Geopriv J. Winterbottom
Internet-Draft M. Thomson
Expires: August 18, 2005 Nortel
H. Tschofenig
Siemens
February 14, 2005

GEOPRIV PIDF-LO Usage Clarification, Considerations and Recommendations
draft-winterbottom-geopriv-pdif-lo-profile-00.txt

Status of this Memo

Copyright Notice

Abstract

The GeoPriv PIDF-LO specification provides a flexible and versatile means to represent location information. There are, however, circumstances that arise when information needs to be constrained in

how it is represented so that the number of options that need to be implemented in order to make use of it are reduced. There is growing interest in being able to use location information contained in a PIDF-LO for message and call routing applications. For such applications to interoperate successfully location information will need to be normative and more constrained than is currently described in the PIDF-LO specification. This paper makes recommendations on how to constrain, represent and interpret locations in a PIDF-LO. It further looks at existing communications standards that make use of geodetic information for routing purposes and recommends a subset of GML that MUST be implemented by applications to allow message routing to occur.

Table of Contents

1. Introduction

PIDF-LO [1] which was developed by the GEOPRIV working group, is the IETF recommended way encoding location information and privacy policies. Location information in PIDF-LO may be described in a geospatial manner based on a subset of GMLv3, or as civic location information. PIDF-LO may be used in a variety of ways. For example, [3] motivates the usage of PIDF-LO in presence based systems. Further usages are envisioned in the context of emergency services and other location based routing applications. This document details the usage of GMLv3 in PIDF-LO by incorporating implementation experience. Recommendations for formats and conventions are provided where interoperability might be problematic. For the sake of compatibility, ease of use and removal of inherent ambiguity apparent in GML the functionality of other geospatial systems, such as the 3GPP MLP standard [4], are examined.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [2].

3. Using Location Information

The PIDF format provides for an unbounded number of tuples. The geopriv element resides inside the status component of a tuple, hence a single PIDF document may contain an arbitrary number of location objects some or all of which may be contradictory or complementary. The actual location information is contained inside a <location-info> element, and there may be one or more actual locations described inside the <location-info> element.

Graphically, the structure of the PIDF/PIDF-LO can be depicted as follows:

```
PIDF document
   tuple 1
       status
            geopriv
                location-info
                    civicAddress
                     location
                 usage-rules
   tuple 2
   tuple 3
```

All of these potential sources and storage places for location lead to confusion for the generators, conveyors and users of location information. Practical experience within the United States National Emergency Number Association (NENA) in trying to solve these ambiguities led the following conventions being adopted:

Rule #1: A GeoPriv tuple MUST completely define a specific location.

Rule #2: Where a location can be uniquely described in more than one way, each location description SHOULD reside in a separate tuple.

Rule #3: Providing more than one location in a single presence document (PIDF) MUST only be done if all objects describe the same location.

Rule #4: Providing more than one location in a single <location-info> element SHOULD be avoided where possible.

Rule #5: When providing more than one location in a single <location-info> element they MUST be provided by a common source. If you have more than one location in the <location-info> element, then the combination (complex of) these elements defines the complete location.

Rule #6: Providing more than one location in a single <location-info> element SHOULD only be done if they form a complex to describe the same location. For example, a geodetic location describing a point, and a civic location indicating the floor in a building.

Rule #7: Where a location complex is provided in a single <location-info> element, the higher precision locations MUST be provided first. For example, a geodetic location describing a point, and a civic location indicating the floor MUST be represented with the point first followed by the civic location.

Rule #8: Where a PIDF document contains more than one tuple containing a status element with a geopriv location element , the priority of tuples SHOULD be based on tuple position within the PIDF document. That is to say, the tuple with the highest priority location occurs earliest in the PIDF document. Initial priority SHOULD be determined by the originating UA, the final priority MAY be determined by a proxy along the way.

Rule #9: Where multiple PIDF documents are contained within a single request, document selection SHOULD be based on document order.

The following examples illustrate the useful of these rules.

