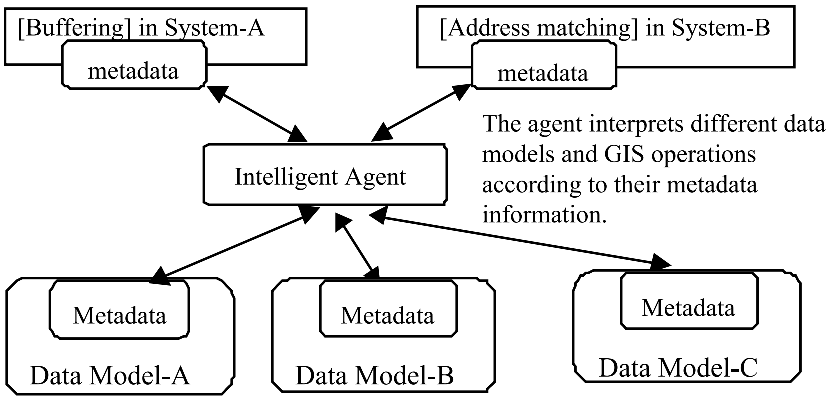

Figure 3. An information interpreter function of an intelligent agent.

information interpreter function. The agent collects metadata from several data objects and from several operator objects, to determine an effective combination. The advantage is that the agent does not carry knowledge with it except for options about metadata access.

3. A Global User Interface Architecture

A comprehensive global interface architecture for distributed network environments should include four task-oriented interface components at a minimum. These components query data, display data, integrate data locally, and process data.

3.1 Spatial and text-based query component

The query interface component helps users define and revise their requests. The query agent initiates searches for resources across the distributed network. The query metadata describes the main query functions, identifying (for example) spatial footprints, data types, and search keywords. The interface agent interprets the metadata specification and sends the requests to database agents, which can search and retrieve requested data from heterogeneous GIS databases (Figure 4a).

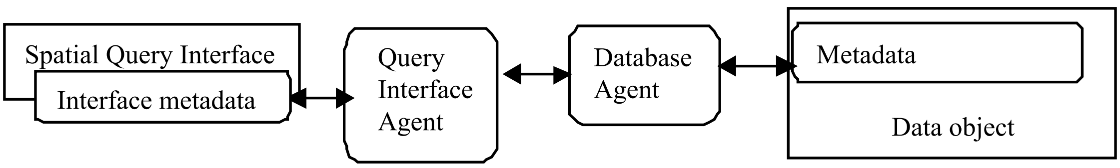

Figure 4a

3.2. Map display component

The map display component utilizes display metadata (including map boundaries, coordinate systems, and projection methods) for the interface agent to display correctly (Figure 4b).

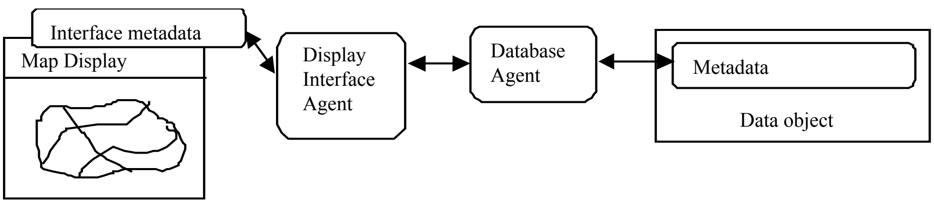

Figure 4b

It is reasonable to expect that display agents accommodate multiple output and display device drivers, and transform display commands accordingly. In the event that these parameters are undefined, the interface agent could generate a "zoom-to-full-extent" command, or initiate a dialog with the user about alternative display options (e.g., define a projection).

3.3 Data download, pre-processing and integration component

This component fuses geospatial data retrieved from remote databases with the local client database. The integration interface agent (Figure 4c) can access and transfer geodata between heterogeneous data models, and thus by definition must incorporate an information interpreter, as described in Section 2. In the figure, an Interface Integration Agent receives a request to retrieve data (top center). The request is passed to a database agent. At the remote database, a temporary buffer is created to pre-process data (for example, to clip a portion of a digital

