

GML — AN OPEN-STANDARD GEOSPATIAL DATA FORMAT

Sheng Ye^{a,b,*}, Feng XueZhi^a, Shaotao Yuan^c, Ruan RenZong^d

^aSchool of Geographic and Oceanographic Sciences, Nanjing University, Hankou Road 22, Nanjing, 210093, P.R.C.;

^bNingbo Land and Resources Bureau, Ningbo, 315010, PRC

^cZhejiang Zhenhai Power Plant Co.,Ltd, Ningbo, 315208, P.R.C.;

^dCollege of Hydrology and Water Resources, Hohai University, 1 Xikang Road, Nanjing, Jiangsu Province, 210098, P.R.C - ys_ningbo@yahoo.com

Technical Session: SS-10: Standards and Standardization of Geo-Spatial Information

KEY WORDS: GML; Internet; Spatial Data Sharing; Geometry; Visualization

ABSTRACT:

In the last three decades, dozens of GIS data formats have been developed for encoding geographic information. Most were specified formats to particular vendor's GIS software. A few were open standards created to enable inter-vendor communication. As the Internet continues to expand rapidly the GIS can benefit greatly from the potential offered by Web based data delivery. A new common geodata format known as the Geography Markup Language (GML) has been specified by the OGC (Open Geospatial Consortium) through a unique marriage of the W3C's (World Wide Web Consortium's) XML (Extensible Markup Language). GML has advantages as XML-based geospatial data format. It is an open standard, which is easy to be manipulated and exchanged between different computer platforms and applications. GML will enable the development of a geospatial world wide web in which common data elements are shared between many data providers over the Internet enabling geodata to be locally developed, and globally integrated. Since GML only includes the content of the Geodata, a new vector graphic format: Scalable Vector Graphics (SVG) is used for presentation until a direct visualization tool for GML is created based on XML. This paper shows how GML is derived, how geodata can be stored and transferred in GML and visualized in SVG, convert GML to SVG via Extensible Stylesheet Language Transformations (XSLT), and make the SVG graphic dynamic with Hypertext Markup Language (HTML), Document Object Model (DOM), or JavaScript.

1. INTRODUCTION

1.1 General Instructions

The rigorous developments of GIS over the past decades have enabled application developers to create powerful systems that are used to manage spatial data. Wider access to data is becoming possible as the Internet and other technologies allow easier and global of information interchange. XML is being increasingly used for Web based applications as an open and standard markup language. XML-based languages have been developed in many application fields. GML is a key component of the geospatial web-services architecture being developed by OGC(Open Geospatial Consortium). As an open, non-proprietary standard, GML overcomes the problems of current GIS software proprietary data models and database structures (Zhang, C., 2003). It is used to carry the geodata between data servers and client applications that allow wide and efficient exchange and manipulation of geodata.

1.2 From HTML to XML

The Web was originally built on the HTML (Hypertext Markup Language) standard for encoding/decoding web site page descriptions that include simple images and formatted text. XML goes well beyond HTML in capability, specifying rules for encoding standards such as HTML. Thus XML enables a wide range and limitless number of data presentation and

structural schemas. Because recent versions of web browsers can interpret XML-encoded data and instructions, these browsers provide an open platform for receiving data and receiving data executing instructions for parsing, displaying and manipulating these data. The data and instruction are in separate content from presentation.

1.3 From XML to GML

Members of OGC saw the potential: that any web browser that supports XML might receive both geodata and the application-specific code needed to display and manipulate those data. They agreed to develop the "Geographic Markup Language" (GML), an XML extension for encoding the transport and storage of geographic information, that includes geometry and properties of geographic features (Open Geospatial Consortium, 2004). GML incorporates the definitions of geometry and properties as set forth in OGC's earlier OpenGIS Simple Features Specification. Its current status is an RFC (Request for Comment) under review within the OpenGIS Consortium. The RFC is supported by a variety of vendors including Oracle Corporation, Galdos Systems Inc, MapInfo, CubeWrx and Compuconsult Ltd. GML is implemented and tested through a series of demonstrations which became part of the OpenGIS Consortium's Web Mapping Test Bed (WMT) conducted in September 1999. These tests involved GML mapping clients interacting with GML data servers and service providers.

