J Geograph Syst (2004) 6:95–116
DOI: 10.1007/s10109-004-0129-0

Journal of
Geographical
Systems
© Springer-Verlag 2004

The roles of geography markup language (GML), scalable vector graphics (SVG), and Web feature service (WFS) specifications in the development of Internet geographic information systems (GIS)

Zhong-Ren Peng[1] **and Chuanrong Zhang**[2]

[1]Director, Center for Advanced Spatial Information Research (CASIR),
Associate Professor, Department of Urban Planning, University of Wisconsin-Milwaukee,
PO Box 413, Milwaukee, WI 53201-0413, USA (e-mail: zpeng@uwm.edu)
[2]Ph.D. Candidate, Department of Geography, University of Wisconsin-Milwaukee,
PO Box 413, Milwaukee, WI 53201-0413, USA (e-mail: czhang@uwm.edu)

Abstract. The objective of this paper is to address two issues of current Internet Geographic Information Systems (GIS) programs – interoperability and graphic image output issues – using standard-based technologies, specifically, the Geography Markup Language (GML), Scalable Vector Graphics (SVG) and the OpenGIS Web Feature Service (WFS) Implementation Specifications developed by the OpenGIS Consortium (OGC). A strategy is proposed to use GML as a coding and data transporting mechanism to achieve data interoperability, SVG to display GML data on the Web, and WFS as a data query mechanism to access and retrieve data at the feature level in real time on the Web. Two case studies are reported to implement this strategy. Our case studies show that the combination of GML, SVG, and WFS has an immense potential to achieve interoperability while not requiring considerable changes to existing legacy data. Data can be in their original formats and still be retrieved using WFS and transformed into GML in real time. SVG can produce superior quality vector maps on a Web browser. More research is needed to explore the full potential of these new standards and to test them in real-world situations.

Key words: Geographic Markup Language (GML), Scalable Vector Graphics (SVG), OpenGIS Web Feature Service (WFS), Internet GIS, Interoperability.

1 Introduction

The Internet and the World Wide Web (WWW) have been widely recognized as an important means to disseminate information (Rohrer and Swing 1997; Doyle and Dodge 1998; Carver 2001; Pundt and Bishr 2002). It has

The authors would like to thank Mr. Nathan Guequierre for editing the manuscript, as well as three anonymous reviewers for their valuable comments and suggestions. The authors are entirely responsible for the contents of the paper.

increasingly been recognized that future developments in geographical information systems (GIS) will center on Internet GIS, accessing geospatial data and conducting geospatial analysis on the Internet (Plewe 1997; Peng 1999; Green and Bossomaier 2001; and Peng and Tsou 2003). There is active interest from researchers, practitioners, and vendors in exploiting Internet GIS and finding ways to improve accessibility to spatial data and spatial data processing services over the Web (David et al. 1998). Most, if not all, GIS vendors have created their own Internet GIS software, such as ESRI's ArcIMS, Intergraph's Geomedia Web Map, Autodesk's MapGuide, MapInfo's MapXtreme, GE SmallWorld Internet Application Server, and ER Mapper's Image Web Server. These Internet GIS software programs provide proprietary ways to allow users to access, display and query spatial data over the Web (Plewe 1997; Green 1997; Stand 1997; Su et al. 2000; Kowal 2002).

This trend has emerged to overcome several limitations of popular desktop GIS software packages. The first limitation of desktop GIS is that every user has to purchase a desktop GIS software package, prohibiting widespread use by the general public and small organizations who may have resource constraints in obtaining desktop GIS software and training. Moreover, the users would have to purchase the full desktop GIS package even though they may require and will use only a small percentage of the functions provided in the software. The second limitation is the inaccessibility to the Desktop GIS from locations other than the computer on which the desktop GIS software is installed. For example, GIS users cannot access the GIS data and analysis tools from remote conference sites or from the field. The third limitation of desktop GIS is the steep curve for learning to use the software. Due to the necessarily complicated user interface, it is difficult for general users to quickly acquire the necessary training to conduct even a simple task such as finding the shortest path from point A to point B. The fourth limitation of desktop GIS is proprietary technology and the lack of interoperability. Once the user makes a decision about purchasing GIS software from a particular vendor, it is very difficult and expensive to switch to another vendor product.

Due to these limitations, access to GIS technology is limited to a few trained GIS professions, and has not reached its potential for use by other non-GIS professionals who have more limited needs from and access to GIS technology. The proprietary nature of the current GIS software products also limits the sharing of data and technology among different departments even within the same organization. Since as much as 80 percent of public and private decision-making is based upon spatial analysis of one sort or another (Dessard 2002), it is clear that desktop GIS will not suffice to meet increasing needs for such analytical tools. Particularly, as in-car navigation systems and location-based services become popular, traditional desktop GIS is not equipped to serve the demands of the public.

Internet GIS has emerged to overcome these limitations and take advantage of the widespread use of Internet technologies. Internet GIS, or distributed GIS, is an Internet-centered GIS technology that uses the Internet as the primary means to access data, conduct spatial analysis, and provide location-based services (Peng 1999; Peng and Tsou 2003). Internet GIS has unique characteristics and advantages over desktop GIS. First, users have access to geospatial data and spatial analysis tools in any place with Internet access at any time. GIS usage is not limited to the office as with Desktop GIS software. This greatly increases the accessibility of GIS data and tools, for GIS professionals, non-GIS professionals, and the general public. Second, the simple and friendly Web interface greatly reduces learning time and thus attracts more non-GIS professionals. The World Wide Web is nearly ubiquitous, and thus learning to use a Web-based GIS interface may be more intuitive for non-professionals. Third, users needn't purchase full-blown GIS software. Internet GIS provides component-based geographic information services. Users can potentially pay, at the time of use, only for the services (functions and data) they utilize. No significant additional costs are required. With the popularity of Internet services, in-car navigation systems, and Internet-enabled mobile devices such as Personal Digital Assistants (PDAs) and cellular phones, Internet GIS and mobile GIS will play an important role in the lives of the general public and the future development of GIS technologies.

Current Internet GIS technologies rely on two major Web technologies: the Hypertext Markup Language (HTML) and the Hypertext Transfer Protocol (HTTP). However, neither HTML nor HTTP is designed to handle spatial data on the Web. HTML is a markup language that shows how each element in a text should be displayed on the Web browser, including information such as the font and the size of the text. It cannot be used to code spatial features like a point, a line, or a polygon. It is also static in that the user cannot change it once it is presented on the Web. HTTP is a communication protocol between the Web browser and the Web server. It is stateless, meaning that the Web server treats every request as a new request, so the user cannot define two points on the Web browser before sending out the request to the Web server. Therefore, for Internet GIS relying on HTML and HTTP alone, it is impossible to draw a circle or a square directly on the Web browser (Plewe 1997).

To overcome the limitations of the static HTML and stateless HTTP, some client-side applications have been developed, such as plug-ins, ActiveX Controls, and Java applets (Peng 1999). With these client-side applications, users can work with spatial data in the form of maps, make queries, select spatial features, and draw lines and polygons just as they can do on desktop GIS. These Internet GIS applications give the general Internet-enabled public access to both GIS technologies and data. With the increased availability, previous criticism of GIS as an elitist technology (Pickles 1995) may no longer be valid. We are now beginning to witness the popularizing of GIS, at least within information technology circles (Carver et al. 1998). Most of these Internet GIS provide basic GIS functions and can handle client requests by allocating tasks to specific Map Servers. They also provide additional GIS analysis functions such as spatial query, buffering, address matching/geocoding, labeling, thematic displays, and distance measurements. Some Internet GIS applications not only handle two-dimensional spatial information, but also deliver multi-dimensional geo-referenced data (Lin et al. 1999; Huang and Lin 2002). These Internet GIS applications make it possible for the public to use GIS functions without purchasing desktop GIS software.

