GML Recommendation and Analysis Paper

Expressing great circles and rhumb lines

By Galdos Systems Inc under contract with the United States Federal Aviation Administration (FAA)

Expressing great circles and rhumb lines

Version 0.2 | September 9, 2005 | Ron Lake, Galdos Systems Inc.

Summary of issue

The aeronautical domain requires the use of a number of special curves including:

- Great Circles
- Geodesics
- Rhumb lines or loxidromes

These types of special curves are currently identified in the AICM E-R model (according to Brett).

Several possible modeling approaches are considered:

- Create new geometry types (e.g. derived from gml:Curve)
- Use LineStrings or other specific curves in particular coordinate systems.
- Use gml:Geodesic curve segment

Discussion

Create New Geometry Types

Great Circle

A Great Circle is a shortest curve on a sphere connecting two specified points. It defines a section of the sphere that contains a diameter. A possible encoding for a Great Circle in GML would be as follows:

```
<abc:GreatCircle>   <!-- derives from gml:Curve -->
   <abc:start>
      <gml:Point> .. </gml:Point>  <!-- Can be any coordinate system - in particular WGS84 -->
   </abc:start>
   <abc:end>
      <gml:Point> .. </gml:Point>
   </abc:end>
</abc:GreatCircle>
```

or more compactly:

```
<abc:GreatCircle>
  <gml:posList> .. </gml:posList>
</abc:GreatCircle>
```

Issues

- Lack of support in both GML and WFS for this new Great Circle geometry.
- Only makes sense in specific coordinate systems.

Rhumb Lines

A rhumb line (also known as a loxodrome) is a line which cuts a meridian on a given surface at any constant angle, bu a right angle. If the surface is a sphere, the loxodrome is a spherical spiral. The loxodrome is the path taken when a compass is kept pointing in a constant direction. It is a straight line on a Mercator projection or a logarithmic spiral on polar projection.

Rhumb lines are NOT geodesics of the surface.

A possible encoding for a Rhumb Line would be as follows:

```
<RhumbLine xmlns="http://www.faa.gov/aixm"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.faa.gov/aixm FAASampleRhumbLine.xsd">
  <gml:posList>20 30 25 40</gml:posList>
  <horizontalAngle uom="deg">12</horizontalAngle>
  <!-- values in this example may not be correct -->
</RhumbLine>
```

A Rhumb Line could be encoded as a Curve Segment, hence the proposed RhumbLine above would derives from gml:AbstractCurveSegmentType and substitute for gml:_CurveSegment. The schema for the above RhumbLine would be:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  targetNamespace="http://www.faa.gov/aixm" xmlns:aixm="http://www.faa.gov/aixm"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:import namespace="http://www.opengis.net/gml"
    schemaLocation="gml.xsd"/>
  <xs:element name="RhumbLine" substitutionGroup="gml:_CurveSegment"
    type="aixm:RhumbLine">
    <xs:annotation>
      <xs:documentation>Rhumb line curve</xs:documentation>
    </xs:annotation>
  </xs:element>
  <xs:complexType name="RhumbLine">
    <xs:complexContent>
      <xs:extension base="gml:AbstractCurveSegmentType">
        <xs:sequence>
          <xs:element ref="gml:posList"/>
          <xs:element ref="aixm:horizontalAngle"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:element name="horizontalAngle" type="gml:MeasureType"/>
</xs:schema>
```

Issues

- Lack of support in both GML and WFS for this new RhumbLine geometry.

Use Linestrings in projective coordinates

Using gml:LineString in projective coordinates we can also construct the special curves so that in the projective coordinates they are always straight lines. For example:

- Gnomonic Projection: straight lines are great circles on sphere (ellipsoid?)
- Mercator Projection: straight lines are rhumb lines on sphere (ellipsoid?)

So if we write:

```
<gml:LineString srsName="Mercator">
   <gml:posList>.51 .45 .45 .61</posList>
</gml:LineString>
```

it is a **Rhumb Line** when lifted onto the sphere.

Whereas, if we write:

```
<gml:LineString srsName="Gnomonic">
   <gml:posList>.51 .45 .45 .61</posList>
</gml:LineString>
```

it is a **Great Circle**.

The problem with this approach is that these definitions make sense ONLY in the coordinate system in which they are defined. If one transforms a LineString (via a WFS) defined in Mercator to Gnomonic, a new LineString is returned - however this LineString represents a different curve on the surface of the earth. This can be easily seen by taking a LineString with end points defined in WGS84 (e.g. (45,100), (45,120)) and transforming it to a Gnomonic Projection. The original LineString lifts to a small circle (i.e. a line of latitude parallel to the equator), while the transformed LineString lifts to a great circle.

Use GML Geodesic Curve Segment

Geodesics are a supported geomerty in GML 3.1.1. A Geodesic is defined by a posList set of control points, or by a pai of pos/pointProperty elements. Interpolation between the control points is specified by a gml:CurveInterpolationType attribute on Geodesic (an enumeration). This attribute is fixed at "geodesic". A GeodesicString is defined as a sequenc of Geodesic segments.

Note that in GML a Geodesic is a "curve segment" and not a curve. To use it, one constructs a curve as in:

```
<gml:Curve srsName = "..">
   <gml:segments>
       <gml:Geodesic curveInterpolationType ="geodesic">
           <gml:posList> .. </gml:posList>
       </gml:Geodesic>
   </gml:segments>
</gml:Curve>
```

The srsName is used by the rendering or analysis application to compute (interpolate) points between the specified geodesic curve segment end points. The srsName may refer to a well known CRS, a well known CRS given by a URN, c a dictionary entry such as:

```
<?xml version="1.0" encoding="UTF-8" ?>
<gml:Dictionary gml:id="sample" xmlns="http://www.opengis.net/gml"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/gml
```

