



GML 2.0

Enabling the Geo-spatial Web

Galdos Systems Inc.

March 14, 2001

Contact: Ron Lake
Phone: (604) 687-7311
E-mail: rlake@galdosinc.com

TABLE OF CONTENTS

1. INTRODUCTION	1
2. BUILDING ON XML	1
3. READY FOR PRIME TIME	4
4. BUILDING DISTRIBUTED RELATIONSHIPS	6
5. ENABLING GEO-SPATIAL INTEROPERABILITY	8
6. THE FUTURE OF GML	9

1. INTRODUCTION

On February 20, 2001, the OpenGIS Consortium (OGC) published Version 2.0 of the [Geography Markup Language](#) (GML), thus laying the foundations for the development of a Geo-spatial world wide web. Since the publication of GML 1.0 in May 2000, interest in GML 2.0 has developed rapidly. Organizations and individuals in every corner of the globe are now pursuing GML technology development.

This article is intended to help you understand the potential impact of GML 2.0 on both the existing world of geo-spatial technology as well as on the emerging world of Location-based Services. While we will take a few peeks “under the hood”, this article is focused more on the implications of GML 2.0 than on its internal workings.

2. BUILDING ON XML

Like its predecessor, [GML 1.0](#), [Geography Markup Language 2.0](#) builds on the evolving world of XML technology, a technology that has impacted almost every area of information processing.

[XML](#) is a means of encoding data in text. A GML 2.0 encoding of a road segment looks something like the following.

```
<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>
```

XML technology today is extremely widespread. It is embedded in the browser on your desktop. It is the “lingua franca” of emerging e-business frameworks, and it powers the generation of thousands of web sites. Why has it become so successful? How did we move so rapidly from merely marking up documents for publication to using XML as a general tool for data description?

In part the explanation lies in the evolution of the Internet itself from an environment of distributed pages of information to one of distributed business

services. While this evolution is only in its infancy, the new Internet demands tools with greater expressive power and ones that integrate together many kinds of information. The Internet of distributed information relied mainly on text and imagery. The new Internet demands the ability to express the elements of automated business interaction, from invoices and purchase orders to currency and other types of financial transactions. Moreover the Internet has moved into every type of business and with that has come the need to express not only the financial aspects of business interaction but also the specialized contents of differing business domains.

The world of the Internet is also a world of information collision, and yearned for integration and fusion. Tap a search engine and you see information of often bewildering diversity. Information collides and we think of new ways to integrate and extend it. This in turn demands technologies that thrive on information integration rather than isolation.

In political terms the Internet has been a great leveller. E-mail reaches across the spaces of the globe and knows nothing of the boundaries between states, nor that between individuals. It fuses us with one another.

There is also a world of reality outside the Internet that especially in geo-spatial terms drives us toward data integration. Events in the world do not take place in isolation. Neither do they align themselves with the boundaries of government departments or provinces or states of national governments. A flood in El Salvador or an earthquake in India ripples around the world. The flood does not care that there is a Ministry of Forests, nor a Ministry of the Environment, or that the administration of one is not integrated with the other. The flood tears through the fabric of the country merging the trees and the soil and the water, mingling agriculture and industry, homes and social infrastructure. To respond to the inherent integration of the world we must integrate our information resources.

The explosion of the Internet has also demanded that our technologies be extensible and comprehensible. This was one of the lessons of HTML. A simple text based language that has come to dominate the world. Visibility and comprehensibility are increasingly demanded in a world that is already too complex.

The character of XML has in many respects been shaped by responses to these issues. XML like HTML is text based. It can easily be read and understood by human beings. Since it is text, XML can readily combine together a wide variety of data types including text, finance, graphics, audio, voice and more. This means that geographic data can readily be integrated with a wide range of non-geographic data types thus greatly enhancing the value and accessibility of spatial information.

XML technology has also evolved in response to the limitations of HTML. While enormously successful, HTML and the World Wide Web are not without

shortcomings. All of us are familiar with the “404” message displayed when a broken hypertext link points us to a non-existent page, or to the different appearances that can derive from viewing a single web page in different web browsers.

Where HTML mixes content and presentation together, XML strictly separates the two. XML, the encoding standard deals only with data structure. This simple fact liberates it from mere document description to become a general tool for data description. GML continues this fundamental idea.

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

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). Note further that the HTML link is a coarse grained mechanism. It only allows one to point to complete web pages and only to single points in those web pages.

XML goes much further. XML provides a mechanism for linking multiple resources into a complex association. XML links also can be traversed in both directions. XML further enables fine-grained associations to be constructed. Where HTML linking only supports the linking or association of web pages, XML linking can associate single XML elements or even element fragments. As we shall see, this has profound implications for GML’s ability to build associations between spatial features.

Since XML separates presentation and content, XML technologies have developed for style transformation. These are now available for a wide variety of devices from the desktop to hand held and wireless PDA’s.

The ubiquity of XML has other implications for GML. With more and more types of data being expressed each day in XML, the ability to combine and associate geo-spatial data with hundreds of other data types, one of the long objectives of the geo-spatial community, moves closer to reality.