Client-side browser add-ons also have specific limitations. The output received by the end-user from many current Internet GIS programs is in the form of raster image file formats such as JPEG and GIF. Therefore, the output maps lack cartographic quality and flexibility (Cecconi and Galanda 2002). This is because most of the user requests are handled by a map server at the server end. The output image file format is embedded in HTML pages and sent back to the Web client. In comparison with vector data, a raster format image has several disadvantages. First, raster map images tend to be of fairly low quality. The quality of the raster image is degraded as it usually has a low resolution in order to reduce the file size. Therefore, when the user zooms for greater detail, the graphic image becomes blurred as the image's pixel structure is magnified. As mentioned by Dash and Lawton (1996), a major problem with raster images is overlapping lines. A line or series of overlapping rectangles become blurred in a raster image. The second problem of raster images is that they require a large amount storage space, especially for high-resolution images. The large image data volume also slows down the data transfer speed between the Web server and the client. This is a serious problem for Internet GIS applications in terms of performance and responsiveness, as slow responses frustrate Internet users, and discourage them from using the technology. Server-side processing requires that every user request be transported to the server, thus creating high volumes of network traffic and reducing the responsiveness of the system (Peng 1999).

Another important limitation of current Internet GIS programs is that they are not interoperable. Each of these commercial Internet GIS programs has its own software design architecture and depends on specific database structures and formats. The mapping and geoprocessing resources distributed over the Web by these Internet GIS programs cannot be shared and interoperated. Although commercial Internet GIS products are making progress in performance improvement and ease of use, they have not yet made dramatic or revolutionary changes in Web mapping (Randy 2002), particularly in the area of standard-based system interoperability. A challenging task for the development of Internet GIS is to enable interoperability among heterogeneous systems and geospatial data.

Proprietary client-side applications work only with their respective servers and different client-side applications are unable to communicate with each other. The client-side applications cannot access data at the feature level directly from other data servers in real time (Peng 2003). This is unsuitable for many applications needing real-time access to data that are located on different remote servers, and often in different formats. For example, emergency response services need simultaneous access to many distributed GIS databases for a particular spatial feature (such as building information at the assessor's office in Oracle database, transportation data at the department of transportation's office in the GeoMedia format, environmental data at the environment agency in Shapefile format) (Peng 2003). Current Internet GIS programs cannot satisfy the time requirements for these kinds of online GIS services.

To enable interoperability, approaches using distributed object technologies like CORBA have been proposed (Zhu 2001), but their implementation has only slowly been adopted. Open GIS Consortium (OGC) has initiated

Web mapping interoperability initiatives and specifications (OGC 2003a), including the Catalog Services Implementation Specification, the Web Map Service Implementation Specification, the Geography Markup Language (GML) Implementation Specification, and the Web Feature Service Implementation Specification (OGC 2003b). The OGC has organized two Web Mapping Testbeds to evaluate those specifications. The OGC specifications can potentially make the Internet an open system and extend the Internet through a new generation of standards based on Extensible Markup Language (XML) (Randy 2002). The OGC Web mapping specifications offer a standard way for users to search for maps and geoprocessing sources over the Web from different map servers and different vendors (Limp 2002). But to make these OGC specifications work, more applications needs to be developed and more tests need to be done to verify the validity and efficiency of these specifications.

This paper addresses two issues of current Internet GIS programs – interoperability and graphic image output – by using these OGC Web mapping interoperability specifications, particularly, the GML and the OpenGIS WFS Implementation Specifications (OGC 2003b), in addition to the SVG, a W3C specification (W3C 2001). The idea is to use GML as a coding and data transporting mechanism to achieve data interoperability, to use SVG to display GML data on the Web, and to use WFS as a data query mechanism to access and retrieve data at the feature level in real time on the Web. The purpose is to test the applicability and efficiency of integrating different standard-based technologies like GML, SVG, and WFS in promoting interoperability of Internet GIS programs and improving cartographic quality of Internet GIS visualization on the Web. This is necessary because the OGC's specifications were developed in a rather fragmented manner. The case studies will test whether these specifications can work together effectively and efficiently. The following sections will introduce the concept of GML, SVG, and WFS, together with two case studies that implement these concepts.

2 GML, SVG and WFS

The Geography Markup Language (GML) is "an XML grammar written in XML Schema for the modeling, transport, and storage of geographic information including both the spatial and non-spatial properties of geographic features" (OGC 2003c). It is developed as an implementation specification by the Open GIS Consortium to foster data interoperability and exchange between different systems. GML 2.1 is based on the OGC Abstract Specification (OGC 2003d) that models the world in terms of features. A feature is an abstraction of a real world phenomenon; a geographic feature is any real-world object that is associated with a location. GML 2.1 models the world in terms of simple features, which are "features whose geometric properties are restricted to 'simple' geometries for which coordinates are defined in two dimensions and the delineation of a curve is subject to linear interpolation" (OGC 2001a). GML 3.0 can represent real-world phenomena using more complex feature types, "including features with complex, non-linear, 3D geometry, features with 2D topology, features with temporal properties, dynamic features, coverages, and observations" (OGC 2003 p.18). In addition, GML 3.0 also conforms to the ISO standards, including ISO DIS 19107 Geographic Information – Spatial Schema, ISO DIS 19108 Geographic Information – Temporal Schema, ISO DIS 19118 Geographic Information – Encoding, and ISO DIS 19123 Geographic Information – Coverages. GML offers standard ways to describe these spatial features and their corresponding properties in terms of GML Schemata, including schemata to describe features, coordinate reference systems, geometry, topology, time, units of measure, and generalized values. GML applications that follow these GML Schemata standards can be interoperable. Furthermore, GML provides XLink and XPointer mechanisms to make geospatial data interoperable. Through XLink and XPointer, different features and feature collections, which may be located remotely, can be associated together at the feature level (Peng 2003).

GML is based on XML, a data description language with strict hierarchical data structures to facilitate data search and discovery on the Web. GML is very flexible and extensible in encoding geographic features. It allows users to define their own tags or elements to describe features. This is in contrast with HTML, with limited and fixed tags to describe contents. Furthermore, like XML, GML is only concerned with the representation of geographic data content. It does not specify how GML data should be presented. To represent the geospatial features in the form of maps, GML data have to be transformed through Extensible Stylesheet Language Transformations (XSLT) to style the GML geographical contents into one of the graphical formats, such as Scalable Vector Graphics (SVG), Vector Markup Language (VML) or the Extensible 3D (X3D) Graphics specification. This process specifies how each element in the GML data should be displayed; for example, a red dot can be used to represent a point feature, or a blue line to represent a stream. Figure 1 illustrates the simplified process of displaying GML data on the Web browser.

Scalable Vector Graphics is an XML-based language used to describe an image, especially for display in the Web browser. It is a standard developed by the W3C (World Wide Web Consortium). As the name indicates, SVG is a vector graphic, which is different from the raster image formats such as GIF, JPEG, and PNG. A vector graphic uses mathematical statements to describe the shapes and paths of an image. A raster graphic is a description of a bit pattern using a grid of x and y coordinates and display or illuminate

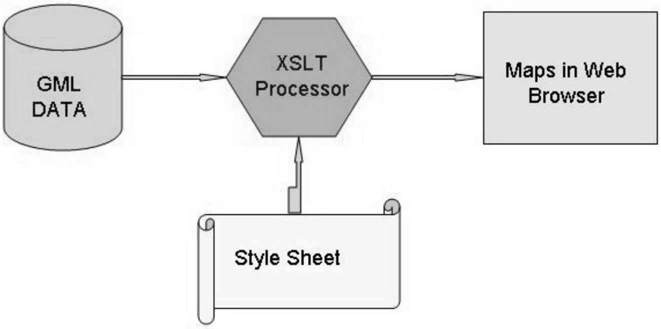


Fig. 1. The GML map making process

information in monochrome or color values on a display space. The use of the word "scalable" in SVG has two meanings. First, vector graphic images can easily be made scalable, i.e., not being limited to a single and fixed pixel size. This means SVG format can be displayed on any device of any size (whether a cell phone or a 19" computer monitor) and any resolution without changing the image clarity. This contrasts with raster image files, which are difficult to modify without loss of information. Second, a particular technology can grow to a larger number of files, a large number of users, and a wide variety of applications on the Web (W3C 2001).