* Sheng Ye (1978-), female, born in Ningbo, Zhejiang Province, Ph.D student; GIS, digital image processing, remote sensing, LUCC. ys_ningbo@yahoo.com

GML1.0, based on XML1.0, was approved by OGC in May 2000. Following additional OGC Test Beds in the summer and fall of 2000, GML 2.0 was developed and passed as the successor to the GML 1.0 recommendation paper in February 2001. GML 2.0 was then passed in March as an OGC Adopted Specification. GML 2.0 is an "implementation specification", a working engineering specification supporting serious application development by anyone who wants to write software that encodes or decodes spatial information or process instructions in XML. In this paper, the GML used in the examples and case study is GML 2.0. At the OGC meetings in Liege, Belgium in April 2001, Consortium members began work on GML version 3.0.

2. GML

GML incorporates the definitions of geometry and properties as set forth in OGC's earlier OpenGIS Simple Features Specification (Open GIS Consortium, 1999). GML 2.0 defines three base schemas for encoding spatial information. The Geometry schema (geometry.xsd) replaces the DTD that appeared in GML 1.0. The Feature Schema (feature.xsd) supports feature types and collections and includes common feature properties such as *fid* (a feature identifier), *name* and *description*. The XLink schema (Xlink.xsd) provides the XLink attributes to support linking functionality. GML enables complex features, features collections and feature associations. For example, a city encoded in GML could have city name, postcode and any other desired properties stored as real values.

2.1 Application schema of GML

Database implementations are required to provide an application schema that defines the expressions for the geographic features that they support, and these are derived from the definitions found in the Feature schema. An application schema declares the actual feature and property types of a particular domain, using components from GML in standard ways. Broadly, these involve defining application-specific types which are derived from types in the standard GML schemas, or by directly including elements and types from the standard GML schemas. The application schema provides a rich set of primitive datatypes (e.g. string, boolean, float, date), and allows the creation of built-in and user-defined datatypes. The application schema offers several advantages when it comes to constraining GML encodings:

- 1) it enables the mixing and reusing of different vocabularies using namespaces;
- 2) it permits finer control over the structure of the type definition hierarchy;
- 3) it confers extensibility and flexibility via derived types and substitution groups

2.2 Getting GML data

There are three main methods to get GML data:

- Getting GML data from GIS software is suitable for users who have certain GIS data and corresponding GIS software. Many GIS companies including Intergraph have developed a component to export and import GML data recently.
- Getting GML data from translator software is suitable for users who only have GIS data but don't have corresponding GIS software. FME (Feature Manipulation Engine) from Safe Software Inc. allows more than 60 kinds of GIS data formats to convert to GML, and vice versa.

- Getting GML data from Web Feature Server (WFS) is suitable for users who have no source data.

2.3 Trend of GML

GML 3.0, published in Jan 29, 2003 includes at least some of the following features: GML encoding extensions for topology, temporals, covers (including images), spatial locators, points of interest, units of measure and 3D geometry.

GML can be used for Internet mapping, Web GIS, and location-based services for wire and wireless users.

3. VISUALIZATION GML WITH SVG

GML adheres to the principle of separating content from presentation, so it does not address the visualization of encoded data. To make a map, GML elements should be interpreted into a graphical format that could be visualized in a web browser. Potential graphical display formats include W3C's SVG, Microsoft Vector Language (VML), and Extensible 3D(X3D). GML could be interpreted either in raster or vector image. Using raster data is slower and more cumbersome than using vector data, since new raster data need to be sent from server whenever the user pans or zooms. This is fine for a small location on a company's website, but for new map-based services and GIS, users need detailed maps and dynamic systems, that allow efficient functionality. The obvious solution is to deliver data in vector format. However, Internet browsers only provide native support for raster images. Internet browsers have visualization problems for spatial data and require a specialized solution. A suite of new technologies such as SVG, XSLT and DOM create new alternative solutions. SVG is a language for describing two-dimensional graphics and graphical applications in XML. It is suitable to display 2D vector data and attributes. So GML is well suited to store geospatial data, and SVG provides a format for viewing it (sheng, 2005).