3. READY FOR PRIME TIME

GML 1.0 was based on a combination of XML DTDs (Document Type Definition) and Resource Description Framework (RDF). 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 did offer namespace support, distributed schema integration, type hierarchies and a simple semantic model. While it was possible to more or less use all of these features it was at best an awkward combination.

GML 2.0, which replaces GML 1.0, is based entirely on XML Schema (October 14, 2000). The adoption of XML Schema (XSD) is a major advance. XML Schema has matured greatly in the past year and now incorporates support for type inheritance, distributed schema integration, and namespaces. Moreover there are now a great variety of tools and parsers that support XML Schema and more are anticipated in the near future.

GML 1.0 offered three different profiles that were referred to as GML.1, GML.2 and GML.3. Such profiles were also somewhat awkward constructs in GML 1.0 as they overlapped different encoding methods (XML 1.0 (DTD) and RDF) with different approaches to the encoding of schemas. GML 2.0 provides a single encoding method (XML Schema) and a single approach to the creation of feature schemas. A simple example illustrates the difference between GML 1.0 and GML 2.0.

```
<Feature typeName="Road">
  <description>Georgia Street</description>
  <property typeName="numberLanes" type="integer">4</property>
  <geometricProperty typeName="linearGeometry">
    <LineString srsName="EPSG:4326">
      <coordinates> 0.0,100.0 100.0,0.0 </coordinates>
    </LineString>
  </geometricProperty>
</Feature>
```

Figure 2.0 Example GML 1.0 Profile 1 (GML.1) Feature Instance

Note that this example makes no use of namespaces (namespaces were not supported in DTDs) and that the feature type is not actually defined. 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 and this approach is continued in GML 2.0 through user defined XML Schemas. A GML 1.0, Profile 2. instance looked as follows:

```
<Road>
  <description>Georgia Street</description>
  <numberLanes>4</numberLanes>
  <centerLineOf>
    <LineString srsName="EPSG:4326">
      <coordinates>0.0,50.0 0.0,100.0 </coordinates>
    </LineString>
  </centerLineOf>
</Road>
```

Figure 3.0 Example GML 1.0, Profile 2 (GML.2) Feature Instance

With Profile 2, GML 1.0 users were able to 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 (e.g. <Road> above). 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., and Profile 3.0 instances very closely resemble those of GML 2.0. The same road in Profile 3. (GML .3) looked as follows:

```
<os:Road>
  <gml:description> Georgia Street </gml:description>
  <os:numberLanes>4</os:numberLanes>
  <gml:centerLineOf>
    <gml:LineString srsName="EPSG:4326">
      <gml:coordinates>0.0,100.0 100.0,0.0</gml:coordinates>
    </gml:LineString>
  <gml:centerLineOf>
</os:Road>
```

Figure 4.0 Example GML 1.0, Profile 3 (GML.3) Feature Instance

Note that in this profile user schemas are again supported as is obvious from the <os:Road> element name. The use of namespace prefixes (the os in front of Road) allows users to create specific vocabularies based on their organization or on the domain or information community of interest. With namespace support we can clearly distinguish <os:Road> from <usgs:Road> or <nrcan:Road> .

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 we can use type inheritance and distributed schema support to build feature type families from one another as shown in Figure 5.0 without concern for feature type name conflicts.

