This article originally appeared in the July 2001 issue of *Geospatial Solutions* (www.geospatial_online.com) and is reprinted with permission.

GML 2.0

Enabling the Geospatial Web

Ron Lake

Adopted by the Open GIS Consortium as well as leading database, GIS, and application developers, GML 2.0 is ready for prime time. Can GML's ability to express relationships between geospatial entities across the Internet help your operation?

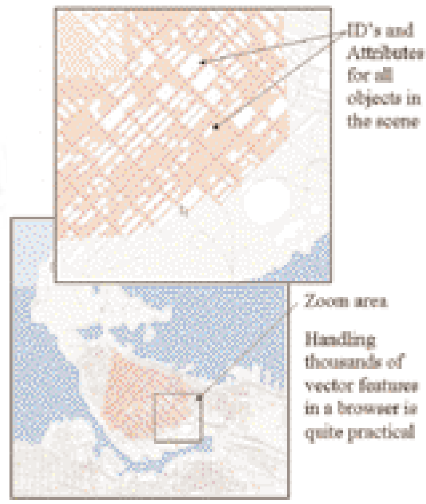

FIGURE 1 GML data can be readily styled into scalable vector graphics for presentation.

On February 20, 2001, the OpenGIS Consortium published version 2.0 of the Geography Markup Language, thus laying the foundations for the development of a geospatial World Wide Web. Since the publication of GML 1.0 in May 2000, interest in GML has developed rapidly, and organizations and individuals worldwide are now pursuing GML technology development.

This article is intended to help you understand what GML 2.0 is all about, how it came into being, and how it may transform the spatial data industry.

Building on XML

Like its predecessor, GML 2.0 builds on the evolving world of XML, a technology that has impacted almost every area of information processing. XML is a means of encoding data in text. Although it started out as a means of marking up a document for selection and presentation, it has quickly evolved into a mechanism for general data description. Today, XML is used in a variety of industries and domains including finance, chemistry, Internet business-to-business, document publishing, multimedia, telecommunications, graphics, and process control to name only a few.

XML has become very popular, in part, because it can be read and understood by developers and data managers alike. This has helped to make XML-based systems more flexible and adaptable to changing requirements and conditions. XML is also easily transformed, and the plethora of XML grammars for different domains makes it relatively easy to integrate and combine XML-based data from many disparate sources.

XML today is not a single technology but a family of technologies derived from core XML encoding (specified in XML 1.0). This family comprises technologies and standards for data modeling as well as simple semantic expression. These include such World Wide Web Consortium standards as the RDF schema definition, XSD, linking and association schema (XLink), element selection and pointing, service description (Web services description language, for instance), distributed computing (such as simple object access protocol), and graphical display (as in scalable vector graphics). The rich set of XML standards is giving rise to robust XML tools and components that can be leveraged in the deployment of any domain specific technology that builds on XML.

Not your father's HTML. The comparison between XML and HTML is a natural one and early hype centered around XML as a replacement for HTML. This is of course misleading. Therefore, it might be appropriate to first differentiate between the two languages.

XML is a language for writing markup, or data encoding languages and grammars, and in fact, HTML has been written in XML (XHTML). Of particular interest in the GML context, however, is the approach that one would take in creating a Web page in an XML environment rather than an HTML-only environment.

An HTML document describes the content (the actual text) and the layout (for instance, which text gets bolded and what type face is used in a given area)

Glossary

DTD: Document type definition

GML: Geography markup language

HTML: Hypertext markup language

OGC: Open GIS Consortium

RDF: Resource description framework

XLink: EXtensible linking language (also called XLL)

XML: Extensible markup language

XSD: XML schema

Ron Lake is founder, president, and CEO of Galdos Systems, Inc. (www.galdos.com), the primary authors of GML. He has been involved with information technologies for 30 years and is currently active in the Open GIS Consortium.

in a single language, namely HTML. XML takes the opposite approach and separates content from presentation. An XML grammar for a document will describe the document structure without regard to its eventual appearance in print or on a computer screen. The XML tags that "mark up" the document will define where the title is, the start and end of each chapter, the prolog and so forth. In XML terms, the document then needs to be styled to be viewed. This involves assigning specific fonts, colors, sizes, and screen locations to the different parts of the document. Often this is done by generating HTML.

The XML approach of separating presentation from content is not something fundamentally new, but is nonetheless key to many XML applications. In the world of the Internet, it allows a Web page's appearance to be more readily tailored to a specific type of user or display device.

