

Take a quick 5 minute survey for a chance to win a **\$500**
American Express Gift Card!

[CLICK HERE TO BEGIN](#) ▶

Home

- [Subscribe to GeoWorld](#)
- [GeoReport Newsletter Sign Up](#)

News

- [Articles & Archives](#)
- [News](#)
- [Submit Press Release](#)
- [Product of the Week](#)
- [Press Releases](#)
- [GeoReport Archives](#)
- [Top News Stories](#)
- [RSS](#)
- [Industry Interviews](#)
- [Online Exclusives](#)
- [Feature Articles](#)

My GeoPlace

- [Blogs](#)

GeoTec Event

- [GLA Award Winners](#)
- [Map Gallery Winners](#)
- [Program](#)
- [Workshops](#)
- [Organizing Committee](#)
- [Contacts](#)

Store

- [Bookstore](#)

Contact

- [Editorial](#)
- [Sales](#)
- [Advertising](#)
- [List Rental](#)
- [Reprints](#)
- [Web](#)
- [Info Request Form](#)

Flip Book

- [Help Page #01](#)
- [Help Page #02](#)

Articles/Archives

Issue Date: March - 2006, Posted On: 3/1/2006

Time Adds Another Dimension to GML

GMLDEVELOPMENT

Time Adds Another Dimension to GML

BY RON LAKE

Ron Lake is president, Galdos Systems Inc.; e-mail: rlake@galdosinc.com.

In the "GIS world," there's much talk about spatial data, but most of the problems involve spatio-temporal information. Spatial objects typically evolve through time, and temporal evolution is critical in almost all application domains, ranging from disaster management (e.g., tracking a hurricane or tsunami path) to national security (e.g., vehicle, container or human movements) and air-traffic management.

Until recently, most GIS products provided little in the way of temporal models and nothing in terms of normative representations for expressing time or the evolution of geographic features. However, Geography Markup Language (GML) provides all of these elements and more. In particular, GML supports the following:

- Standard temporal encodings from XML Schema (e.g., time, dateTime)
- Rich, temporal encodings
- User-defined clocks and calendars, including ordinal calendars
- User-defined temporal reference systems
- The ability to attach temporal properties (e.g., gml:validTime) to user-defined geographic features in a sequence of TimeSlices, with each TimeSlice incorporating the feature's time-varying properties

Temporal properties of geographic objects can be exploited in queries to OGC Web Feature Services (WFSs), thus enabling the retrieval of features or parts of features that satisfy specific temporal constraints. This capability will be further extended in WFS 1.2 (this also will be ISO 19142) with the introduction of finer GML object operations and temporal operators.

Temporal Examples

To understand the temporal capabilities of GML, it's best to consider a few simple examples.

Suppose someone has an interest in a land parcel, and they want to see when the tenure is effective. This can easily be done by creating a property in the application



1 Spatial
Announces
Acquisitions
in Ireland
and Scotland



Introducing GeoEye,
the largest commercial
remote sensing
company in the world.



schema that uses the content model `gml:TimePrimitivePropertyType` or by using the property `gml:validTime`. An instance would look something like:

```
<abc:Tenure gml:id="t234">
  <abc:extent>
    <gml:Polygon id="p1">
      <gml:exterior>
        <gml:LinearRing>... </gml:LinearRing>
      </gml:exterior>
    </gml:Polygon>
  </abc:extent>
  <gml:effectiveDate>
    <gml:TimeInstant>
      <gml:timePosition>2005-12-13T00:00:00-08:00</gml:timePosition>
    </gml:TimeInstant>
  </gml:effectiveDate>
</abc:Tenure>
```

Note that GML can share "time objects" among features just as it does "geometric objects." The aforementioned `TimeInstant` object represents a moment or point in time, but GML also can represent time intervals or durations.

Suppose that a building was constructed at a certain date and remodeled or removed at a later date. In this case, the building may have a `lifeTime` property in which the value is a `gml:TimePeriod`.

This `TimePeriod` object represents a duration or interval in time from `beginPosition` to `endPosition`. `TimePeriods` can be encoded in several different ways in GML to allow users to tradeoff between simplicity and expressiveness.

`TimeInstant` objects shared among several geographic features, for example, could bound a `TimePeriod`. Alternatively, a user may be interested only in the temporal duration of the `TimePeriod` and not the specific start and end positions.