3.1 Single Civic Location Information

Jane is at a coffee shop on the ground floor of a large shopping mall. Jane turns on her laptop and connects to the coffee-shop's WiFi hotspot, Jane obtains a complete civic address for her current location, for example using [5]. She constructs a Location Object which consists of a single PIDF document, with a single geopriv tuple, with a single location residing in the <location-info> element. This is largely unambiguous, and if this location is sent over the network, providing it understands civic addresses, correct handling of any request should be possible.

3.2 Civic and Geospatial Location Information

Mike is visiting his Seattle office and connects his laptop into the Ethernet port in a spare cube. Mike's computer receives a location over DHCP as defined in [6]. In this case the location is a geodetic location, with the altitude represented as a building floor number. This is constructed by Mike's computer into the following PIDF document:

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
   xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
   xmlns:gml="urn:opengis:specification:gml:schema-xsd:feature:v3.0"
   entity="pres:mike@seattle.example.com">
   <tuple id="sg89ab">
     <status>
      <gp:geopriv>
        <gp:location-info>
          <cl:civilAddress>
             <cl:country>US</cl:country>
             <cl:FLR>2</cl:FLR>
          </cl:civilAddress>
         </gp:location-info>
         <gp:usage-rules>
         </gp:usage-rules>
       </gp:geopriv>
      </status>
     <timestamp>2003-06-22T20:57:29Z</timestamp>
   </tuple>
   <tuple id="sg89ae">
     <status>
      <gp:geopriv>
        <gp:location-info>
          <gml:location>
            <gml:Point gml:id="point1" srsName="epsg:4326">
              <gml:coordinates>37:46:30N 122:25:10W</gml:coordinates>
            </gml:Point>
           </gml:location>
        </gp:location-info>
        <gp:usage-rules>
        </gp:usage-rules>
      </gp:geopriv>
     </status>
    <timestamp>2003-06-22T20:57:29Z</timestamp>
   </tuple>
</presence>
```

So the resulting PIDF document contains two geopriv elements each in a separate PIDF tuple element, the first being a civic address made up of only country and floor, the second containing the received geodetic information. If the location is required for emergency routing purposes, which information does a SIP proxy use? Applying rule #8, we will likely fail, or at a minimum need to fall back to the second tuple describing the geodetic location, a routing by the second floor somewhere in the US is not particularly descriptive. If rule #6 and #7 are applied, then the PIDF-LO document would look like:

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
   xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
   xmlns:gml="urn:opengis:specification:gml:schema-xsd:feature:v3.0"
   entity="pres:mike@seattle.example.com">
   <tuple id="sg89ab">
     <status>
      <gp:geopriv>

        <gp:location-info>
          <gml:location>
            <gml:Point gml:id="point1" srsName="epsg:4326">
              <gml:coordinates>37:46:30N 122:25:10W</gml:coordinates>
            </gml:Point>
          </gml:location>
          <cl:civilAddress>
             <cl:country>US</cl:country>
             <cl:FLR>2</cl:FLR>
          </cl:civilAddress>
         </gp:location-info>
         <gp:usage-rules>
         </gp:usage-rules>
       </gp:geopriv>
      </status>
     <timestamp>2003-06-22T20:57:29Z</timestamp>
   </tuple>
</presence>
```

It is now clear that the main location of user is a geodetic location at latitude 37:46:30 North and longitude 122:25:10 West. Further that the user is on the second floor of the building located at those coordinates.

3.3 Manual/Automatic Configuration of Location Information

Erin has a predefined civic location stored in her laptop, since she normally lives in Sydney, the address in her address is for her Sydney-based apartment. Erin decides to visit sunny San Francisco, and when she gets there she plugs in her laptop and makes a call. The location sent to the local proxy is her Sydney address, her local outbound SIP proxy inserts a new PIDF document asserting her new location (or returns an error message with the current location information). Using rule #9, the resulting PIDF order should be San Francisco document first, followed by Sydney document. If the San Francisco proxy were to add the location to Erin's existing PIDF document, then the San Francisco tuple SHOULD be placed ahead of the Sydney tuple following rule #8.