Other characteristics of SVG include a smaller file size and searchable text information. An SVG file is usually smaller than a raster file for the same map resolution and thus can be transferred across the Internet more quickly. Text information inside SVG is still text and can be searchable, while text information inside the raster file becomes integrated into the image and is no longer recognized as text. SVG is also particularly suitable for displaying intelligent maps, because geometric objects such as points, lines, and polygons are recognized as such and are identifiable. Raster images on the other hand contain information about every pixel, and points, lines and polygons that are no longer recognizable. Therefore, the user can directly work with spatial features on an SVG but not on a raster graphic image.

SVG is also based on XML and therefore conforms to other XML-based standards and technologies, such as XML Namespace, XLink, and XPointer. XLink and XPointer allow for linking from within SVG files to other files on the Web, like a GML data element, HTML pages, or other SVG files (Boye 1999).

For the development of Internet GIS, SVG has the potential to play an important role for three reasons. First, it reduces the size of the map images by allowing complex scalable cartography in a highly compressed form. Second, as an XML application, SVG provides hyperlinks to many other files and vector and raster graphics. It can also work directly with other XML-based technology. Third, since an SVG file is an XML file, it offers superior portability (George 2001). That is, an SVG file could be edited and displayed in any environment regardless of computer operating systems and Web browsers. The combination of these three characteristics means that SVG can play an important role in the development of Internet GIS.

While GML provides a means to encode and transport geospatial features into XML, SVG provides a means to display these GML-coded geospatial features into vector maps on the Web. One issue of concern is how to conduct queries and extract features from the database to respond to user requests. The Web Feature Service (WFS) Implementation Specification has been developed by the Open GIS Consortium (OGC) to serve this role.

The WFS is an OpenGIS implementation specification (OGC 2002) that allows a client to retrieve geospatial data encoded in GML from multiple Web Feature Services. The WFS is written in XML and uses GML to represent features, but the database (or datastore in OGC's term) could be in any format. In fact, the structure of those databases should be opaque to client applications. Any access to the database should be through the WFS interface. The data retrieval process using WFS is shown in Figure 2. The client application sends a request in XML to the WFS, which connects with the datastore, processes the requests, and sends the response back to the

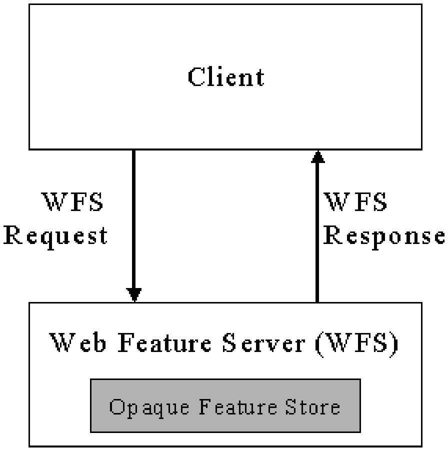


Fig. 2. Simplified WFS architecture (*Source*: OGC 2001b)

client. The communication between the client application and the WFS uses HTTP as the distributed computing platform.

The WFS allows the client applications to access, query, create, update, and delete data elements from the GML feature database server. The WFS provides interfaces for four basic data manipulation operations on GML features; creating, deleting, or updating a feature instance, or retrieving and querying features based on specified spatial and non-spatial restraints. Client applications can post request for feature level data in XML. The request includes query or data transformation operations, which can be applied to one or more features in one or more datastores, locally or remotely. The WFS server reads and parses the request and returns the result in the form of GML.

GML, SVG, and WFS are standard technologies, each of which has a unique role on the Web and Internet GIS. Combining them provides a greater potential to the development of Internet GIS. Two examples are now presented to test the feasibility of using GML, SVG, and WFS together to create a new kind of Internet GIS which is open, standard-based, interoperable, and with true vector graphics capabilities.

The first case study shows the encoding of spatial features in GML and displaying the encoded GML data on the Web in the form of maps using SVG. This case study illustrates the basic concepts of GML and SVG, as well as the process of GML encoding, transformation, and SVG display. The second case study uses WFS to access and query existing data (in ESRI's Shapefile format) at the feature level from the data server, and displays the query results in the form of SVG on the Web browser. The purpose of the second case study is to illustrate how WFS couples with GML and SVG for feature level data retrieval from different data servers.

3 Case study one – Encoding geospatial features in GML and displaying the GML data as SVG on the Web browser

This case study uses GML to encode basic spatial features (points, lines, and polygons) and display them in SVG on the Web browser. The working procedure of this case study is illustrated in Fig. 3. First, we encode the spatial

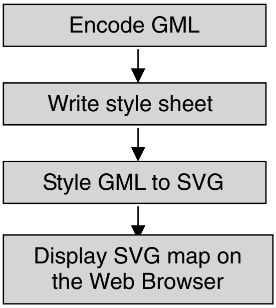


Fig. 3. Working process of GML encoding of spatial features and SVG display

features into GML (also called a GML feature data instance) and its corresponding GML Schema. Second, we write a Style sheet to define symbols used to represent the spatial elements in the GML spatial feature data. The style sheet defines the symbol representing a point, line, and polygon elements. Third, we transform the GML file, GML Schema, and the Style sheet into the SVG format. This is done with an XSLT processor. Currently there are two popular XSLT Processors — Xalan (Apache 2003) and Saxon (Kay 2003). With the help of the Saxon XSLT processor, the GML data and the Style sheet are transformed or "styled" into the SVG file for display. Finally, on the Web client side, the Adobe SVG Viewer plug-in is downloaded and installed to display the SVG file on the Web browser in the form of maps.

Only three real-world phenomena (or objects) are used in this case study: a building, a road, and a lake. These three spatial objects are coded in GML based on the simple feature model. That is, the building is coded in GML as a point feature, the road as a line feature and the lake as a polygon feature. These three features are viewed together as a feature collection in GML.

Two types of files are needed to encode the spatial features in GML: a GML application file and one or more GML Schema files. The GML application file is also called the Feature data Instance file of GML, which store the feature data instance. The GML Schema files are created to define the specific tags and elements used in the Feature data file, including geometry, topology, time, units of measure, and so on.

The following is an example of the GML codes to describe a simple point feature, the *building* element:

```
<Building fid = "Building1">
          <gml:description> Point example </gml:description>
          <gml:name> Shorewood Post Office </gml:name>
          <address> 2030 E. Greenway Road </address>
          <gml:location xmlns:gml = "http://www.opengis.net/gml">
            <gml:Point srsName = "http://www.opengis.net/gml/srs/epsg.xml#4326">
              <gml:coordinates>
                377,296
              </gml:coordinates>
            </gml:Point>
          </gml:location>
  </Building>
```

The corresponding feature schema file is shown as follows:

```
<complexType name = ″BuildingType″>
    <complexContent>
        <extension base = ″gml : AbstractFeatureType″>
            <sequence>
                <element ref = ″gml : description″ />
                <element ref = ″gml : name″ />
                <element name = ″address″ type = ″string″ />
                <element ref = ″gml : location″ />
            </sequence>
            <attribute name = ″fid″ type = ″string″ use = ″optional″ />
        </extension>
    </complexcontent>
< /complexType>
```

As is shown, the above GML application instance file and the GML feature schema do not store information about the display of the spatial feature. A Style sheet needs to be created to properly display it.