New grammar. GML can be thought of as the XML for the geospatial domain. It provides a grammar for encoding geospatial content (feature properties, location, extent, feature relationships) such as rivers, roads, churches, and political boundaries. A GML 2.0 encoding of a road segment might look something like this

```
<uka:Road fid ="highway11"
   <uka:numLanes>3</uka:numLanes>
   <uka:surfaceType>gravel</uka:surfaceType>
   <gml:centerLineOf>
      <gml:LineString srsName = "epsg4361">
         <gml:coordinates> . . . . </gml:coordinates>
      </gml:LineString>
   <gml:centerLineOf>
<uka:Road>
```

Because GML is concerned with the description of geographic content, it must be styled for presentation. Presentation may mean being styled to a graphical form such as a map (see Figure 1), but it could also involve being styled to text or even to a sequence of voice instructions.

Linking pages and elements

Another key difference between XML and HTML comes in the ability to link pages and elements. HTML provides a simple form of linking one Web page to another. This linking mechanism is one of the key foundations of the Web. The link is established through an anchor or bookmark embedded in the target page and a link reference embedded in the source page. Note that such a link associates only two resources (the source and target pages) and it does so in a unidirectional manner (source to target). Further, the HTML link is a coarse-grained mechanism. It only allows users to point to complete Web pages and only to single points in those pages.

XML goes much further. It provides a mechanism for linking multiple resources into a complex association. Its links can also be traversed in both directions. XML further enables fine-grained associations to be constructed. It can associate single XML elements or even element fragments. This has profound implications for GML's ability to build associations between spatial features.

Since XML separates presentation and content, XML technologies have been developed for style transformation. These are now available for a wide variety of devices from the desktop to handheld and wireless PDAs. The ubiquity of XML has other implications for GML as well. With more types of data being expressed each day in XML, the ability to combine and associate geospatial data with hundreds of other data types, a long-time objective of the geospatial community, moves closer to reality.

Ready for prime time with GML 2.0

GML 1.0 was based on a combination of XML DTDs and RDFs. This was an awkward but useful combination. DTDs were in widespread use, but lacked the ability to support type inheritance, had no underlying semantic model, and did not support namespaces. RDF, on the other hand, was less accepted but offered namespace support, distributed schema integration, type hierarchies, and a simple semantic model. It was at best an awkward combination.

GML 2.0 is based entirely on XSD. OGC's adoption of XSD is a major advance. XSD has matured greatly in the past year and now incorporates support for type inheritance, distributed schema integration, and namespaces. Moreover, a great variety of tools and parsers now support XSD and more are anticipated in the near future.

GML 2.0 provides a single encoding method (XSD) and a single approach to the creation of feature schemas, replacing GML 1.0's three different profiles — referred to as GML.1, GML.2, and GML.3. Such profiles were somewhat awkward as they overlapped different encoding methods

FIGURE 2 With GML, developers can build feature-type families without concern for feature-type name conflicts.

(XML 1.0 [DTD and RDF]) with different approaches to the encoding of schemas.

The evolution to GML 2.0, though, did build on the successes of GML 1.0. For instance, though GML 1.0 Profile 1 offered no means for feature schema expression independent of the data instance itself, Profile 2 provided more schema support through user-defined DTDs. GML developers continued this approach in 2.0 through user defined XSDs.

With Profile 2, GML 1.0 users could define their own feature types using XML DTDs. Profile 2 instances were easily recognized by the presence of feature types as XML elements or tags. Note, however, that there was no namespace support and no notion of type hierarchies.

Some of these restrictions were lifted in GML 1.0 Profile 3, which very closely resembles GML 2.0. The use of namespace prefixes allows users to create specific vocabularies based on their organization or on the domain, or information community, of interest.

GML 2.0 takes us even further. As in GML 1.0 Profile 3, namespaces can be exploited to create different vocabularies or feature-type families. Moreover, users can take advantage of type inheritance and distributed schema support to build feature-type families from one another (as shown in Figure 2) without concern for feature-type name conflicts.

Figure 2 shows three vocabularies. One is a set of basic definitions for features and geometry (common geospatial vocabulary), while the other two provide specific feature types for the forestry and environment domains. Note that by using namespace prefixes, each of the domain vocabularies shown in the figure can define the same feature types without concern for name conflicts. Each can, for example, contain its own

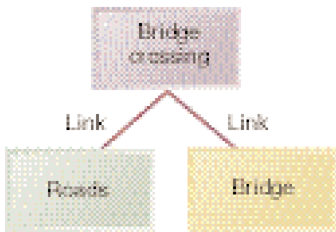

FIGURE 3 GML links three distinct databases, using the bridge crossings to define associations between bridges and the roads they carry.

notion of road using the name "road," (<env:Road> and <for:Road>) and users of these schemas can clearly distinguish one from the other by means of the namespace prefix. Thus GML 2.0 provides the basic definitions and mechanisms for building a distributed hierarchy of feature types, laying the foundations for the geospatial Web.