4. Geodetic Coordinate Representation

The geodetic examples provided previously were all based around the gml:location element which uses the gml:coordinates elements (inside the gml:Point element) and this representation has several drawbacks. Firstly, it has been deprecated in later versions of GML (3.1 and beyond) making it inadvisable to use for new applications. Secondly, the format of the coordinates type is opaque and so can be difficult to parse and interpret to ensure consistent results, as the same geodetic location can be expressed in a variety of ways. An alternative is to use the gml:position and gml:pos elements. These elements have a structured format, in that each field is represented as a double, and a single space exists between each field. Such a format does not introduce the same degree of misinterpretation. A suggested representation therefore for expressing geodetic coordinates for emergency call routing would be:

```
<gml:position>
  <gml:Point gml:id="point1" srsName="urn:EPSG:geographicCRS:4326">
    <gml:pos>37.775 -122.422</gml:pos>
  </gml:Point>
</gml:position>
```

The coordinate reference system (CRS) indicates which numbers in the sequence equate to latitude, longitude etc, and in addition to this the CRS also provides an indication of direction represented by the sign of the number. For example, in WGS-84 (represented as CRS:4326), as shown in the code snippet above, the format is latitude followed by longitude. A positive value for latitude represents a location north of the equator while a negative value represents a location south of the equator. Similarly for longitude, a positive value represents a location east of Greenwich, while a negative value represents a location west of Greenwich.

EPSG 4326 is the two dimensional WGS-84 representation, if we wanted to represented this in three dimensions, that is with an altitude as well, then we would use EPSG 4979 and the format would be as follows:

```
<gml:position>
  <gml:Point gml:id="point1" srsName="urn:EPSG:geographicCRS:4979">
    <gml:pos>37.775 -122.422 22</gml:pos>
  </gml:Point>
</gml:position>
```

The format using CRS:4979 is similar to CRS:4326, though we now have an altitude value. Specifically the altitude is provided in metres

above the geoid, which will not be useful for general routing applications since the geoid is generally neither ground-level nor sea-level. However, for more specialized geographic applications it may be useful.

Revisiting the final version of Section 3.2, but using gml:position and gml:pos, has the following structure:

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
   xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
   xmlns:gml="urn:opengis:specification:gml:schema-xsd:feature:v3.0"
   entity="pres:mike@seattle.example.com">
   <tuple id="sg89ab">
     <status>
      <gp:geopriv>
        <gp:location-info>
          <gml:position>
            <gml:Point gml:id="point1" srsName="epsg:4326">
              <gml:pos>37.775 -122.422</gml:pos>
            </gml:Point>
          </gml:position>
          <cl:civilAddress>
             <cl:country>US</cl:country>
             <cl:FLR>2</cl:FLR>
          </cl:civilAddress>
         </gp:location-info>
         <gp:usage-rules>
         </gp:usage-rules>
       </gp:geopriv>
      </status>
     <timestamp>2003-06-22T20:57:29Z</timestamp>
   </tuple>
</presence>
```

5. Uncertainty in Location Representation

The cellular mobile world today makes extensive use of geodetic based location information for emergency and other location-based applications. Generally these locations are expressed as a point (either in two or three dimensions) and an area or volume of uncertainty around the point. In theory, the area or volume represents a coverage in which the user has a relatively high probability of being found, and the point is a convenient means of defining the centroid for the area or volume. In practice, most systems today use the point as an absolute value and ignore the uncertainty. It is difficult to determine if systems have been implement in this manner for simplicity, and even more difficult to predict if uncertainty will play a more important role in the future. An important decision is whether an uncertainty area should be specified.

There are six common ways to represent location and uncertainty, but are listed below for completeness:

- o Arc band
- o Ellipsoid point with uncertainty circle
- o Polygon
- o Ellipsoid point with altitude
- o Ellipsoid point with uncertainty ellipse
- o Ellipsoid point with altitude and uncertainty ellipsoid

GML was designed to provide a very flexible abstraction on which specific representations of geometric and geographic schemes could be extended. Representing some of the above shapes is difficult if not impossible using base GML. However, only a subset of GML, namely feature.xsd, is mandatory for a PIDF-LO implementation. Extending GML to easily represent these shapes may lead to interoperability issues and so is not recommended. The authors of this document were unable to find a means to express either an ellipse or and ellipsoid using only the elements defined in feature.xsd.

The following sections describe four shapes that can be defined in GML, and show the equivalent representation in 3GPP MLP [4].

5.1 Arc band

Arc band is used primarily where timing advance (TA) information is known. Timing advance is a mechanism used in wireless communications to help ensure that handsets and base-stations remained synchronized. Timing advance is stepped based on signal propagation and is fairly deterministic, for GSM each increase in TA value represents 553.85 metres.

The arc band type was developed to represent the area between two successive TA values and an antenna opening. This is presented in 3GPP as a point, two radii, and two angles representing the start and the stop of the angles for the opening.