```
  gml.xsd">
  <gml:name>Sample CRS Dictionary for FAA</gml:name>
  <gml:dictionaryEntry>
    <GeographicCRS gml:id="Sphere">
      <srsName/>
      <usesEllipsoidalCS/>
      <usesGeodeticDatum>
        <GeodeticDatum gml:id="BasedonAuthalicSphere">
          <datumName>ABC</datumName>
          <usesPrimeMeridian>
            <PrimeMeridian gml:id="Greenwhich">
              <meridianName>Greenwhich</meridianName>
              <greenwichLongitude>
                <angle uom="degrees">0</angle>
              </greenwichLongitude>
            </PrimeMeridian>
          </usesPrimeMeridian>
          <usesEllipsoid>
            <Ellipsoid gml:id="epsg_7048">
              <ellipsoidName>1980 Authalic Sphere</ellipsoidName>
              <semiMajorAxis uom="#m">6371000</semiMajorAxis>
              <secondDefiningParameter>
                <inverseFlattening uom="non-dimensional">0</inverseFlattening>
              </secondDefiningParameter>
            </Ellipsoid>
          </usesEllipsoid>
        </GeodeticDatum>
      </usesGeodeticDatum>
    </GeographicCRS>
  </gml:dictionaryEntry>
  <gml:dictionaryEntry>
    <GeographicCRS gml:id="WGS84" xmlns="http://www.opengis.net/gml"
      xmlns:gml="http://www.opengis.net/gml"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.opengis.net/gml
      coordinateReferenceSystems.xsd">
      <srsName/>
      <usesEllipsoidalCS/>
      <usesGeodeticDatum>
        <GeodeticDatum gml:id="epsg_6326">
          <datumName>World Geodetic System 1984</datumName>
          <usesPrimeMeridian>
            <PrimeMeridian gml:id="epsg_8901">
              <meridianName>Greenwhich</meridianName>
              <greenwichLongitude>
                <angle uom="degrees">0</angle>
              </greenwichLongitude>
            </PrimeMeridian>
          </usesPrimeMeridian>
          <usesEllipsoid>
            <Ellipsoid gml:id="epsg_7050">
              <ellipsoidName>WGS 84</ellipsoidName>
              <semiMajorAxis uom="#m">6378137</semiMajorAxis>
              <secondDefiningParameter>
                <inverseFlattening uom="non-dimensional"
                >298.257223563</inverseFlattening>
              </secondDefiningParameter>
            </Ellipsoid>
          </usesEllipsoid>
        </GeodeticDatum>
      </usesGeodeticDatum>
    </GeographicCRS>
  </gml:dictionaryEntry>
</gml:Dictionary>
```

This CRS defines a spherical earth, hence the geodesic is a **Great Circle**.

It should be noted that for curves encoded as gml:Geodesic, analysis and rendering applications will need to be able to compute points on the geodesic. They will know which equation to use based ofn the coordinate system referenced in the Curve geometry that contains the Geodesic segments. Note that this is NOT different than the case for LineStrings it is just that the fomula for the computation is more complex.

Suppose we are given two points in (lat,long); this means that we have coordinates (u,v) in the plane for these two points and that we can compute their positions on the earth model via the standard parameterization. For an ellipsoid this is:

x(u,v) = acos(theta)sin(phi)
y(u,v) = asin(theta)sin(phi)
z(u,v) = ccos(phi)

where a,c are the ellipsoid parameters. These parameters are given by the GRS80 ellipsoid in the case of WGS84.

Note that two cases must be distinguished when thinking about "coordinate transformation":

- coordinate conversion - underlying surface model is **not** changed
- surface transformation - underlying surface model is changed.

Coordinate Conversion

In this case we simply get new coordinates for the end points of the Geodesic. The consumer of the data can see that the earth model is unchanged, and uses the same equation to compute points along the curve as prior to the conversion.

Surface Transformation

This case applies when transforming from one surface model (earth model) to another. This may also involve a change in the type of parameterization of the new surface (e.g. the surface model may be changed say from ellipsoidal to spherical and the parameterization from geographic (spherical coordinates) to mercator. The data consumer can see that the underlying surface model has changed (from the srsName tracing to the different earth model), has the new coordinates of the geodesic end points, can use the parameterization implied by the new coordinate system to compute the equivalent end points on the new surface, and must obtain an equation for geodesics on the target surface in order to compute intermediate points.

NB: Going from Spherical Earth to Ellipsoidal Earth is NOT a change of coordinates but a manifold transformation.

Examples

The following examples are provided:

1. Sample CRS dictionary with Sphere based Geographic CRS and WGS84 based CRS
2. Sample features with geodesic geometry curve segments using above two CRS
3. Mathematica code that computes points on the Geodesic using CRS parameters

Computation and Visualization

The following paragraph illustrates the basic steps in computing or visualizing these special curves. It is assumed that the curves in question are obtained from a WFS.

1. Obtain the curve in question from the WFS using a geospatial query (i.e. the curve will be obtained as the value

of a geometry property of some feature - e.g. as part of a route segment).

2. For a gml:Geodesic segment, obtain the earth model via the CRS reference (the CRS reference may be well known - so this is likely done at code development time), and use this to obtain the geodesic equation. Comput points on the geodesic as required for analysis or visualization. This applies to both Great Circles on a spherical earth nodel and to geodesics on a WGS84 ellipsoid earth model.
3. For a gml:RhumbLine segment, obtain the earth model via the CRS reference (the CRS reference may be well known - so this is likely done at code development time), and use this to obtain the RhumbLine equation. Compute points on the RhumbLine as required for analysis or visualization.

Recommendations

- Represent geodesics by gml:Geodesic. The underlying point space (manifold) on which the geodesic is defined can be determined from the <usesEllipsoid/> property of the associated CRS referenced from srsName.
- Represent Great Circles by gml:Geodesic. The CRS referenced must be a Geographic CRS with a spherical ellipsoid model.
- Represent Rhumb Lines by introducing a new geometry and additionally encoding the bearing.

Attachments:

- FAASampleCRSDictionary.xml (text/xml)
- FAASampleFeatureGeodesic.xsd (text/xml)
- SampleGeodesicOnWGS84Ellipsoid.xml (text/xml)
- SampleGreateCircleOnSphere.xml (text/xml)

Document generated on 06 Oct 2005 22:30