However, the Style sheet cannot be directly read by the SVG viewer (or SVG plug-in). As such the Style sheet and the GML were transformed into a SVG file as shown below:

```
<?xml version = ″1.0″encoding = ″utf – 8″?>
<svg width = ″500″ height = ″400″>
  <defs>
    <g id = ″Lib1.3″ transform = ″scale(0.25)″>
      <path style = ″fill-
              rule:nonzero;fill:#EBF322;stroke : #000000;stroke-miterlimit : 4; ″
              d = ″M68.262,74.51H0.5V37.255h67.762V74.51z″/>
    </g>
  </defs>
  <use xmlns:xlink = ″http : //www.w3.org/1999/xlink″ id = ″ ″xlink:href = ″#Lib1.3″
      transform = ″translate(369.6078431372549027,88.42105263157894560)
      scale(2)″/>
</svg>
```

Finally, the SVG file is interpreted by the SVG plug-in on the Web browser on the client side in the form of an SVG map as shown in Fig. 4.

4 Case study two – Feature-level data query and retrieval using WFS

This case study uses WFS to access and retrieve existing data files at the feature level and display them in SVG on the Web browser. The data used in this case study come from the Waukesha Transit Trip Planning Project, an

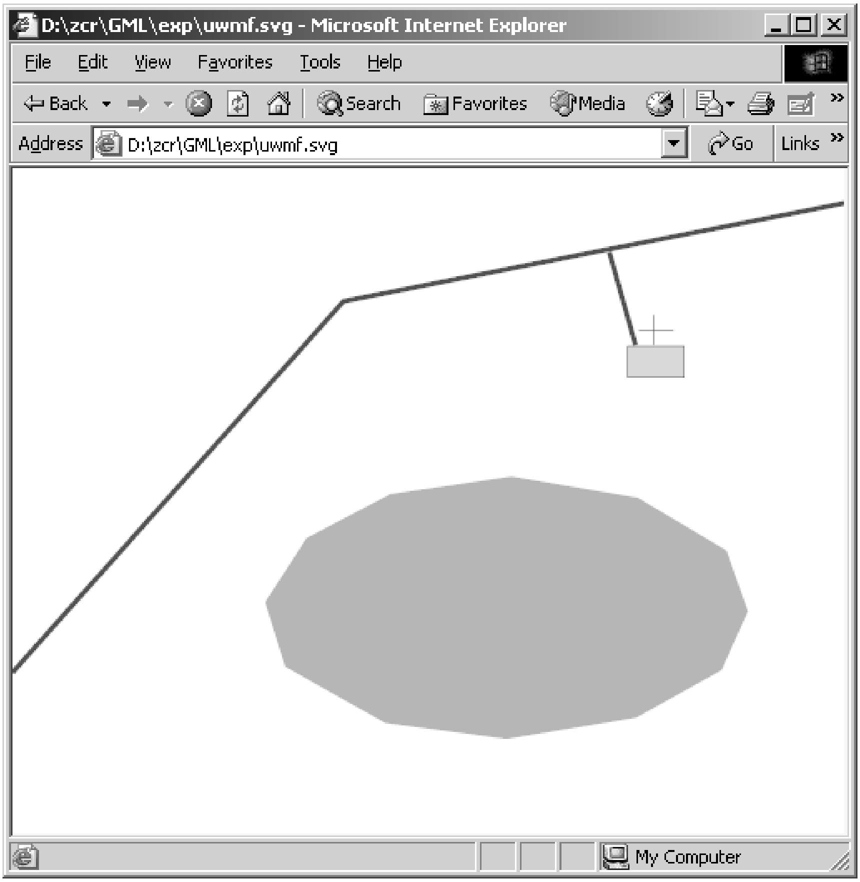

D:\zcr\GML\exp\uwmf.svg - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address D:\zcr\GML\exp\uwmf.svg
Go Links
My Computer

Fig. 4. A SVG map displayed on the Web browser

online bus trip-planning Website for the City of Waukesha, Wisconsin, which is available at http://metro-trip.ci.waukesha.wi.us/waukesha/. The data include a bus stop file, a bus route file, and a landmark/facility file that are located in three different servers. The files are originally in the ESRI Shapefile format. The working process of this case study is illustrated in Fig. 5.

The first step is to select and install a Web Feature Server. Currently there are several commercial WFS software programs available, such as MapServer, developed by the University of Minnesota which has WFS support (MapServer 1996), the Ionic Web Feature Server (Ionic 2001), GeoServer (GeoServer 2002), and GeoServerLite (Rosyada 2003). GeoServerLite was chosen for this case study because it is open-source software, has a graphic client, and all other supporting software is also open-source.

GeoServerLite is a simple Web Feature Server written in the PHP scripting language, using the freely available MySQL as its backend database. GeoServerLite also has a Geoclient to be used together with GeoServerLite as a client interface on the Web browser. Both GeoServerLite and Geoclient

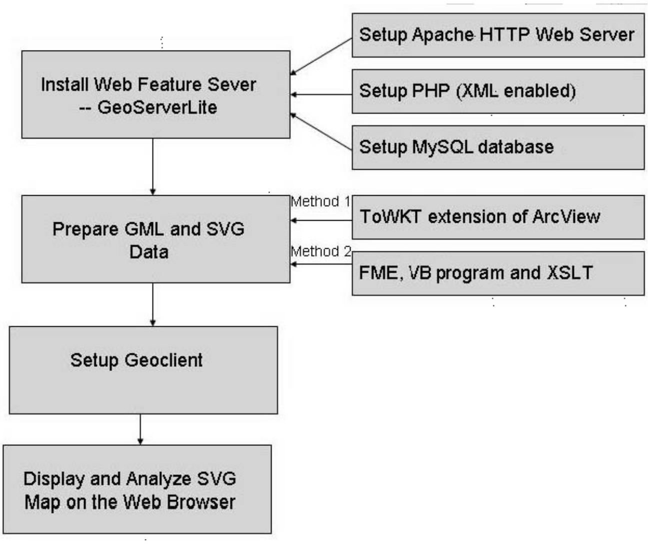

Install Web Feature Sever -- GeoServerLite
Setup Apache HTTP Web Server
Setup PHP (XML enabled)
Setup MySQL database
Prepare GML and SVG Data
Method 1
ToWKT extension of ArcView
Method 2
FME, VB program and XSLT
Setup Geoclient
Display and Analyze SVG Map on the Web Browser

Fig. 5. Working process of feature-level data retrieval using WFS

are developed under the GeoServer Project (GeoServer 2002) and can be downloaded from the Website at http://www.mycgiserver.com/˜amri/.

To work with GeoServerLite and Geoclient, the environment must be properly configured. This includes installing and configuring an Apache HTTP Web Server (Apache 1999); installing the PHP software (PHP 2001), which is a widely used general-purpose scripting language that is especially suited for Web development; and setting up the MySQL database for GeoServerLite. These required environment setups are unique to GeoServerLite and Geoclient. Other WFS servers may require other setups.

The second step is to prepare the GML and SVG data. The format of the original data files (Bus Stop, Bus Route, and Facilities) is in Shapefile. To setup the WFS server, these Shapefile data need to be converted into the GML and SVG format. Two methods can be used. The first is using ToWKT, an extension for ArcView developed by the Geoclient project. ToWKT can export the Shapefile files into a MySQL database in the "well known text" format. It can also transform the Shapefile data into GML data format or SVG data format. The PHP-based GeoServerLite can connect to the MySQL database, or the GML data, or the SVG data to serve them. The second method is using FME software and Visual Basic codes to convert the Shapefile data into GML data. The GML data are then converted into SVG files with the help of the XSLT processor and Style sheet. The second method is more flexible, while the first is easier. To display GML files on the Web, an SVG or a Java Applet is needed because the GML file cannot specify visualization elements and legends, such as how to display a point (a red dot or a star) or a line (a red solid line or a dashed line).

