Web-Based GIS

Anselmo Cardoso de Paiva
University of Maranhão, Brazil

Cláudio de Souza Baptista
University of Campina Grande, Brazil

INTRODUCTION

Currently, the Internet is becoming the main vehicle for publishing geographical information, which enables data interchange, analysis, and geographical data visualization. The rapid evolution of Web technology has lead to an improvement of the geographical information utilization and availability.

The rise of the Internet has created an infrastructure ideally suited to the widespread distribution and dissemination of geographical information. By using Internet GIS applications, users may view, query, analyze, and download spatial information from anywhere at anytime. While this improvement provides new opportunities for public domain as well as commercial use of spatial datasets, new problems arise. One of them is the problem of transferring data efficiently from server to client. Geographical datasets are generally very large, and this process may demand too much time.

The Internet has created an interesting environment for geospatial data sharing, in which data providers make their databases available through the Web, and users may transfer, visualize, manipulate, and interact with them (Bertolotto & Egenhofer, 1999). This environment introduces new problems that must be addressed to make possible an efficient and effective use of these datasets. One such problem is related to the availability of huge geospatial data amounts in repositories with limited connection bandwidth (Flewelling & Egenhofer, 1999).

One of the main concerns on Web-based geographical information systems (Web GIS) is related to performance issues, as these datasets need to migrate, as fast as possible, from server to client tiers (Peng & Tsou, 2003). This is the problem of generation and transmission of the digital maps, which are suitable for user needs. Another important issue is to enable fast and easy Internet GIS application deployment.

In this chapter, we discuss solutions proposed for Web GIS based on the vector format, particularly the iGIS framework (Baptista, Leite Jr., Silva, & Paiva, 2004). The iGIS is a Web GIS that renders maps using multiresolution techniques, and enables user interaction them, using the W3C SVG specification. The chapter also addresses the issues that affect Web GIS performance, presenting an overview of the techniques that increase Web GIS performance and the results obtained by using the iGIS framework.

BACKGROUND

Web GIS applications evolved from the delivery of static maps conveyed in raster formats (e.g., JPEG, GIF, PNG) to an actual stage in which users may choose the window, the scale, and possibly also the data layers to be displayed, and the map is generated dynamically from a database and transmitted in the vector format. Web mapping deals at least with two basic problems: map generation, and map transmission.

Regarding map generation, there are, basically, two approaches. The first one is based on the use of a database that stores geometry information at the most precise scale, and each visualization at less precise scales are automatically derived from that, mainly through cartographic generalization (Müller, Lagrange, Weibel, & Salgé, 1995; Weibel & Dutton, 1999). This implies the execution of map generalization procedures on the fly (Harrie, Sarjakoski, & Lehto, 2002; Lehto & Kilpelainem, 2001). The second approach utilizes a multiresolution database in which spatial objects may be associated to a variety of geometric representations that are scale dependant (Zhou, Prasher, & Kitsuregawa, 2001). In this case, it is necessary to process an off-line computation of a multiscale database containing several independent levels of detail.

Associated with the map generation problem, there is also the problem of vector map transmission. A possible solution is the use of progressive data transmission that is well known and successfully applied to raster images, in which coarser versions of the data are displayed before the complete image is downloaded.

One of the first works on progressive vector data transmission on the Internet has been proposed by Buttenfield (1999, 2002). His work focuses on the hierarchical subdivision of vector data by using tree structures based on the Douglas-Peucker generalization algorithm (Douglas & Peucker, 1973) similarly to the BLG tree by (Oosterom & Schenkelaars, 1995).

Bertolloto and Engenhofer (2001) did a work based on progressive transmission of vector data in conjunction with on-the-fly mapping over the Internet. They proposed the creation of multiple dataset representations corresponding to different levels of details. These levels of details are then sequentially transmitted and added to the currently displayed representation. Although the proposal is very interesting, it remains an open issue; the integration of two different levels of details dataset on the client side in a way that maintains the visualization completely and correctly at each step.

WEB-BASED GIS: DEVELOPMENT TOOLS AND PERFORMANCE ANALYSIS

The iGIS framework makes it possible and easy to build a GIS application in few minutes. By configuring an XML file using the deployment tool, a user can put any application online. Firstly, it is necessary to populate the database. Then, the user sets data location by providing the parameters: login, password, and database URL; the layers to be displayed; the colors that are going to be used; and other parameters.

The iGIS architecture is based on three tiers: presentation, application, and database. In the presentation layer, Java Server Pages, JSP, are used for implementing dynamic pages. Scalar vector graphics (SVG) format and JavaScript are used for map drawings and graphical tools. SVG was chosen due to many reasons including the fact of being a W3C recommendation; the ability of using vector maps with client manipulation operations such as zooming and panning; and map data compression.

The application layer is responsible for the business logic, and it is composed of the following modules:

- Data loader, which is responsible for loading data from different data sources, configured in XML. To add a new data source, it is necessary to extend some classes from this package;
- Application data model, which uses the OpenGeoSpatial Simple Features standard for implementing the system logic classes;
- Data formatter, which formats data according to the rendering type chosen. This module is extensible and is configured using XML. Currently, the iGIS uses SVG for rendering; and
- Static maps, which loads and stores SVG maps, improving system performance.

Finally, the data layer contains different data sources. Currently, there are drivers for Oracle, IBM DB2, and Postgresql database servers, and spatial data in ESRI shapefile format. Again, this module can be extended to support other data source types.

iGIS copes with spatial and nonspatial data in an integrated way: both data types are stored in the database server and manipulated via data manipulation language and data definition language statements. For example, information about a hospital would include name, address, number of rooms, physicians, equipment, and its latitude and longitude coordinates. Fortunately, there are several database servers with support to spatial dimension, such as Oracle, IBM DB2, Postgresql, MySQL, and, more recently, the Microsoft SQL Server. Therefore, by using SQL statements, users may submit spatial queries together with nonspatial ones.

iGIS works with different measure units that facilitate internationalization. Moreover, iGIS implements multiresolution by enabling users to navigate through several levels of detail in a map. For example, an application initially presents information at country level of detail. Then, after some operations of zooming in, it can present information at state level of detail, then at city level, and so on. In order to achieve that, iGIS deals with static and dynamic map generation.

On the other side, it is also important to address the aspects that affect the Web GIS performance once it is required to a Web-based GIS an efficient data transmission for the publication of spatial information. Furthermore, the provision of adapted data to heterogeneous clients, which may range from mobile to desktop ones, is another important requirement. Thus, the use of optimization techniques in order to obtain both data reduction and just-in-time delivery mechanism are also relevant to reach high performance. Some of these techniques, as pointed out by Baptista, Nunes, Sousa, Silva, Leite Jr., and Paiva (2005), are data simplification, relative coordinates, static maps, multiresolution, compression, on-demand loading, and progressive transmission.

Data simplification is one of the most used techniques for Web GIS optimization. Depending on user needs, a more detailed view may be required in map rendering. However, for some users, it is enough to have a broader view of map contents and some details in few zoom levels. Hence, map rendering can be done according to the details required by users. Moreover, maps are rendered for clients using their specific resolution, which may also introduce information loss.

Data simplification is a process that identifies, a priori, the map information that can be omitted, and removes it before map transmission.The amount of transmitted data without loss in visual quality is reduced. Map coordinates are generally set with high precision; consequently, the number of points is too large. The simplification process will reduce the number of transmitted points.

In general, vector graphics languages accept the representation of polygonal in two ways: using either absolute

Related Content