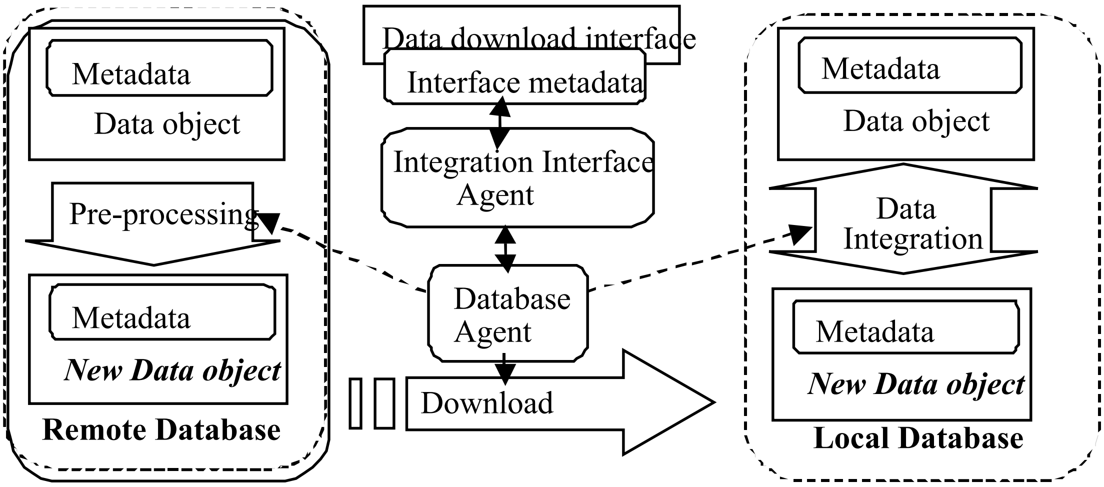

Figure 4c

orthophoto) and create a metadata description. Following data download, the metadata for the new object is accessed to direct data integration into the local database. This might involve creating another data object, as for example if the orthophoto were mosaicked with other retrieved data. The interface agent lets users monitor the download process and data integration procedures.

3.4 GIS processing and spatial analysis component

The processing component is used for analysis and modeling (Figure 4d). Based on the user's data type, the processing agents provide appropriate processing operations, transparently. For example, buffering operators would be available for point line and polygon entity data, whereas field data would have a "region growing" operator. A model for change detection would be made available for data objects which have a timestamp encapsulated in their metadata. Interface mechanisms to implement such context-dependent functionality are demonstrated in Tsou and Buttenfield (1996).

In general, the architecture of global use interface adopts an object-oriented modeling method, where each interface component encapsulates its own functions and capabilities. The interface agents and database agents are also objects, able to access and invoke the GIS

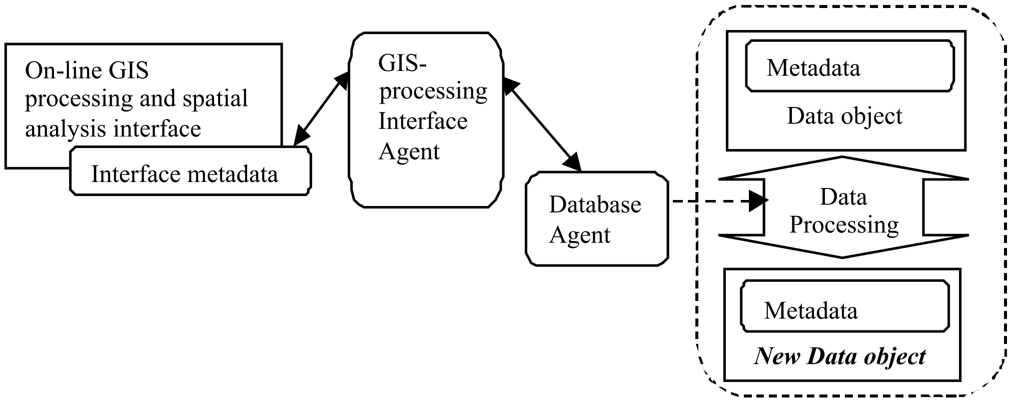

Figure 4d

operators, communicate with interface components and with objects stored in GIS databases. Thus, every global user interface component is modularized, self-describing, and interoperable.

4. Discussion

Without pre-defined user tasks, metadata and agents, a global user interface for distributed geographic information services would be difficult if not impossible to implement. At a minimum, user tasks include query, display, data integration, and processing. Each task has a component architecture associated with it. Each interface component must have encapsulated interface metadata, which describes available functions, methods, and behaviors for each interface component and GIS operators, and facilitate a modularized user interface infrastructure. An agent must perform two functions, namely to help users filter and interpret information in a complex distributed network environment. Two kinds of intelligent agents, interface agents and database agents, must be developed for each specified user task. The main functions of interface agents are to interpret metadata, and to collaborate with database agents. The interface agent accepts task specifications from users and translates them into system commands or query languages. Database agents are designed to access heterogeneous geodata objects and databases in distributed network environments. The database agent receives system commands and distributes requests to multiple databases. As the requests are accomplished, results will be gathered and filtered by database agents, then sent to the user directly or carried through the interface agent.

The collaboration between interface agents and database agents provides a comprehensive information process model for distributed geographic information services (Figure 5). The combination of intelligent agents and distributed components creates a self-describing, dynamic, reusable and interoperable interface architecture for global user interfaces.