The GML data with SVG format are now ready to be published to the Web by GeoServerLite. The next step is to set up the GeoServerLite server and GeoClient, and placing the GML and SVG data into the corresponding directory. This step allows the WFS server to serve data to the clients.

On the Web client side, the GeoClient is utilized for user interactions. But before the GeoClient can be displayed, the Adobe SVG Viewer plug-in has to be installed (Adobe 2001). Alternatively, a Java applet could be written to display the GML or SVG data without GeoClient. With the GeoClient, the user can browse and analyze GML data in the form of maps on the Web browser. Figure 6 shows the graphic interface of GeoClient. Currently, there are only very basic map manipulation tools available in GeoClient, including zooming in, zooming out, and panning SVG data; displaying x and y coordinates as the mouse moves; displaying the legend; labeling a feature automatically according to its attribute field; and switching on and off different map layers. In addition, users can query the data as shown in Fig. 7, which illustrates one query result. Some importation transaction operation functions defined in WFS implementation specifications have not yet been implemented in GeoServerLite.

GeoServerLite publishes GML data in two ways: static GML data with SVG formatting (static GML data serving) and dynamic GML data in real-time (dynamic data serving). The difference lies in that serving static GML data requires the data to be pre-converted into GML and SVG, while serving dynamic GML data does not require this data pre-conversion. The first procedure requires that GML data are converted into SVG format on the server side by using the XSLT processor method (method 2 in Fig. 5) or using the toWKT method (method 1 in Fig. 5). When the client requests the data, GeoServerLite interprets the client request and sends back the converted SVG Data (with GML attributes and coordinates) and the related documents to the client. After receiving the SVG data, the client can display the data and perform GIS analysis with the data on his/her computer.

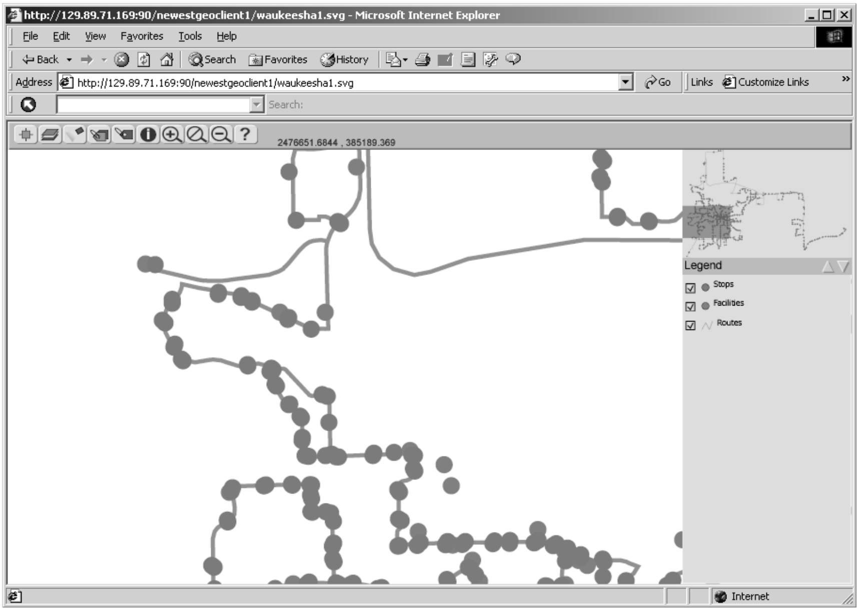
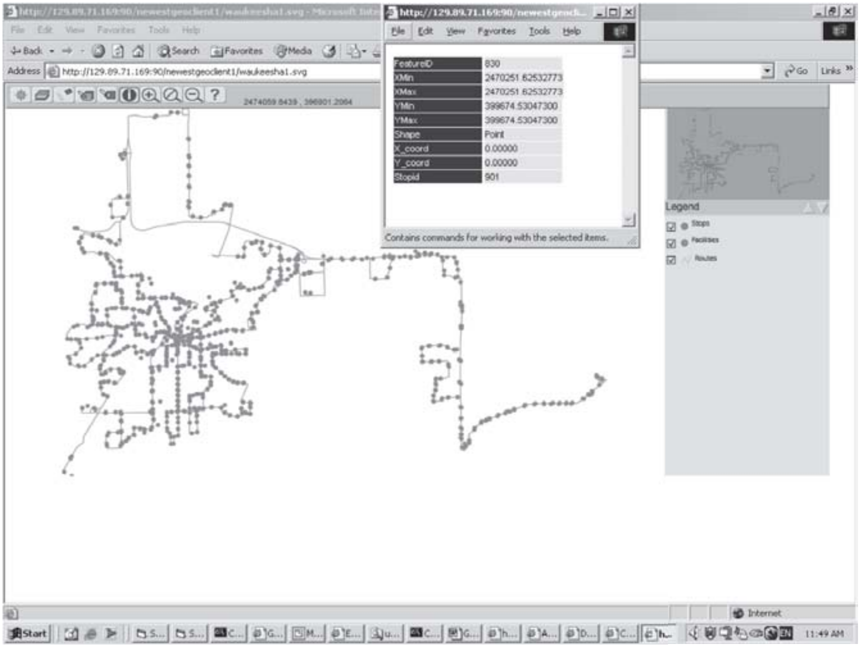

http://129.89.71.169:90/newestgeoclient1/waukeesha1.svg - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites History
Address http://129.89.71.169:90/newestgeoclient1/waukeesha1.svg
Go Links Customize Links
Search:
?
2476651.6844 , 385189.369
Legend
☑ ● Stops
☑ ● Facilities
☑ Routes
Internet

Fig. 6. The graphic user interface of the GeoClient


http://129.89.71.169:90/newestgeoclient1/waukeesha1.svg - Microsoft Inte
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
http://129.89.71.169:90/newestgeoclient1/waukeesha1.svg
2474059.8439 , 396901.2064
http://129.89.71.169:90/newestgeocli...
File Edit View Favorites Tools Help
Go
Links
Legend
Stops
Facilities
Routes
Contains commands for working with the selected items.
Internet
Start
11:49 AM

FeatureID	830
XMin	2470251.62532773
XMax	2470251.62532773
YMin	399674.53047300
YMax	399674.53047300
Shape	Point
X_coord	0.00000
Y_coord	0.00000
Stopid	901

Fig. 7. An example of a query result using GeoClient

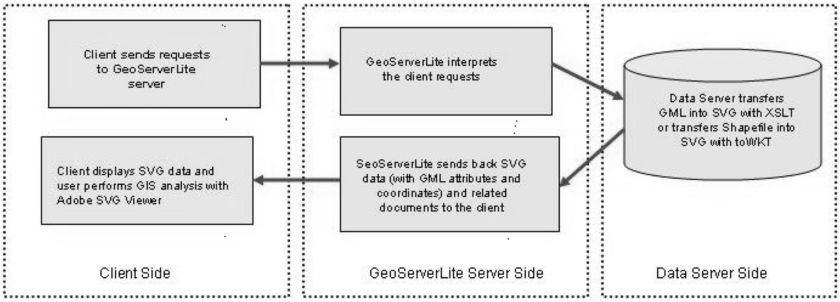

Client sends requests to GeoServerLite server
GeoServerLite interprets the client requests
Data Server transfers GML into SVG with XSLT or transfers Shapefile into SVG with toWKT
SeoServerLite sends back SVG data (with GML attributes and coordinates) and related documents to the client
Client displays SVG data and user performs GIS analysis with Adobe SVG Viewer
Client Side
GeoServerLite Server Side
Data Server Side

Fig. 8. Static GML data transfer process between client and server

Figure 8 illustrates the static GML data transfer process between the client and the server.