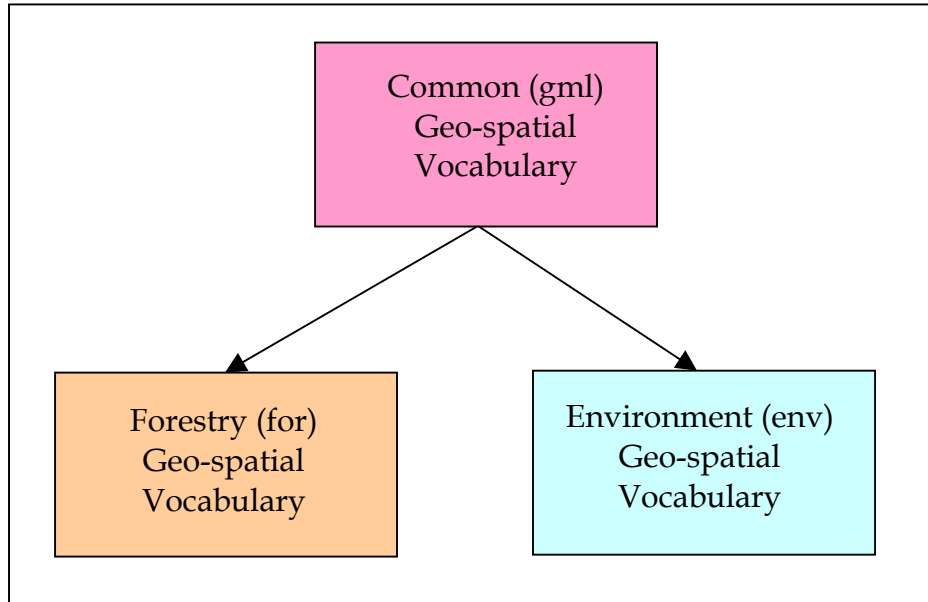


Figure 5.0 Building Feature Type Vocabularies

Figure 5.0 shows three vocabularies. One is a set of basic definitions for features and geometry (Common Geo-spatial Vocabulary), while the other two provide specific features types for the Forestry and Environment domains. Note that by using namespace prefixes, each of the domain vocabularies shown in Figure 5.0 can define the same feature types without concern for name conflicts. Each can, for example, contain its own notion of road using the name Road, and users of these schemas can clearly distinguish one from the other by means of the namespace prefix. (From Figure 5.0 we would have <env:Road> and <for:Road>.)

GML 2.0 provides the basic definitions (as shown in Figure 5.0) and the mechanisms for building a distributed hierarchy of feature types. It thus lays the foundations for the Geo-spatial Web. GML 2.0 is ready for prime time.

4. 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 GIS systems have provided support for feature relationships but these have been restricted in their expressive capability and they have not been suited to relationships that are distributed over the

Internet. Some were restricted to simply topological relationships. GML 2.0 changes all of this.

GML 2.0 makes use of the [XLink](#) and [XPointer Specifications](#) to express relationships between geo-spatial entities. This means that such relationships can be expressed between features in the same database or between features across the Internet. Furthermore, GML 2.0 allows relationships to be constructed between GML feature elements in different databases without requiring any modification of the 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 geo-spatial features.

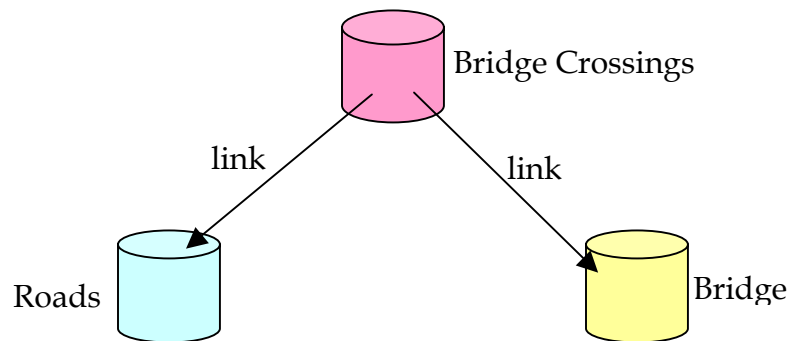


Figure 6.0 GML 2.0 supports distributed feature relationships

Figure 6.0 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. While GML 2.0 can readily express simple binary relationships using in-line encodings, it can also express complex relationships involving multiple distributed resources.

As HTML was critical to the development of the Internet as a linked collection of web pages, GML 2.0 will enable the development of a geo-spatial “Internet” as a linked collection of geo-spatial features.

5. ENABLING GEO-SPATIAL INTEROPERABILITY

Within the OpenGIS Consortium, work is underway on a number of specifications that are critical to the future development of distributed spatial systems. These include interfaces for:

- Requesting geo-spatial features.
- Describing map styles.
- Requesting maps and map generation.
- Invoking feature coordinate transformations.
- Definition of and request for coordinate transformations.
- Geo-coding and Gazetteer requests.
- Image and map annotation

Each of these specifications is itself dependent on GML 2.0. GML 2.0 is thus playing a critical role in enabling geo-spatial interoperability.

GML 2.0 supports geo-spatial interoperability in a number of ways. The first is that GML provides a common schema framework for the expression of geo-spatial features. While GML builds on XML Schema it provides a more constrained model for expression of a geo-spatial 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. While two different schema authors might for example model a road in different ways they can share the same mechanisms for geometry description and it is then very likely that one can interpret the correspondence between the two schemas. This is illustrated in Figure 7.

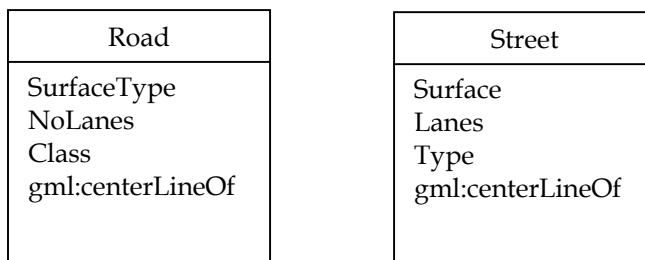


Figure 7.0 Simple Street or Road UML Model

Figure 7.0 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 `gml:centerLineOf` geometry property. GML assures geometry level interoperability.

6. THE FUTURE OF GML

With [GML 2.0, Geography Markup Language](#) has reached a stage of maturity that enables the construction of real spatial datasets, the interchange of spatial information and the construction of distributed spatial relationships. We anticipate that GML 2.0 will have a significant impact on the geo-spatial industry and most importantly in the domain of location-based services.

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

GML is moving forward to enable the geo-spatial web.