Building distributed relationships

The real world around us is one of relationships — buildings front onto streets, streets intersect one another, and animal habitat zones depend on the occurrence of specific plant species. In the past, some GISs provided support for feature relationships, but these were restricted in their expressive capability and were not suited to relationships that are distributed over the Web. Some were restricted to topological relationships. GML 2.0 changes all of this.

It makes use of the XLink and XPointer specifications to express relationships between geospatial entities. Consequently, such relationships can be expressed between features in the same database or between features across the Internet. Furthermore, GML 2.0 allows for relationships to be constructed between GML feature elements in different databases without modification of those participating databases. No more than read access is required to establish a relationship.

The Internet itself was built on the ability of HTML to express linkages between widely distributed Web pages. GML 2.0 takes this simple concept further by providing linkages between widely distributed geospatial features. For example, Figure 3 shows three GML data stores. One of these is a database of GML road features, while another is a database of GML bridge features. These two databases are assumed to be developed and maintained by separate organizations and to be physically distinct. The third database, that of bridge crossings, is in effect a database of links defining associations between the bridges and the roads that they carry.

Relationships in GML 2.0 can themselves be treated as GML features and, hence, can have their own properties in addition to expressing the associations between distinct features. This might be the case, for example, for a bus route, a traffic intersection, or a highway interchange. In addition to expressing simple binary relationships using in-line encodings, GML 2.0 can also express complex relationships involving multiple distributed resources.

What's under the hood?

So how does GML work? As noted, GML is an XML encoding that is based on XSD. In fact, we can view GML itself as a schema writing language for spatial information, that provides

- a set of standard XML tags or elements for such things as spatial features and a variety of geometry types;
- a data model based on the notion of a spatial feature and its associated properties that derives from the OGC Abstract Specification. (Each GML feature type is an identified list of properties that characterize that type. A road, for example, is determined by its list of properties including number of lanes, classification, and surface type.);
- a set of rules for constructing GML schemas.

Some GML tags provide specific functionality (geometry tags), others identify the kind of object you are creating (such as an abstract feature), and some provide convenient holders for things like feature names and descriptions.

GML is organized into three schemas, namely Feature, Geometry, and Xlinks. The Feature schema provides the means to create a feature (a geographic entity like a road or river) as a list of properties, some of which may be geometric (such as location or extent). It also provides a set of common geometry properties that might be used to describe a feature such as location, edgeOf or extentOf.

The Geometry schema provides a set of elements for common geometry types including point, linestring, polygon, and a set of aggregations of these simple types. Using GML, one can describe the geometry of a lake (including the islands inside it), a national park, or a superhighway.

The Xlinks schema provides the attributes to enable links between GML features or GML geometries. Such links represent semantic associations between the features or geometries. This allows us to build the distributed feature associations discussed earlier.

To use GML, someone must create one or more GML schemas. These describe the geospatial features of interest by identifying their properties and geometric descriptions. This is similar to a database schema except that it can be more readily viewed on the Internet or e-mailed from one place to another. GML documents are data fragments built to conform to these schemas — the schema acting much as a data template. In fact, the ability to use schemas in a distributed environment means that we can share the description of geographic features across whole communities such as environmental planning, transportation, or mining.

Enabling interoperability

Within the OpenGIS Consortium, work is underway on various specifications that are critical to the future development of distributed spatial systems (see "GML, OGC, and Information Communities" sidebar). These include interfaces for requesting geospatial features, describing map styles, requesting and generating maps, invoking feature coordinate transformations, defining and requesting coordinate transformations, geocoding and Gazetteer requests, and image and map annotation. Each of these specifications is itself dependent on GML 2.0.

GML 2.0 supports geospatial interoperability in various ways. First, it provides a common schema framework for the expression of geospatial features. While GML builds on XSD, it provides a more constrained model for expressing a geospatial feature type in terms of the properties that characterize that feature type. This means that one can readily compare features by looking at their corresponding feature schemas.

