Gis2web And Its Inter-Application Communications Issues

Xiannong Meng  Richard H. Fowler
Department of Computer Science
University of Texas - Pan American
Edinburg, TX 78539-2999
{meng,fowler}@panam.edu

Eric R. Rieken
Department of Physics and Geology
University of Texas - Pan American
Edinburg, TX 78539-2999
ericr@panam.edu

Abstract
Gis2web is a system intended to bridge the gap between GIS software and datasets and the WWW. The system operates by periodically accessing distributed datasets ofGIS information and creating a common repository of WWW accessible information on a gis2web server. In this way gis2web users are provided seamless access to different datasets which may be of quite different forms. This paper discusses the architecture and communications issues of gis2web.

1 Introduction
Geographic Information Systems (GIS) such as Arc/Info and GRASS [16] contain rich and diverse information such as census data, municipal data, business data, highway networks, and geologic data. Information is stored in formats including tables, images, and text. Typical GIS software maintains, manipulates, and displays this information in dynamic and graphical ways. The traditional mode of operation is to have the data accessible from within a specific GIS application on a single platform.

With the recent wide-spread acceptance and availability of World Wide Web (WWW) [3] technology, it is natural to look toward building linkages between the two distinct worlds of GIS and the WWW in order to make the wealth of information available in the GIS world publicly available through the WWW.

This paper describes the architecture and the communication aspects of gis2web, a system which facilitates communications and information exchange between existing GIS software, its data, and the WWW. The gis2web prototype accesses distributed datasets in different Arc/Info formats, converts the data to forms appropriate for WWW distribution, and provides WWW users access to these data through a gis2web server. Gis2web allows distributed GISs to dynamically update their databases and have this information available to the WWW through automatic retrieval by gis2web.

Section 2 presents background information and related works. Section 3 discusses the design and implementation of the gis2web architecture. Section 4 examines inter-application communications issues involved in gis2web and concluding remarks are presented in Section 5.

2 Background and Related Works
2.1 Geographic Information Systems
Geographic information systems store and access many and varied kinds of information. Typical sorts of information range from geologic and biologic inventory maps to census and remote sensing data [4, 5]. Use of such information is increasing in a wide range of research, education, commercial, and government decision-making contexts. Any collection of GIS information would include vector maps, raster images, tables, graphs, and text which are interrelated and dependent on each other. Thus, GIS users can manipulate the database by interacting with any form of the data. These basic data are used to generate different output maps with associated tabular data for different users. For example, a GIS database might have files of irrigation canals, such as the network depicted in Figure 1 for counties in south Texas. Users may be interested in overlaying flood channel data to determine suitable locations for water delivery and flood mitigation considerations. This could be of interest to builders, home buyers, planning groups, or government agencies. A typical GIS maintains the geographic information in the form of attribute tables, polygons and image maps. Depending on the type of information and analysis of interest to users, the GIS system performs operations to generate the results and display them in a suitable format.

2.2 World Wide Web
Internet-based information systems have seen a tremendous growth in the past few years, e.g., FTP (File Transfer Protocol) [6], Gopher [7], WAIS (Wide
The Wide Web (WWW) is gaining wide acceptance due in part to its flexible capabilities for transferring multi-media information and its integration with many existing, successful network tools. WWW employs a client-server computing paradigm in which the client requests the information, the server provides the data in multi-media and hyper-media formats such as text, audio, video, and links to other information sites.

2.3 The Gap Between GIS and WWW

Information in a GIS is usually not readily available on WWW because of different data formats. A number of projects have focused on making GIS information readily available for WWW users in order to close the gap between the two communities. BADGER (Bay Area Digital GeoResource) is a project providing geographic information for the San Francisco Bay area in digital format accessible through the WWW [10]. BADGER generates a sequence of digital maps of different details, and users can view or download the images as needed. The U.S. government has made efforts to standardize catalog repositories of datasets so that users can browse, evaluate, and order them efficiently. As part of this effort, the Federal Geographic Data Committee (FGDC) [11] has published its geospatial metadata standard [12]. At the state level, similar projects are underway. For example, the Texas Natural Resources Information System (TNRIS) [13] has made available on the WWW a library of remotely sensed data, a set of topographic maps, and other information describing land use, vegetation, and hydrology. The Texas General Land Office [14] lists land and minerals information, as well as environmental and economic information on the U.S./Mexico border. The information presented in these web sites requires significant human resources to enter and maintain the data to ensure that the contents are accurate and up-to-date.