The other method involves dynamically publishing the GML data on the fly. This way, a client can access and retrieve GML data at the feature level in real-time on the Web. First, the client sends a series of requests (for example GetCapatibility, DescribeFeatureType, GetFeature, GetExtendedProject-Descriptor, GetStyledLayerDescriptor, GetExtendedLayerDescriptor) to the server. These requests are posted in the XML format. After receiving the requests, the server first parses these requests, then transfers the data stored in the MySQL database, which are converted by ToWKT from

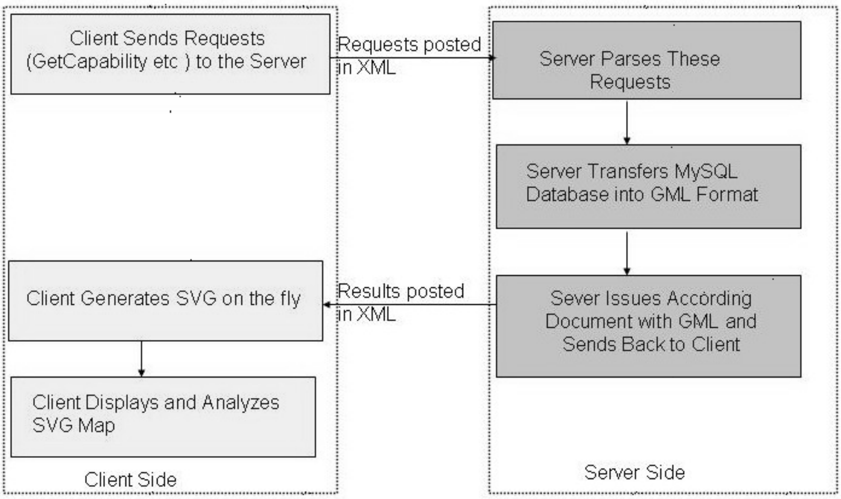

Client Sends Requests (GetCapability etc) to the Server
Requests posted in XML
Server Parses These Requests
Server Transfers MySQL Database into GML Format
Sever Issues According Document with GML and Sends Back to Client
Results posted in XML
Client Generates SVG on the fly
Client Displays and Analyzes SVG Map
Client Side
Server Side

Fig. 9. Dynamic GML data transfer process between the client and the server

Shapefile into the GML format on the fly. Next, the server produces corresponding documents with GML and sends back the query results to the client in XML. The Web client then generates the SVG on the fly with the posted XML data. Finally, the client displays the SVG data and the user performs GIS analysis with the help of Adobe SVG Viewer. Figure 9 illustrates the dynamic GML data serving process.

5 Discussion

In the process of conducting the case studies integrating GML, SVG, and WFS, we have shown that these three standards are compatible and hold great promise in the future development of Internet GIS. The advantage of this standard-based approach is interoperability. No proprietary client-side applications are required; only the generic SVG plug-in is needed. Data in their original formats were retrieved using WFS and transformed into GML on the fly. Thus, the end user can retrieve information at the feature level from different data sources with different data formats. Nevertheless, there are some limitations and issues to be resolved. Below is a summary of the lessons learnt so far.

5.1 Standards and interoperability

GML, SVG, and WFS are standards and thus are built upon an open system. As a result, an Internet GIS program that is based on GML, SVG, and WFS is created to be compatible with other standards including XML, XSLT, Document Object Model (DOM), SMIL, HTML 4.0, and XHTML 1.0. It also has sufficient accessibility options per the W3C's Web Accessibility Initiative (WAI) (W3C 2003). In our case studies, we found almost all open standards and application program interfaces (APIs) can be used to develop the Internet GIS applications. We did not have to be restricted to one language or one software program.

Because of the use of open standards, the greatest potential advantage of this standard-based Internet GIS approach is that it can access, query, and retrieve geospatial data at the feature level from different data sources, locally or remotely. Theoretically, the open standards provide the foundation for the Web client to communicate with any server that conforms to the WFS Implementation Specifications. Moreover, both GML and SVG support XLink and XPointer, which can be used to directly link data elements at the feature level at different data locations. Through XLink and XPointer, different features and feature collections can be associated together, regardless of locations. GML makes it possible to link geospatial data from other departments, other cities, counties, states, or across the globe (Peng 2003). Furthermore, both GML and SVG can link geospatial data with a wide variety of non-spatial data types including text, business transactions, graphics, audio, voice, and more.

Several demonstrations are available illustrating the implementation of interoperability of this standards-based approach. GML Relay Demonstration (Laser-Scan 2003), which was conducted by several organizations (Laser-Scan, Intergraph, Oracle, and Snowflake Software) in 2002, and the OpenGIS Web Mapping Testbed activities (OGC 1999), are two good examples of testing the interoperability of GML. These examples have demonstrated the interoperation capability of building services that combine data from different sources in a seamless client user interface (Watson 2001 and Lessware 2000). For example, Watson (2001) allowed an Ionic client to visualize a combined map output of imagery served from an MIT server, roads from a Laser-Scan server, and hospitals from an Object/FX server. But the client side of the GeoClient has not yet implemented the mechanism to communicate with other similar WFS-compliant servers in other locations, and it cannot communicate with other WFS servers at this moment.

On the other hand, there is a potential problem in implementing GML in terms of interoperability. GML provides a standard yet extensible language to encode geospatial features in the real world, along with their associated geometries and topologies. This extensibility is a strength, but it may bring about problems in data interoperability, particularly in the use of extensible schemata. GML is extensible in that it can extend the GML applications to encode different features, feature collections, geometries, and topologies. Unlike HTML, GML can create unlimited tags or elements to describe spatial objects. However, this great flexibility and extensibility could also create interoperability issues as the different implementation may cause confusion about semantics, a problem demonstrated in the GML Relay (GIS Monitor 2003).

5.2 SVG and vector map display

Our case studies showed that SVG can deliver better and higher quality maps over the Web than raster systems. Unlike raster-based images – such as GIF and JPEG formats – SVG is a standard vector format that offers very high quality maps at any resolution. Users can zoom in on any portion of SVG data without any degradation and can view and print the high-quality maps on the Web client. There are no "staircase" effects as seen when printing raster-based GIF and JPEG images.

Furthermore, because SVG conforms to the Document Object Model (DOM), it becomes possible to implement straightforward and efficient vector graphics animation via scripting. In addition, an SVG object can be assigned to a rich set of event handlers such as on_mouse_over and on_click, which makes SVG more dynamic and interactive.

In addition, since GML separates contents from the presentation, it offers the user more flexibility to display data. With the XSLT processor, users can convert GML data into different graphical presentations. For example, we may style the same GML data into two different SVG files so that the displayed maps have different presentations, as shown in Figure 10. It should be noted that there are other methods besides SVG to display GML data into maps. For example, we have written a Java applet to read GML data and display GML data directly on the Web browser in another project. However, since different Java applet codes may have different ways to display GML data, the output may not be as interoperable as the standard SVG codes.

5.3 Performance

The standard-based GML + SVG + WFS approach is a client-side application. That is, the GML and SVG data are sent to the Web client. Users interact with the GML and SVG data directly on the Web client without requiring a return to the server for every operation. This is in contrast with the server-side approach, where every user operation, even a simple zoom or pan operation, has to go back to the server for processing (Peng 1999). Furthermore, this client side approach has important implications for performance.