The next section focuses on generating SVG map from GML and displaying it in a web browser. The features and attributes in the GML file can be rendered by the XSLT map style sheet, and the GML data and XSLT map style sheet can be transformed into SVG using an XSLT engine, and then view the SVG graphic in web browser with the Adobe SVG viewer or other SVG viewer. JavaScript can be programmed with SVG to make SVG map interactive.

3.1 SVG

SVG is a modularized XML language for describing two-dimensional graphics with animation and interactivity, and a set of APIs upon which to build graphics-based applications (W3C 2004). SVG enables website developers and designers to create dynamically generated, high-quality graphics from real-time data with precise structural and visual control. With this powerful new technology, SVG developers can create a new generation of Web applications based on data-driven, interactive, and personalized graphics.

SVG allows for three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and polygons), images, and text. Graphical objects can be grouped, styled, and transformed into previously rendered objects. The feature set includes nested transformations, clipping paths, alpha masks, filter effects and template objects. Sophisticated applications of

SVG are possible by use of a supplemental scripting language which accesses SVG Document Object Model (DOM), which provides complete access to all elements, attributes and properties. A rich set of event handlers such as onmouseover and onclick can be assigned to any SVG graphical object.

3.2 Converting GML to SVG using XSLT

Since the Internet Explorer does not offer a special interpreter for GML, a special XSLT engine is needed to interpret the GML file and XSLT map style sheet. An XSLT style sheet provides a declarative specification for the transformation XML data. In this case, the attributes in the SVG file such as color, line weight, and font are defined in XSLT map style sheet. A map style engine is thus used to locate GML elements and a particular XSLT map style sheet is then used to interpret the

into existing DOM means SVG elements can be controlled and modified by the usual JavaScript interfaces. The DOM for SVG allows for straightforward and efficient vector graphics animation via scripting. Because of its compatibility and leveraging of other Web standards, features like scripting can affect HTML and SVG elements simultaneously within the same Web page. Thus the web page does not need to be reloaded when the event is triggered. The following functions can be realized in real time: layers are turned on and off; maps are zoomed in and zoomed out; the name of the region is show when the mouse is moved over the region; the info table of the city is jumped out when the city is clicked.

4 Internet solutions of GML/SVG

Client-Server computing model is the key to successful information sharing on the Internet/Intranet. Basically, Web

elements to SVG file. Certain extension functions must be provided by the XSLT Engine to process the complex calculations since a GML file has a lot of geometrical coordinates.

3.3 Make SVG Interactive

The combination of SVG, DOM, and a scripting language such as JavaScript allows the production of highly interactive web-maps. SVG elements are exposed through the DOM. JavaScript can create and manage graphical elements and animation over time or in reaction to user events.

The DOM is an Application Programming Interface (API) for valid HTML and well-formed XML documents. In this paper, the DOM is implemented using JavaScript. Integration of SVG GIS follows the Client-Server Model. There are two variations to the basic Web GIS application: the Client-side and the Server-side application.

The steps in GML map making can readily be performed on either the client side or the server side. The application of GML/SVG in web GIS requires large transmissions of data between client and server tiers, and demand for advanced spatial analytical capabilities. All these requirements indicate that a heavy burden will be placed on the server tier, so the three-tier model is necessary to degrade the overall performances.