```
                 ,'.__
                /     `-.
               /         `-.
              /             `.
             /                \
            /__                \
           .   `-.              \
     start.       `.             \
     angle          \             .
         .          |             |
        o           '             |
          .        /              '
           .      /              ;
        stop . .,'              /
        angle   `.             /
                  `.          /
                    `.      ,'
                      `.  ,'
                        `'
```

```
<pd>
  <time utc_off="+1000">20041201092843</time>
  <shape>
    <CircularArcArea>
      <coord>
        <X>42.5463</X>
        <Y>-73.2512</Y>
      </coord>
      <inRadius>1938.5</inRadius>
      <outRadius>2492.3</outRadius>
      <startAngle>63.7</startAngle>
      <stopAngle>118.4</stopAngle>
    </CircularArcArea>
  </shape>
</pd>
```

The GML representation of this is below:

```
<?xml version="1.0"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
```

```
             xmlns:pidf="urn:ietf:params:xml:ns:pidf"
             xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
             xmlns:gml="http://opengis.net/gml"
             entity="pres:user@example.com">
      &lt;tuple id="a6fea09"&gt;
        &lt;status&gt;
          &lt;gp:geopriv&gt;
            &lt;gp:location-info&gt;
              &lt;gml:extentOf&gt;
                &lt;gml:Polygon&gt;
                  &lt;gml:exterior&gt;
                    &lt;gml:Ring&gt;
                      &lt;gml:curveMember&gt;
                        &lt;gml:Curve&gt;
                          &lt;gml:segments&gt;
                            &lt;gml:ArcByCenterPoint &gt;
                              <gml:pos
                                  srsName="urn:EPSG:geographicCRS:4326">
                                  42.5463 -73.2512
                              &lt;/gml:pos&gt;
                              &lt;gml:radius uom="urn:EPSG:uom:9001"&gt;
                               2492.3
                              &lt;/gml:radius&gt;
                       <!--
                         It is difficult to determine the correct
                         interpretation of GML and EPSG #4326 to
                         determine how these angles are to be
                         interpreted.
                         Neither specification specifies how the values
                         are to be interpolated.  That is, the direction
                         of rotation from start angle to end angle.  It
                         is therefore assumed that a "clockwise"
                         (Northing to Easting) direction is chosen to
                         link the two points on the arc.
                         It is also assumed that a value of 0 degrees
                         indicates Northing and 90 degrees indicates
                         Easting.
                        -->
                              &lt;gml:startAngle uom="urn:EPSG:uom:9102"&gt;
                                63.7
                              &lt;/gml:startAngle&gt;
                              &lt;gml:endAngle uom="urn:EPSG:uom:9102"&gt;
                                118.4
                              &lt;/gml:endAngle&gt;
                            &lt;/gml:ArcByCenterPoint&gt;
                            &lt;gml:LineStringSegment&gt;
                              <gml:posList
                                srsName="urn:EPSG:geographicCRS:4326">
```