GML data are text-based, therefore they are easier to transport over the network than images. But when GML-coded geospatial data are transported, all the markup elements that describe spatial and non-spatial features, geometry, and spatial reference systems of the data are also

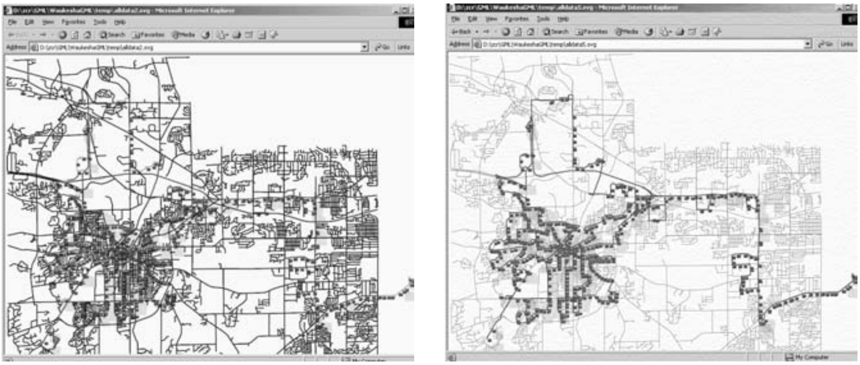


Fig. 10. Style the same GML data into different presentation graphics

transported to the recipient. This is important for data interoperability, because the GML-coded data could be saved and used by any other client-side applications that can read GML data. One shortcoming of using GML as a means of transporting data is its size. Compared with some binary GIS data format, the size of GML and SVG data files is large (see Table 1). For example, the size of the street file of the City of Waukesha increased from 3000 Kb in Shapefile format to about 9000 Kb after being converted into the GML format, and to more than 8000 Kb after being transformed into the SVG file. The bigger file sizes of GML and SVG files may hinder their use as a means of data transport over the Internet if the whole file needs to be transported to the client at once.

The size of the GML and SVG files is related to map resolution. By reducing the precision from millimeter to centimeter resolution, a smaller number of vertices is necessary and the GML and SVG file size is reduced by more than 50%. The size of the images, which are used by ArcIMS to transfer the spatial data, is relatively smaller (ranging from 7 kb to 60 kb for above test data) and is relatively insensitive to the data content. But the resolution of the JPEG image in the above test is only about 90 m (the whole map is about 540 km^2). For the same resolution, the JPEG image file would be even larger than the SVG and GML file. Therefore, there is a tradeoff between the size of the file and the resolution precision. With GML and SVG data files, the user can always get the full resolution map no matter how closely the user zooms in anywhere on the SVG map. But with raster image files, the image becomes blurred when the user zooms in to the details of the map image.

Table 1 also shows the response time between Geoclient and ArcIMS image servers. Geoclient downloads SVG files in full to the client while ArcIMS image server downloads the map images from the server to the client. Hence, when the file size is small (less than 200 kb for the SVG file), the initial download of the SVG file by Geoclient is faster than by ArcIMS. But when the file size is large (larger than 200 kb for the SVG file), the performance of Geoclient deteriorates rapidly compared with ArcIMS, because ArcIMS downloads only the map images rather than the vector data. Once the SVG data are downloaded from the server, the client side

Table 1. File size and performance comparison

Size of shape file (kb)	Size of GML file (kb)	Size of SVG file (kb)	Initial download time by Geoclient (second)	Operation response time by Geoclient (second)*	Initial image download time by ArcIMS (second)	Operation time by ArcIMS (second)
6	3	19	0.82	0.4	3.42	1.03
6	6	21	0.84	0.4	3.44	1.17
43	114	108	0.88	0.4	3.38	1.30
60	205	180	1.64	0.4	3.56	1.18
61	188	171	1.95	0.4	3.50	1.14
135	479	409	2.99	0.4	3.47	1.16
224	777	660	4.91	0.4	3.55	1.14
536	1851	1559	10.75	0.4	3.56	1.85
969	3208	2728	19.52	0.4	3.72	1.95

* approximate

operation by the user is very fast for Geoclient, much faster than the ArcIMS operation. This is because Geoclient works directly with SVG data on the client side while every operation in ArcIMS has to go back to the server for processing.

There are two basic means to improve the performance of SVG and GML. The first method is to use compression. Both GML and SVG data files are text-based, which are much easier to compress than other formats. In fact, a compressed SVG file format is already available. More research is needed to explore efficient ways to compress the GML and SVG data files on the server side and decompress them on the client side on demand. But compression and decompression may not solve the problem if the original file size is very large. For example, if the original GML file size is 5000 kb, a compression of 90% would still result in a compressed file size of 500 kb.

Another way to improve the performance is to send the GML and SVG data to the client in stages or progressively. That is, send out only part of the data to the client site initially. When the client requires more data, the client then goes back to the server to retrieve other parts of the data. The size of each part would be reasonably small. This will greatly reduce the size of each downloaded dataset and improve the Web application performance. With the implementation of WFS, it is feasible for the user to extract only a small amount of data at the feature level on demand. For example, the user can only extract a few roads, or even only a road segment rather than the whole area as in the case of other commercial Internet GIS programs like ArcIMS.

In summary, the standard-based GML + SVG + WFS approach to Internet GIS offers a significant performance advantage on the client side. The client-side approach makes the application more responsive to user interactions. Users can interact with the SVG viewer, conduct GIS analysis and get most responses locally rather than always having to go back to the server. Since spatial features in SVG are intelligent, they are more responsive to user requests than the traditional server-side approach. In our experience, this interaction with the GML and SVG files is faster than the current desktop GIS programs once the initial GML and SVG files are downloaded (seamlessly) to the local machine. But it does take more time for the initial downloading of SVG files from the server. This performance issue can be resolved by the combination of a better compression mechanism and the progressive downloading of SVG data.

5.4 Security issues in GML

GML is text-based, and can be opened using a simple text editor such as Notepad or any word processor. GML files are therefore easy to understand, edit, maintain, and update. Like HTML files, the user can right click the mouse to see and save the source codes of the GML and SVG files, which can then be imported, viewed, and modified in different environments. Security, however, is a drawback associated with text-based formats. Because GML and SVG files are text-based, and can be opened in any word processor, it is easy for users to accidentally alter the data. Fortunately, some XML editors have appeared on the commercial market that can prevent the accidental deletion of GML elements. Another drawback is that if some data are copyrighted, the data in GML format can be easily stolen. An encryption mechanism must therefore be developed to protect copyrighted or sensitive data.

6 Conclusions

By using the combination of GML, SVG, and WFS, this research has addressed two major issues impacting current Internet GIS programs: interoperability and the presentation of static raster images. Our studies show that the combination of the three standard technologies has great potential to address these two issues and more. GML is an effective means to encode, store, and transport geospatial data. It is also an efficient means to foster data portability and interoperability. SVG produces high-quality graphics on the Web, which is ideal for displaying spatial data and making intelligent maps. The OpenGIS Web Feature Service works well with querying and retrieving GML data from different servers.

The greatest advantages of GML, SVG, and WFS are that they are open standards, and that they are based on XML. Therefore, they can work with any other XML technologies and other standard APIs. We see the possibility that they will play an important role in fostering interoperable and distributed GIS in the .NET and Web Services framework. Conformance with XLink and XPointer, along with the use of other standards like Simple Object Access Protocol (SOAP) makes GML and SVG a great means to deliver and access component-based Internet GIS processor and geospatial data. This is a great step toward implementing truly interoperable geoprocessing applications and data over the Web.

However, there are still many issues to be resolved before this can be accomplished. The three standards are all relatively new; more research and experiments are needed to test these concepts. For example, the compression of GML and SVG files is an immediate need and probably the easiest issue to be resolved. More sophisticated client-side SVG user interfaces and data processing tools should be developed to assist users as they interact with GML data and conduct spatial analysis.

The integration of standard-based programs represents the future direction of Internet GIS, which is most likely to be Web Geoprocessing Services (WGSs) (Limp 2002; Hecht 2002). That is, the creation of a wide range of services that can be accessed across the Web, and Web services that can perform actions and return the outcome in a standard format to the client. GML, SVG, and WFS show great promise and could become key standard technologies to facilitate the development of Web Geoprocessing Services. By delivering geoprocessing services in the Internet's open, standards-based environment, it will make access to and use of geodata and geoprocessing resources much easier and less expensive. In the future, WGSs combined with a wide range of scientific and business models could provide GIS professionals and non-technical users convenient tools to perform complex analysis and build what-if scenarios on the Web based on spatial and non-spatial information. This open and standard-based Internet GIS would allow GIS technology to play an even greater role in society. Increased access to GIS analysis tools and data will certainly help both policy makers and the public to make better-informed decisions.