GML further supports interoperability by providing a common set of GML geometry types. Although two different schema

GML, OGC, and Information Communities

Almost every interoperability interface in OGC builds on previously developed interfaces that enable spatial technologies. GML enhances the value of previously developed OGC Simple Features and Coverages, Web Map Server, Coordinate Transformation Services, and Catalog Services interfaces. What will come next from OGC? Interfaces to address a new paradigm of interoperability — information communities.

Described in OGC's 1995 *OpenGIS Guide*, implementation of information communities has awaited the foundation provided by GML and underlying technologies. The information community concept addresses the most difficult interoperability issue of geodata semantics. This issue has been a major focus of the Federal Geographic Data Committee, state GIS councils, and countless other data coordination groups. The OpenGIS Guide defines these communities.

"An Information Community is a collection of people (a government agency or group of agencies, a profession, a group of researchers in the same discipline, corporate partners cooperating on a project, and so forth) who, at least part of the time, share a common digital geographic information language and share common spatial feature definitions. This implies a common world view as well as common abstractions, feature representations, and metadata. The feature collections and geoprocessing functions that conform to the Information Community's standard language, definitions, behaviors, and representations belong to that information community.

"The OpenGIS Information Communities model helps solve the human problem of communication between communities who, by necessity or chance, describe geographic features in different ways. Different Information Communities find it difficult to query each other's data, difficult to interpret the return sets, difficult to interpret the quality statements, and difficult to merge the return sets with information from their native community."

Overcoming the interoperability issues caused by semantic differences will depend on data coordination efforts, as it has in the past. But during the next two years, geospatial technology providers cooperating in OGC will build a framework for tools that will both support data coordination and automate and optimize the sharing of data between information communities.

On March 23, 2001, OGC issued a request for technology in support of an OGC Web Services Initiative. Part of this initiative is a four-stage Information Community Enablement (ICE) initiative, now moving forward with a proposal for a feasibility study. The ICE Testbed is expected to result in a prototypical Internet-based system for enhancing intra- and intercommunity data sharing.

Information communities will register themselves. The registry will contain metadata about each registered community. Each community will need to reach consensus on a common taxonomy of types and subtypes.

Communities will use a standard encoding scheme to enter their Type Dictionaries, which are records of the spatial data types and vocabularies used in that community's spatial data application schemas. These types and vocabularies are also used to encode the metadata of data type schema, and to encode the metadata of services. Any Internet service that sends a request to an information community registry should be able to resolve such encoding, which will be done using GML.

It is proposed that a section of the OGC Network (www.ogcnetwork.org) Web site will serve as a "friendly, well-lit place" where information communities can register their type dictionaries and application schemas, or otherwise browse, exploit, or download those that are created by other communities.

Once two or more information communities have registered their type dictionaries, attention will be turned to "cross-walks" from a node in one taxonomy to a node in another. The next step will be to develop a GML-based semantic translator that a data coordination group for an information community can configure to translate and filter data.

For more information, contact Mark Reichardt at mreichardt@opengis.org

— **Mark Reichardt**, Director, Marketing and Public Sector Programs, Open GIS Consortium, Inc.

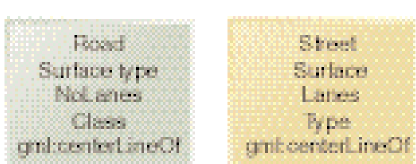

FIGURE 4 Though different schema authors might model a road differently, in GML, they can share mechanisms for describing geometry.

authors might model a road in different ways, they can share the same mechanisms for geometry description and can very likely interpret the correspondence between the schema.

Figure 4 shows two classes, one describing a road and the other describing a street. The properties of these two schemas are clearly different, although they have a common geometry description, achieved by each author using the common geometry property <gml:centerLineOf>.

The definition and publication of GML schemas that can be shared across communities of interest such as transportation, environmental issues, petroleum exploration, and land management, opens the door to interoperability on the semantic level. This is beginning to happen on both a regional and national level and we can expect to see a variety of domain-specific GML application schemas published over the next year.

The evolving geospatial Web

GML's stage of maturity enables the construction of real spatial datasets, the interchange of spatial information, and the construction of distributed spatial relationships. GML 2.0 will significantly impact the geospatial industry, most specifically in the area of location-based services.

Many leading corporations are working today on the development of GML. Consequently, a range of GML products should appear in the near future including Web feature servers, Web coverage servers, geocoders, and coordinate transformation services. Data conversion companies will soon make available conversion engines that provide read/write support for GML.

GML 3.0, slated for this fall, will offer many enhancements while retaining backward compatibility with GML 2.0. Features to look for include topology support, new geometry classes, events, histories and feature time stamps, units of measure, metadata, and coverages.