3 Gis2web

3.1 Goal

GIS databases are a significant resource that is underutilized through the WWW. To address this problem, many existing sites extract information from a GIS and enter it into a WWW server manually. This approach is expensive, and often leads to inaccurate, out-dated information. Gis2web supplies automated processes that can extract information from GIS servers, utilize a gis2web server to parse the information, translate it into HTML, and make the formatted data available for WWW servers which WWW clients can access.

Our goal is to maintain autonomous operations on GIS servers, gis2web servers, and WWW servers. This allows GIS servers to update GIS databases independently of gis2web servers' extraction and collection operations and the WWW servers' responses to clients' requests. This eliminates manual intervention and supplies GIS information to WWW users on demand.

3.2 Gis2web functions

Gis2web is designed to provide the following functionality. 1) Periodically extract geographic information from GIS databases. 2) Collect information from GIS servers and send it to a gis2web server for processing. 3) Parse the information collected from GIS servers and convert it to formats used by WWW servers. 4) Communicate with WWW servers to provide information at users' request.

3.3 Architecture

Gis2web consists of five different modules, as shown in Figure 2 within the dotted box. The modules act as independent processes and communicate with each other using Internet protocols. The GISExtractor accesses information on GIS servers. The Parser interprets the GIS information, creating the HTML format and storing it on the gis2web server. The Messenger passes data between the GISExtractor and the Parser. The Gatekeeper acts as an interface between gis2web and WWW servers, and the Coordinator schedules and controls the other modules.

The following is a detailed description of each of the modules.

GISExtractor: The GISExtractor works closely with the GIS software, ArcView in the prototype. It extracts information from the GIS database and sends it to the gis2web server.

Messenger: The Messenger communicates with the GIS server and accepts information sent by the GISExtractor. The information can be in various formats, e.g., text, tables, or images. The Messenger takes the data from the GISExtractor as a binary stream along with their headers (see Section 3.4.2 for data formats) and stores them on local disk in separate files.

Parser: The Parser converts GIS information received by the Messenger into formats that can be recognized by WWW servers. This includes converting plain text into HTML, and converting images to GIF or other formats. The Parser also builds necessary hyper-links into the text where needed.

Gatekeeper: The Gatekeeper is an interface between the local WWW servers and gis2web. It takes the user request and retrieves information from the generated by the Parser and sends them to the WWW server.
3.4 Implementation Issues

The architecture and design of gis2web is general enough that it can be implemented in any language and operating systems environment that supports WWW, a programming interface to the GIS, and Internet protocols. Our current implementation includes the Coordinator, the Gatekeeper, the Parser, and the Messenger on the UNIX platform written in C++, and the GISExtractor on the UNIX platform written in the Arc/View native programming language, Avenue[15].

3.4.1 GIS Server

In the prototype implementation, the GIS server is an Arc/View server. As such, the files are organized as a base map with various geographic themes and tabular attributes. For example, the themes might cover hydrographic distributions, highways, geologic information, as well as political boundaries. Attribute data might have integrated census data, hydrologic parameters, and environmental parameters with the geographic themes. See Figure 1 for an example. The base map and various themes are stored in separate files, with attributes typically stored in a database format. Themes can be selectively displayed to create combinations of map information or to support queries of the tabular data requested by users. Furthermore, any of the data can be updated as they would in a typical GIS environment. All of the operations such as search, join, and print are supported. The task of the GISExtractor is to extract information from these sources and pass it along to the gis2web server.

The GISExtractor operates on the GIS database. In the prototype it interprets various Arc/View file formats and directory structures, including tables of attributes, vector geographic files, and raster image files. We have used Avenue, the native programming language in Arc/View, because of these requirements. Avenue provides a convenient interface to the rest of the UNIX programming environment through RPC, or Remote Procedure Calls, which can directly communicate with a UNIX process.