References

Adobe (2001) SVG Zone, available at http://www.adobe.com/svg/, last accessed on August 30, 2003

Apache (1999) Apache HTTP Server Project, available at http://httpd.apache.org/index2.html, last accessed on August 29, 2003

Apache (2003) Xalan-Java version 2.5.1, available at http://xml.apache.org/xalan-j/, last accessed on August 29, 2003

Boye J (1999) SVG Brings Fast Vector Graphics to Web, available at http://www.tech.irt.org/articles/js176/, last accessed on August 29, 2003

Carver S, Richard K, Ian T (1998) Accessing GIS over the Web: an aid to Public Participation in Environmental Decision-Making, available at http://www.geog.leeds.ac.uk/papers/98-3/, last accessed on August 29, 2003

Carver S (2001) Public participation using Web-based GIS. *Environment & Planning B*: *Planning & Design* 28: 803

Cecconi A, Galanda M (2002) Adaptive Zooming in Web Cartography. *Computer Graphics Forum* 21: 787–799

Dash J, Lawton G (1996) GIS hits the road. *Software Magazine* 96: 16

David JA, Kerry T, Ross A, Stuart H (1998) An exploration of GIS Architectures For Internet Environments. *Comput Environ and Urban Systems* 22: 7–23

Dessard V (2002) GML and Web Feature Server, available at http://www.geoinformatics.com/issueonline/issues/2002/maart_2002/pdf/38_41_ionic.pdf, last accessed on August 29, 2003

Doyle S, Dodge M (1998) Toward Virtual London: Developing a Virtual Internet GIS. In: Proceedings of International Conference on Modeling Geographical and Environmental Systems with Geographical Information Systems, June 22–25, 98, Hong Kong, pp 624–629

George R (2001) GIS meets XML SVG-Scalable Vector Graphics, available at http://www.academy-computing.com/svgWeb/svg-gis.html, last accessed on February 10, 2003

GeoServer (2002) The GeoServer Project, available at http://geoserver.sourceforge.net/html/index.php, last accessed on August 29, 2003

GISmonitor (2003) DOES GML ENABLE DATA SHARING? SORT OF..., available at http://www.gismonitor.com/news/newsletter/archive/011603.php, last accessed on August 29, 2003

Green DG, Bossomaier T (2001) *Online GIS and Spatial Metadata*. Taylor and Francis, New York

Green DR (1997) Cartography and the Internet. *The Cartographical Journal* 34: 23–27

Hecht LJ (2002) Web Services Are the Future of Geoprocessing. *GeoWorld* 15: 26

Huang B, Lin H (2002) A Java/CGI approach to developing a geographic virtual reality toolkit on the Internet. *Computers and Geosciences* 28: 13–19

Ionic S (2001) Geographic Markup Language, available at http://www.ionicsoft.com/communities/gml.jsp#gmlwfs, last accessed on August 29, 2003

Kay MH (2003) Saxon: The XSLT and XQuery Processor, available at http://saxon.sourceforge.net, last accessed on August 29, 2003

Kowal KC (2002) Tapping the Web for GIS and Mapping Technologies: For All Levels of Libraries and Users. *Information Technology & Libraries* 21: 109

Laser-Scan (2003) GML Relay Demonstration, available at http://www.laser-scan.com/demos/index.htm#, last accessed on August 31, 2003

Lessware S (2001) Interoperable Web Mapping and GML, available at www.posc.org/meetings/nov00/nov00_sl.ppt, last accessed on August 29, 2003

Limp WF (2002) Web mapping 2002. *GeoWorld* 15: 30–32

Lin H, Gong J, Wang F (1999) Web-Based three-dimensional geo-referenced visualization. *Computers & Geosciences* 25: 1177–1185

MapServer (1996) MapServer, available at http://mapserver.gis.umn.edu/, last accessed on August 29, 2003

OGC (1999) Open GIS Consortium Web Mapping Testbed Public Page, available at http://ip.opengis.org/archive/wmt/, last accessed on August 29, 2003

OGC (2001a) Geography Markup Language (GML) 2.0, available at http://www.opengis.net/gml/01-029/GML2.html, last accessed on August 29, 2003

OGC (2001b) OpenGIS Web Feature Server Implementation Specification, available at http://www.opengis.org/techno/discussions/01-023.pdf, last accessed on August 29, 2003

OGC (2002) Web Feature Service Implementation Specification, Available at http://www.opengis.org/techno/specs/02-058.rtf, last accessed on August 29, 2003

OGC (2003a) Overview of OGC's Interoperability Program, available at http://www.opengis.org/pressrm/summaries/20020813.TS.IP.htm, last accessed on August 29, 2003

OGC (2003b) OpenGIS Implementation Specifications, available at http://www.opengis.org/techno/implementation.htm, last accessed on August 29, 2003

OGC (2003c) OpenGIS Geography Markup Language (GML) Implementation Specification, available at http://www.opengis.org/techno/documents/02-023r4.doc, last accessed on August 29, 2003

OGC (2003d) OpenGIS Abstract Specification, available at http://www.opengis.org/techno/abstract.htm, last accessed on August 29, 2003

Peng Z-R (1999) An Assessment Framework of the Development Strategies of Internet GIS. *Environment and Planning B: Planning and Design* 26: 117–132

Peng Z-R, Edward B (1998) Internet GIS: Applications in Transportation. *Transportation Research (TR) News* 195: 22–26

Peng Z-R (2003) A Framework of Feature-Level Transportation Geospatial Data Sharing Systems. Paper presented at the Transportation Research Board Annual Meeting January 2003. Washington DC

Peng Z-R, Tsou M-S (2003) *Internet GIS: Distributed Geographic Information Services for the Internet and Wireless Networks*. John Wiley & Sons, New York

PHP (1999) PHP, available at http://www.php.net/, last accessed on August 30, 2003

Pickles J (1995) Ground Truth: the social implications of geographical information systems. Guildford Press, New York

Plewe B (1997) *GIS Online: Information Retrieval, Mapping, and the Internet*. OnWord Press, Santa Fe, New Mexico pp 311

Pundt H, Bishr Y (2002) Domain ontologies for data sharing-an example from environmental monitoring using field GIS. *Computers and Geosciences* 28: 95–102

Randy G (2002) Maximize online mapping with SVG/XML. *GeoWorld* 15: 42–44

Rosyada A (2003) Amri SVG, available at http://www.mycgiserver.com/˜amri/, last accessed on July 1, 2003

Rohrer RM, Swing E (1997) Web-based information visualization. *IEEE Computer Graphics and Applications* July/August: 53–59

Stand EJ (1997) Java creates new channels for GIS information. *GIS World* May: 28

Su Y, Slottow J, Mozes A (2000) Distributing proprietary geographic data on the World Wide Web – UCLA GIS Database and Map Server. *Computers and Geosciences* 26: 741–749

W3C (2001) Scalable Vector Graphics (SVG) 1.0 Specification, available at http://www.w3.org/TR/SVG/index.html#minitoc, last accessed on August 29, 2003

W3C (2003) Web Accessibility Initiative (WAI), available at http://www.w3.org/WAI/, last accessed on August 29, 2003

Watson P (2001) Proceedings of the OEEPE XML/GML Workshop, 19–20 November 01, Marne-la-Vallée, near Paris, France. (Available at http://www.eurogeographics.org/News/Events/oeepe_xml_workshop/powerpoint/LSL_GML.ppt, last accessed on August 29, 2003)

Zhu X (2001) Developing Web-based Mapping Applications Through Distributed Object Technology. *Cartography and Geographic Information Science Journal* 28: 249–258