From the implementation perspective, a global user interface can be built with intelligent agents and interface components by using currently available technology. JAVA is an appropriate language of global user interfaces because it is platform-independent, and provides comprehensive user interface capabilities in its Abstract Windowing Toolkits (AWT). The JAVA language can easily support a distributed infrastructure for intelligent agents and interface components. Metadata descriptions can be embedded within each interface component by the Interface Definition Language (IDL) provided by CORBA

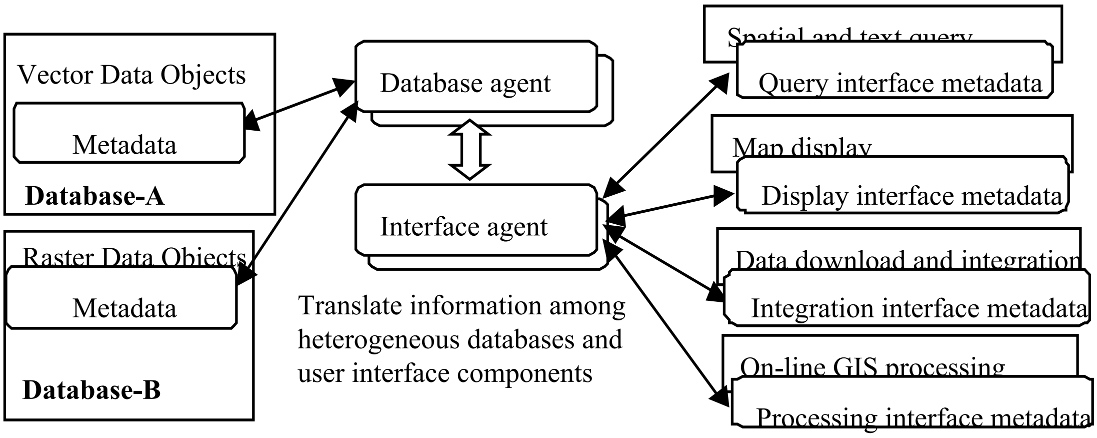

Figure 5. The collaboration of interface agents and database agents.

infrastructure. Interface agents will be built with the four task-oriented interface components. Some knowledge rules and learning mechanisms must be embedded inside interface agents in order to effectively manage and communicate with different interface components.

On the other hand, the implementation of interoperable geodata objects and metadata can be somewhat problematic where current technology does not fully support the implementation. Although the Open GIS Consortium (OGC) is launching the OpenGIS project and its feature specifications, few commercial tools are presently available for geodata objects. Exceptions to this can be seen in Map Objects by ESRI and GeoMedia by Intergraph. Other solutions are beginning to appear. Another constraint is that most current GIS databases rely upon relational databases, not object-oriented databases. One possible work-around solution is to "wrap" traditional legacy databases under a CORBA or DCOM server to communicate with database agents. The migration from relational databases to object-based databases will become a major task in implementing distributed geographic information services.

Another impediment for global user interfaces involves communication between interface agents and database agents. Current technology provides a simple, first level protocol called "Internet interoperable ORB protocol (IIOP)", supported by CORBA 2.0. However, IIOP only focuses on the “invoke” mechanism between agents and data objects. What is needed is to focus on interactions between agents (Knapik and Johnson, 1998). An efficient and concise protocol for the communication between agents needs to be developed. A generic protocol should be designed with rules about how specific interface agents and database agents behave. This capability is essentially a meta-protocol. Hopefully, prototypes of meta-protocols will be developed and tested in the next few years.

5. Summary

This paper establishes an architecture for global user interfaces by defining appropriate user tasks and designing an agent-based interface mechanism for distributed geographic information services. Some issues need to be explored in the future.

The first issue relates to the level or degree of user control. In our architecture, agents take some control away from the users. The debate between proponents of direct manipulation interfaces and intelligent agent-based interfaces is similar to the comparison of standard transmission cars and automatic transmission cars. In practice, the level of control should depend highly on the level of user skills, preferences and tasks. The problem also emerges with the use of "visible" versus "invisible" agents. Currently, design of most intelligent agents is visible, which means that the agent makes its presence known through dialog with the user. It is expected that some users feel that agent interruptions are intrusive. Invisible agents may solve this problem but introduce another, which is that users will be unable to monitor and control agent behaviors. These issues can be resolved through subject testing in free and structured interface use (Buttenfield and Larsen, 1996).

The second issue relates to the 'necessary evil' of distributed network environments. Consider the situation when a network link breaks down. Since a general spatial operation or analysis procedure takes a reasonably long time, unpredictable interruptions of the network could jeopardize the process of spatial operations. To protect the integrity of GIS operations and user sessions, disaster recover functions and session controls become essential in an unreliable networking environment. Feedback messages in GIS operations and analysis processes are also important, especially for indicating the current status of the network environment. Database developers are already working to overcome these problems.