3.4.2 Messenger and Data Formats

The Messenger receives information from the GISExtractor and passes it to the Parser to be interpreted. Because the information is in various formats, a protocol is needed between the GISExtractor and the Messenger for them to communicate. Three types of information are recognized and processed by the gis2web agents: 1) free text, which is used to describe various information contained in images or tables, 2) comma delimited table information, which is a direct translation of attribute tables in the GIS database, and 3) image, which usually is a map of some kind. To handle these types of information, a common gis2web data format is used, as shown in Figure 3. Information transferred by the Messenger consists of a header and the body of the message. The header indicates the type of information being transferred, total amount of information, total number of messages for this transfer, and a sequence number of the current message. One piece of information may need to be broken into a number of messages for transfer due to the amount of information, such as a map image. Error checking may indicate re-transmissions is necessary.

3.4.3 The Parser

The Parser converts the various file formats received from the Messenger to a HTML compatible format for display. Currently, the Parser is responsible for putting plain text into HTML format with possible hyper-links, converting comma delimited Arc/View table information into HTML compatible table format, and converting PostScript-based image to GIF image using an external program.

4 Gis2web Communications

GIS and WWW are two independently developed applications areas. The intent of gis2web is to provide a communication channel between the two applications so that rich GIS data sets can be readily available to WWW users. This section examines the many communication issues in gis2web.

4.1 GIS Server and GISExtractor

Arc/View is used in our prototype as the GIS server. Arc/View provides graphics user interface to the users, which lets users modify existing interfaces or create new ones using a native object-oriented programming language, Avenue. Additionally, Avenue provides inter-application communication facilities through RPC (Remote Procedure Calls) which can communicate with any processes running under the UNIX operating system. Figure 4 shows how a GIS server running Arc/View can communicate with regular UNIX process through RPC.

In order to use the RPC facility in Arc/View, Avenue programs are written in the Arc/View development environment, and can be exported as external text files and executed. The Avenue program would be called to extract the GIS information, to have these data sent to the GISExtractor in various formats (texts or images) through RPC calls. A header section is sent along with the data specifying the type and the size of the data. The receiving side (GISExtractor) then can take proper actions with these data.
Currently gis2web recognizes three types of data: images in GIF or PostScript format, comma delimited tabular information, and free format texts.

4.2 The GISExtractor And The Parser

The GISExtractor and the Parser are implemented as two independent UNIX processes communicating through TCP/IP sockets. The Parser is responsible for collecting data from the GISExtractor and saving the data in various files according to data types (i.e. images, comma-delimited tabular information, and free-format texts). The socket between the two is constantly open to serve on-demand requests sent from WWW users.

4.3 Other Issues

Once the Parser has saved the information in specific files, it calls the Converter to transform these data into suitable formats for WWW clients to display. The WWW clients access the converted data through the browser. The browser passes requests to the gis2web Gatekeeper which in turn invokes the retrieval of the information. If the requests cannot be served because of missing data, the Coordinator would send the requests back to the GISExtractor to retrieve data from the GIS server on demand. Depending on the storage capacity of the gis2web server, certain GIS data can be cached on the gis2web server, saving processing and communication time for retrieving these data from the GIS server.

5 Concluding Remarks

In this paper, we described the architecture and communication issues of gis2web, a system that enables data exchange and information sharing between widely used GIS software and the WWW. Gis2web can provide one element in making the wealth of information stored in GIS databases available through the WWW. The system consists of five major modules in which: the GISExtractor directly accesses GIS databases; the Messenger transfers information from GIS servers to the gis2web server; the Parser converts GIS information into formats for WWW servers; the Gatekeeper acts as an interface between WWW servers and gis2web; and, finally, the Coordinator controls the system’s general operation. Key features of gis2web include: GIS servers’ autonomous operations and WWW servers’ allowing the GIS information to be updated independently of WWW operations; automatic, periodic retrieval and transfer of GIS information to WWW servers; and support of multiple data formats. The framework proposed here can be used in other similar applications. The communications between GIS servers and gis2web is through RPCs to GIS servers controlled by gis2web. Three types of data are transferred between GIS servers and the gis2web server: images, comma-delimited tabular information, and free-form texts.

Acknowledgment

This research was supported by NASA grant NAG9-852.

References