In this example, we are concerned with a building that was constructed at a certain date and which was remodeled or removed at some later date, perhaps by one of those spectacular demolitions, or by the impact of a natural or man made disaster. In this case, the building may have a `lifeTime` property whose value is a `gml:TimePeriod` as in the following code fragment:

```
<abc:Building gml:id="b11">
  <abc:extent>
    <gml:Polygon id="p1">
      <gml:exterior>
        <gml:LinearRing>... </gml:LinearRing>
      </gml:exterior>
    </gml:Polygon>
  </abc:extent>
```

```

        </gml:Polygon>

    </abc:extent>

    <abc:height>40</abc:height>

    <abc:lifeTime>

        <gml:TimePeriod>

            <gml:beginPosition>2005-12-13</gml:beginPosition>

            <gml:endPosition>2105-12-13</gml:endPosition>

        </gml:TimePeriod>

    </abc:lifeTime>

    □□□□□□□□ </abc:Building>

```

Take a Slice

A variety of object properties can change through time. It can be a building's height, a vehicle's speed and direction, an impact zone's extent, an ore body's depth or the volume of gas remaining in a gas field. GML represents such time-varying properties in terms of a TimeSlice.

A user-provided application schema defines the set of properties in a TimeSlice. For example, in a building, the height (number of stories) and number of rooms can be time varying, while the building's extent ("footprint") is time invariant.

In this instance, the history property encloses a sequence of TimeSlices derived from GML TimeSlice types and the inherited gml:validTime property. User-defined properties provide the height and number of rooms in each TimeSlice.

A building represented as a GML DynamicFeature might then look like:

```

    <abc:Building gml:id="b1">

        <abc:extent>

            <gml:Polygon id="p1">

                <gml:exterior>

                    <gml:LinearRing>... </gml:LinearRing>

                </gml:exterior>

            </gml:Polygon>

        </abc:extent>

        <gml:history>

            <abc:TimeSlice>

                <gml:validTime>

                    <gml:TimeInstant>

                        <gml:timePosition>2005-12-13T00:00:00Z

    </gml:timePosition>

                    </gml:TimeInstant>

```

```

        </gml:validTime>

        <abc:height>20</abc:height>

        <abc:numRooms>250</abc:numRooms>

    </abc:TimeSlice>

    <abc:TimeSlice>

        <gml:validTime>

            <gml:TimeInstant>

                <gml:timePosition>2005-11-
11T11:11:11Z</gml:timePosition>

            </gml:TimeInstant>

        </gml:validTime>

        <abc:height>40</abc:height>

        <abc:numRooms>480</abc:numRooms>

    </abc:TimeSlice>

</gml:history>

</abc:Building>

```

Moving Objects

Because the case of a moving object is common, GML provides a dynamic feature (a `gml:track`) with each `TimeSlice`: a `MovingObjectStatus` that has the properties of `gml:validTime`, `gml:location`, `gml:speed`, `gml:bearing`, `gml:acceleration` and `gml:elevation`.

A sample instance would look like:

```

<abc:Truck gml:id="kjh">

    <gml:description>Large freight truck</gml:description>

    <abc:manufacturer>Freightliner</abc:manufacturer>

    <abc:model>Coronado</abc:model>

    <abc:year>1998</abc:year>

    <gml:track>

        <gml:MovingObjectStatus>

            <gml:validTime>2005-12-13T00:00:00Z </gml:validTime>

            <gml:location>

<gml:Point>

<gml:pos>100 200</gml:pos>

</gml:Point>

```

```
</gml:location>

    <gml:speed>40</gml:speed>

    <gml:bearing>

        <gml:CompassPoint>NE</gml:CompassPoint>

    </gml:bearing>

<gml:MovingObjectStatus>

<gml:MovingObjectStatus>

    <gml:validTime>2005-12-13T01:00:00Z </gml:validTime>

    <gml:location>

<gml:Point>

<gml:pos>200 200</gml:pos>

</gml:Point>

</gml:location>

    <gml:speed>50</gml:speed>

    <gml:bearing>

        <gml:CompassPoint>E</gml:CompassPoint>

    </gml:bearing>

    <gml:MovingObjectStatus>

</gml:track>

<abc:Truck>
```

GML offers a flexible and powerful set of encodings for time and the temporal evolution of geographic features. Perhaps it's "time" to embrace this world of change.

Author's Note: For more details about GML, see Geography Markup Language—Foundation of the GeoWeb, published by John Wiley and Sons, or consult my blog at geoweb.blog.com.

Comment Box

You must sign in to post

Username

Password

No Blogger account? Sign up here.

Publish Comments

[Back to Top](#)

[Bookmark this Site](#)

Copyright © 2007 GeoPlace All rights reserved. | [Legal Terms](#) | [Console Login](#)