In spite of these unsolved issues, the need for distributed processing, coupled with the complexity of current and anticipated GIS modeling tasks, mandates consideration of distributed architectures for networking environments. The utilization of intelligent agents and global user interface components will crystallize geographic information knowledge and provide a comprehensive approach for implementing distributed geographic information services. We have the technology in hand to implement such architectures. Let us begin.

Acknowledgments
This paper forms a portion of the Alexandria Digital Library Project, jointly sponsored by NSF, NASA, and DARPA. Funding by the National Science Foundation (NSF IRI-94-11330) is greatly appreciated. Matching funds from the University of Colorado are also acknowledged.

References
Buehler, K., & McKee, L., editors. (1996). *The OpenGIS Guide: Introduction to Interoperable Geoprocessing*. Wayland, Massachusetts: Open GIS Consortium, Inc.

Buttenfield, B.P. and Tsou, M. -H. (1997) Encapsulated Operators for Processing Geographic Information. Second Workshop on Progress in Automated Map Generalization, Gaevle, Sweden, 19-21 June 1997. Summarized in Mackaness, W.A., Weibel, R. and Buttenfield, B.P. (1997) Research Progress in Map Generalization. Santa Barbara, California: *NCGIA Technical Report 97-3.* Also available at http://www.geo.unizh.ch/ICA/.

Buttenfield, B. P., & Goodchild, M. F. (1996). The Alexandria Digital Library Project: Distributed Library Services for Spatially Referenced Data. In *Proceedings of GIS/LIS'96 , Denver, Colorado*: 76-84.

Buttenfield, B. P. and Larson, S. (1996) Building a Digital Library for Earth System Science: The Alexandria Project. *Proceedings* Geological Society of America Meetings, Denver Colorado, November 1996: 23-27.

Davies, C. (1995). Tasks and Task Descriptions for GIS. In T. L. Nyerges, D. M. Mark, R. Laurini, & M. J. Egenhofer (editors), *Cognitive Aspects of Human-Computer Interaction for Geographic Information Systems*. Dordrecht: Kluwer Academic Publishers: 327-342.

Gould, M. D. (1993). Chapter 8: Two Views of the User Interface. D. Medyckyj-Scott, & H. M. Hearnshaw (editors), *Human Factors in Geographical Information Systems*. London: Belhaven Press: 101-110.

Huse, S. M. (1995). *GRASSLinks: A New Model for Spatial Information Access in Environmental Planning*. Unpublished doctoral dissertation, University of California at Berkeley, Berkeley, California, USA.

Kirby, K. C., & Pazner, M. (1990). Graphic Map Algebra. In *Proceedings of the 4th International Symposium on Spatial Data Handling, Zurich, Switzerland*. IGU: 413-422.

Knapik, M., & Johnson, J. (1998). *Developing Intelligent Agents for Distributed Systems: Exploring Architecture, Technologies and Applications*. New York: McGraw-Hill.

Lewis, C., & Rieman, J. (1993). *Task-Centered User Interface Design*. ftp://ftp.cs.colorado.edu.

Li, B., & Zhang, L. (1997). A Model of Component-oriented GIS. In *Proceedings of GIS/LIS'97, Cincinnati, Ohio*: 523-528.

Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communications of the ACM,* 37(7): 31-40.

Orfali, R., & Harkey, D. (1997). *Client/Server Programming with Java and CORBA*. New York, New York: John Wiley & Sons .

Ostensen, O. (1995). Mapping the future of geomatics. *ISO Bulletin, December 1995. URL: http://www.statkart.no/isotc211/isobulen.html.*

Plewe, B. (1997). *GIS Online: Information Retrieval, Mapping, and the Internet*. Santa Fe, New Mexico: OnWord Press.

Putz, S. (1994). Interactive Information Services Using World Wide Web Hypertext. In *Proceedings of the First International Conference on the World-Wide Web, May 25-27, 1994, Geneva, Switzerland*: 273-280.

Russell, S. J. and Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, New Jersey: Prentice Hall.

Shneiderman, B. (1998). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (Third ed.). Reading, Massachusetts: Addison-Wesley.

Tomlin, C. D. (1990). *Geographic Information Systems and Cartographic Modeling*. Englewood Cliffs, New Jersey: Prentice Hall.

Tsou, M. -H., & Buttenfield, B. P. (1996). A Direct Manipulation Interface for Geographical Information Processing. In *Proceedings of the 7th International Symposium on Spatial Data Handling, Delft, The Netherlands*. IGU: 13B.37-13B.47.