4.1 Server-based solution for GML/SVG

The server-side solution for GML/SVG, shown in Figure 1, requires sophisticated hardware configuration on the server platform to handle the highly processing requests from the client

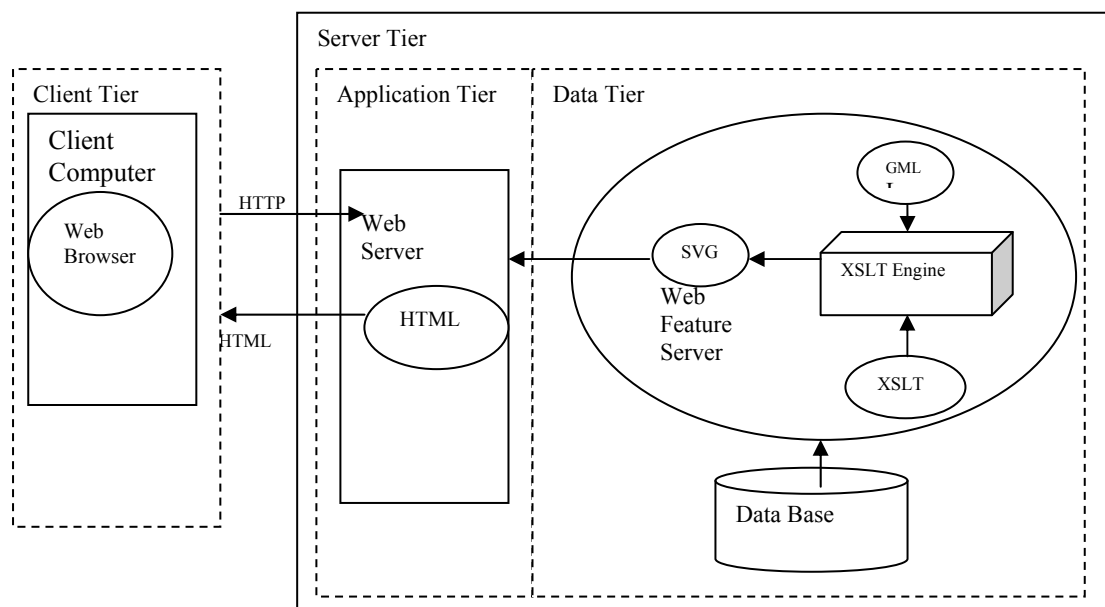


Figure 1. The Server-based solution for GML/SVG

In the server-based solution for GML/SVG, a single Request-and-reply cycle operates as follows:

- A Web browser, such as Netscape or Internet Explorer, makes a HTTP (Hyper-Text transfer protocol) request to the Web server.

- Having received the request, the Web server responds to the requests and re-directs part of the request to other appropriate resources such as Web Feature Server to get GML file.
- Once the web feature server has obtained the GML file

from the database, it then converts GML to SVG in the Data Tier. Finally it sends the SVG file to the Web server.

- The Web server makes a HTTP reply in the form of HTML documents embedded with SVG to the client.
- The client browser displays the received HTML documents and SVG files.

The server-side solution can eliminate the problems related to data incompatibility, data inconsistency and data unreliability. This is because all the GIS operations are channeled by the Web server (Apache, iPlanet Web Server, Microsoft Internet Information Server) and processed by dedicated data servers. All clients use the same graphical user interface to perform the same set of GIS functions on the same GML database provided by the data tier.

On the other side, there are some disadvantages of the server-side solution. Since the server must transmit HTTP and SVG file to clients for every GIS operation over the Internet (e.g., users reset a display window by panning or zooming, turn a layer on or off), a heavy loading in the network is unavoidable. The situation is even worse if the SVG file is big or many clients are connecting to the server with slow modems.

4.2 Client-based solution for GML/SVG

The client-side solution for GML/SVG, shown in Figure 2, is a highly client-dependent platform configuration which requires the client machine to take up all the responsibilities of processing GIS operations.

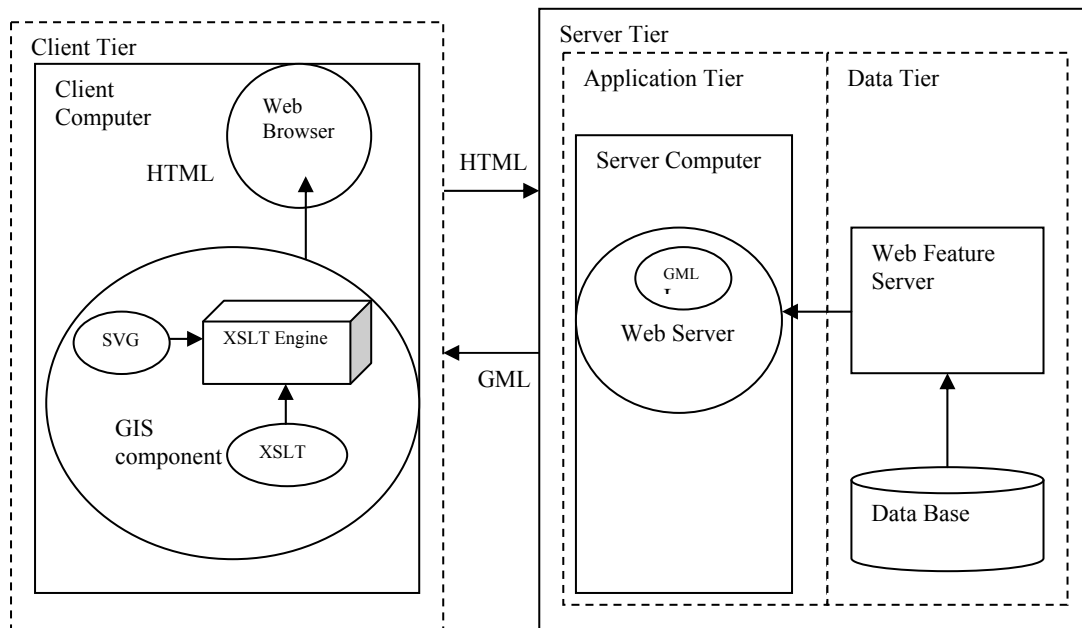


Figure 2. The Client-based solution for GML/SVG

In the client-based solution for GML/SVG, a single Request-and-reply cycle operates as follows:

- A Web browser, such as Netscape or Internet Explorer, makes a HTTP request to the Web server.
- Having received the request, the Web server responds to the requests and re-directs part of the request to other appropriate resources such as Web Feature Server to get GML file.
- Once the server has obtained the GML file from the re-directed server: web feature server, it then makes a reply in the form of GML file to the client.
- The client converts the GML file to SVG and HTML through the GIS component in which is installed prerequisite.
- The client browser displays the received HTML documents and SVG files.

HTML is a document formatting language that cannot be used to carry out even simple calculations such as additions. To perform the GIS operations in the client machine, the Web browser must be assisted with client-side scripting languages such as JavaScript and VBScript, Web-enabled programming

languages such as Java, and GIS components in form of a plug-in, an ActiveX or a JavaApplet.

Unlike server-based solution, client-based solution only send GML files to the client tier. Converting GML to SVG is realized in the client computer by a plug-in which also offers other functions such as query and analysis. In this case, the server's job is to extract the required GML data from the database and furnish them to the client. Once the GML data are on the client computer, the application is quite capable of manipulating the data locally. So the client tier contains not only the HTML information, but also the SVG data that support GIS operations in the client computer. The user can create complicated GIS operations in the client computer to manipulate the SVG data that usually offer more efficient and flexible data processing than the raster data format.

Despite the good features mentioned above, there are several disadvantages exposed by the client-side solution that need to be remedied. First, the security issues related to the use of vector data in the client computer must be addressed so that the ownership and the copyright of the vector data will not be abused. Second, the frequency of vector data transmission

between server and client must be optimized in order to reduce the high usage of bandwidth in the Internet.

5 CONCLUSIONS

GML is open standard XML-based geodata format specifically designed for the web. It is available to all application developers, and users will not be locked to a single vendor. Without such standards, supported by Internet browsers, the vector solution will always require ActiveX or applet plugins such as, ArcIMS, GeoMedia Web Map, Map Guide Internet-GIS software. The user who applies other GIS data formats need to buy expensive GIS software to display and manipulate the geodata. Alternatively the user can choose GML as the geodata format to solve the problem. With GML, handling spatial data and attributes become open and accessible to any user and any data provider.

Because GML is XML based, it is straightforward to take data in GML format and use them to generate SVG graphics. This can be done statically or dynamically on the server or in the browser. GML is easy to be understood and edited since it is text-based. GML also provides a useful long-term archive format for geospatial data, which reduces the risk that new versions of data format make old datasets unreadable.

Since SVG works with common technologies (e.g. DOM, JavaScript), it is easy to make the SVG file dynamic and deal with the relationship of features and attributes, such as showing the attributes of features, showing the coordinates of the cursor, and highlighting the moused over feature. Since DOM is integrated with SVG, these functions on SVG object can be realized in real time without reload. But the user has to install the SVGViewer to view the SVG files. Hopefully the next version of the Internet Explorer will be able to interpret SVG.

Since GML and SVG are new technologies, application of GML is not widely adopted or supported. OGC Web Feature Server and other GML data servers have not provided many available data. Until now most products of GML mainly focus on rendering the geodata; some other functions such as feature editing, spatial analysis and client-side manipulation have not been realized yet. GML 3.0 supports Topology and 3D geometry and makes it possible to do spatial analysis, 3D visualization and analysis with GML3.0. Although GML provides a solution for manipulating geodata in GIS, its future role in this field is not clear.

ACKNOWLEDGMENT

Acknowledgment is due to Prof. Dr. -Ing. Dietrich Schröder, Department of Surveying and Geoinformatics of the University of Applied Sciences in Stuttgart, Germany.

REFERENCES

- [1] Houlding, S. W., 2001. XML-an opportunity for <meaningful> data standards in geosciences.[J] Computers & Geosciences 27 (7), 839-849.
- [2] Badard, T. and Richard, D. 2001. Using XML for the exchange of updating information between geographical information systems.[J] Computers, Environment and Urban Systems 25 (1) 17-31.
- [3] Michael J.Y., 2002. XML - Schritt für Schritt: [schnell und einfach lernen].[M] 2.ed Microsoft Press Deutschland, 489
- [4] Benoit M., 2000 Applied XML solutions. [M] Sams, Indiana, U.S.A., 360P
- [5] Sheng Ye, Feng Xuezhi, Shaotao Yuan, Li Juliang. Visualization GML with SVG. [C]IGARSS 2005 3648-3651

FURTHER READING

- [1] W3C, XML, <http://www.w3.org/XML/>
- [2] W3C, XML Schema, <http://www.w3.org/XML/Schema>
- [3] OGC, Geography Markup Language (GML) WG (GML WG), <http://www.opengeospatial.org/groups/?iid=31>
- [4] OGC, Geography Markup Language (GML) RWG (GML RWG), <http://www.opengeospatial.org/groups/?iid=48>
- [5] W3C, Scalable Vector Graphics (SVG), <http://www.w3.org/Graphics/SVG/>
- [6] W3C, XSL Transformations(XSLT) version 2.0, <http://www.w3.org/TR/xslt20/>
- [7] Ionic Software, GML, SVG, <http://www.ionic.be>
- [8] Max Dunn, Silicon Publishing, SVG Frequently Asked Questions <http://www.svgfaq.com/>
- [9] The Web Academy, SVG resource, <http://www.webacademy.com/svg/>
- [10] carto.net, Vector-based Web Cartography: Enabler SVG, http://www.carto.net/papers/svg/index_e.shtml
- [11] W3C, Resource Description Framework (RDF) <http://www.w3.org/RDF/>
- [12] Galdos Systems Inc., 2000. Making Maps With GML <http://www.galdosystems.com/files/MakingMapsInGML2.pdf>
- [13] Galdos Systems Inc., 2001. GML2.0 Enabling the Geospatial Web <http://www.galdosystems.com/files/GML2-PoweringTheGeoWeb.pdf>
- [14] Galdos Systems Inc., 2001. Location-Based Services & GML <http://www.galdosystems.com/files/LocationBasedServices.pdf>