```
                            42.535651 -73.224473 42.538018 -73.230411
                          </gml:posList>
                        </gml:LineStringSegment>
                        <gml:ArcByCenterPoint >
                          <gml:pos
                             srsName="urn:EPSG:geographicCRS:4326">
                             42.5463 -73.2512
                          </gml:pos>
                   <!--
                     Note that the decision to go with a "clockwise"
                     pass means that the start position of this
                     second arc is not contiguous with the end of
                     the last line.
                   -->
                          <gml:radius uom="urn:EPSG:uom:9001">
                            1938.5
                          </gml:radius>
                          <gml:startAngle uom="urn:EPSG:uom:9102">
                            63.7
                          </gml:startAngle>
                          <gml:endAngle uom="urn:EPSG:uom:9102">
                            118.4
                          </gml:endAngle>
                        </gml:ArcByCenterPoint>
                        <gml:LineStringSegment>
                          <gml:posList
                           srsName="urn:EPSG:geographicCRS:4326">
                           42.554016 -73.230007 42.556220 -73.223952
                          </gml:posList>
                        </gml:LineStringSegment>
                      </gml:segments>
                    </gml:Curve>
                  </gml:curveMember>
                </gml:Ring>
              </gml:exterior>
            </gml:Polygon>
          </gml:extentOf>
        </gp:location-info>
        <gp:usage-rules>
        </gp:usage-rules>
      </gp:geopriv>
    </status>
    <timestamp>2004-12-01T09:28:43+10:00</timestamp>
  </tuple>
</presence>
```

This representation poses a few potential problems over the 3GPP representation . In the 3GPP representation the point is absolute,

and everything else is defined relative to this point, ensuring that the band is indeed bounded. The representation of arc band above does not share all of these properties. In the GML arc band representation above, the point and radii are relative, but the bounding lines of the starting and finishing angles are not, these are necessarily defined as independent line segments. By having to define the arc enclosures as individual line segments it is possible to define an unbounded arc band which would consist of two arcs some arbitrary distance apart with two lines that may or may not intersect them.

A second concern with this representing uncertainty using this method, is that there is no explicit statement or way of indicating to the receiving application what type of uncertainty is being represented. Today several different representations of uncertainty are valid with in the same application, so knowing which type is being used, and how to interpret it is important, and this is particularly true if the shape must also be validated as is the case above.

Ensuring the legality of this shape type when represented in GML is more complex than in MLP as the type must first be determined before its validity can be assessed. Users of this shape type may be better served by a formal shape definition being introduced into GeoPriv so that these problems can be more readily overcome.

5.2 Ellipsoid Point With Uncertainty Circle

This shape type is used extensively over the North American NENA defined E2 interface for transporting mobile geodetic location from the MPC/GMLC to the ALI and subsequently the PSAPs. In 3GPP this is defined as a WGS-84 point (ellipsoid point), and a radius or uncertainty around that point, specified in metres. The 3GPP MLP representation for an ellipsoid point with uncertainty is defined as follows:

```
<pd>
  <time utc_off="+1000">20041201092843</time>
  <shape>
    <CircularArea>
      <coord>
        <X>42.5463</X>
        <Y>-73.2512</Y>
      </coord>
      <radius>850.24</radius>
    </CircularArea>
  </shape>
```

```
</pd>
```

This shape is similarly defined in GML below:

```
<?xml version="1.0"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
          xmlns:pidf="urn:ietf:params:xml:ns:pidf"
          xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
          xmlns:gml="http://opengis.net/gml"
          entity="pres:user@example.com">
  <tuple id="a6fea09">
    <status>
      <gp:geopriv>
        <gp:location-info>
          <gml:extentOf>
            <gml:Polygon>
              <gml:exterior>
                <gml:Ring>
                  <gml:curveMember>
                    <gml:Curve>
                      <gml:segments>
                        <gml:CircleByCenterPoint >
                          <gml:pos
                            srsName="urn:EPSG:geographicCRS:4326">
                            42.5463 -73.2512
                          </gml:pos>
                          <gml:radius uom="urn:EPSG:uom:9001">
                            850.24
                          </gml:radius>
                        </gml:CircleByCenterPoint>
                      </gml:segments>
                    </gml:Curve>
                  </gml:curveMember>
                </gml:Ring>
              </gml:exterior>
            </gml:Polygon>
          </gml:extentOf>
        </gp:location-info>
        <gp:usage-rules>
        </gp:usage-rules>
      </gp:geopriv>
    </status>
    <timestamp>2004-12-01T09:28:43+10:00</timestamp>
  </tuple>
</presence>
```

This type does not have all of the problems associated with the arc

band representation, in that the radius of the circle is relative to the centre, and so the validation is unnecessary. However it does suffer from the potential problem that the application still needs to determine the type of uncertainty being represented, though this maybe made more clear through the explicit use of the gml:CircleByCenterPoint element.

5.3 Polygon

A polygon is defined as a set of points to form an enclosed bounded shape. It is here that GML and the 3GPP shapes are most similar. The representation for a polygon in GML is given first:


```
<?xml version="1.0"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
          xmlns:pidf="urn:ietf:params:xml:ns:pidf"
          xmlns:gp="urn:ietf:params:xml:ns:pidf:geopriv10"
          xmlns:gml="http://opengis.net/gml"
          entity="pres:user@example.com">
  <tuple id="a6fea09">
    <status>
      <gp:geopriv>
        <gp:location-info>
          <gml:extentOf>
            <gml:Polygon>
              <gml:exterior>
                <gml:LinearRing>
                  <gml:posList
                    srsName="urn:EPSG:geographicCRS:4326">
                    42.556844 -73.248157
                    42.549631 -73.237283
                    42.539087 -73.240328
                    42.535756 -73.254242
                    42.542969 -73.265115
                    42.553513 -73.262075
                    42.556844 -73.248157
                  </gml:posList>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </gml:extentOf>
        </gp:location-info>
        <gp:usage-rules>
        </gp:usage-rules>
      </gp:geopriv>
    </status>
    <timestamp>2004-12-13T14:49:53+10:00</timestamp>
```

```
    </tuple>
  </presence>
```

The GML object here is clear in its definition. A gml:LinearRing MUST have a minimum of four points, with the first and last points being the same. The 3GPP MLP representation for a polygon is provided below.


```
<pd>
  <time utc_off="+1000">20041201092843</time>
  <shape>
    <Polygon>
      <outerBoundaryIs>
        <LinearRing>
          <coord>
            <X>42.556844</X>
            <Y>-73.248157</Y>
          </coord>
          <coord>
            <X>42.549631</X>
            <Y>-73.237283</Y>
          </coord>
          <coord>
            <X>42.539087</X>
            <Y>-73.240328</Y>
          </coord>
          <coord>
            <X>42.535756</X>
            <Y>-73.254242</Y>
          </coord>
          <coord>
            <X>42.542969</X>
            <Y>-73.265115</Y>
          </coord>
          <coord>
            <X>42.553513</X>
            <Y>-73.262075</Y>
          </coord>
          <coord>
            <X>42.556844</X>
            <Y>-73.248157</Y>
          </coord>
        </LinearRing>
      </outerBoundaryIs>
    </Polygon>
  </shape>
</pd>
```

While these two representations are very similar and precise, they are not widely used at present. If only a coverage area is required without a nominal central point requiring specification, then this form is ideal for representation using GML.

6. Baseline Geometry

PIDF-LO suggests to use GMLv3 feature.xsd, which provides a subset of the available GML functionality. As a consequence a number of further XML files are implicitly included, namely geometryBasic0d1d.xsd, geometryBasic2d.xsd, temporal.xsd, measure.xsd, units.xsd, gmlBase.xsd, dictionary.xsd, xLinks.xsd and basicTypes.xsd, as being necessary to support. This provides for a vast range of possibilities which would pose significant complications to implementors wish to develop location dependent routing applications. By agreeing to a minimal set of data appropriate for routing, a minimum set of GML that MUST be implemented by a given application type can also be set. This does not preclude the additional functionality from being implemented, merely that it may not be understood by some nodes.

6.1 Zero Dimensions

The minimum supported set of elements is position/Point/pos provided by geometryBasic0d1d.xsd.

Thus a point location has only one representation as follows:

```
<gml:position xmlns:gml="http://www.opengis.net/gml">
     <gml:Point srsName="urn:ogc:def:crs:EPSG:4326">
             <gml:pos>4.5 -36.2</gml:pos>
     </gml:Point>
</gml:position>
```

The <location> and <coord> objects MUST NOT be used since they are deprecated in GML 3.1 and their functionality can be substituted with the above-described elements.

Note that pos allows altitude to be expressed based on the selected Coordinate Reference Systems (e.g., EPSG:4979 or EPSG:4326). Most Coordinate Reference Systems use altitude above the geoid and not altitude above the ground.

6.2 One Dimensions

Support for one dimensional shapes (such as the LineString or the posList object) is not required except as a part of two dimensional shapes.

geometryBasic0d1d.xsd provides these geometric properties and objects.

6.3 Two Dimensions

The examples previously used were all contructed using elements from this schema which reuse functionality from geometryBasic2d.xsd. As was described earlier the arcband definition in GML is problematic for producing a closed solid and SHOULD consequently be avoided. As a result of this, elements required exclusively for representing the arcband shape have not been included in the minimum supported element set. The minimum element set is therefore restricted to circle and polygon.

Circle:

```
extentOf/
   Polygon/
      exterior/
         Ring/
            curveMember/
               Curve/
                  segments/
                     CircleByCentrePoint/   -> Circle
                        pos
                        radius
```

Alternatively it would be possible to use the following structure to express a circle using the <gml:Circle> element with three pos elements as well. However, the usage of pos and radius, as shown above, is inline with the model used by the 3GPP.

Polygon:

```
extentOf/
   Polygon/
      exterior/
         LinearRing/
            pos or posList   -> Polygon
```

6.4 Three Dimensions

Support for three dimensions is not required

6.5 Envelopes

The Envelope element is a representation of a bounding box and can be expressed in two or three dimensions. Defining a space using the Envelope element should be done with extreme caution due to continuity problems at the extremities of the CRS. In WGS-84, two

envelopes are required at the 180th meridian. The minimum set of elements required to support an Envelope are:

```
boundBy/
   Envelope/
      upperCorner/
         Point/
            Pos
      lowerCorner/
          Point/
             Pos/
```

6.6 Temporal Dimensions

Support for temporal elements is not required

6.7 Units of Measure

The base SI units as a minimum MUST be supported. For measures of distance this is metres. The EPSG URN for metres is:

```
metres = urn:ogc:def:uom:EPSG:9001:6.6
```

Angles are frequently expressed in terms of both degrees and radians, consequently both MUST be implemented.

```
degrees = urn:ogc:def:uom:EPSG:9102:6.6
radians = urn:ogc:def:uom:EPSG:9101:6.6
```

Further units of measurement are not required.

6.8 Coordinate Reference System (CRS)

There are a very large number of coordinate reference systems in existence today, but many are, however, not in widespread use. Existing communications protocols such as those used in both the ANSI, 3GPP and NENA standards (see [7], [8], [9]) have standardized on WGS-84. It is recommended for routing purpose that only WGS-84 coordinate types MUST be implemented and further that this set be resatricted to the following:

```
WGS84(2D) = urn:ogc:def:crs:EPSG:4326:6.6
WGS84(3D) = urn:ogc:def:crs:EPSG:4979:6.6
```

7. Recommendations

As a summary this document gives a few recommendations on the usage of location information in PIDF-LO. Nine rules specified in Section 3 give guidelines on the ambiguity of PIDF-LO with regard to the occurrence of multiple location information. It is recommend that gml:position, gml:pos types be used to specify locations when locations are needed for routing and specifically emergency routing. Enhancements to GMLv3 feature.xsd may need to be defined to allow complex shapes types to be specified in a way that makes them easy to distinguish and validate. This is particularly important if the data is to be used during the decision making process of routing signaling messages.

Only a limited subset of GML functionality from the feature.xsd schema is necessary to describe a geodetic location with sufficient precision to allow a routing decision to be made. Restricting both the amount of GML that MUST be implemented, and the number of variations in which this data can be expressed significantly reduces the likelihood of interoperability issues in the future. Precedents exist in the other communications protocols for restricting CRS types and representations for the sake of simplicity and interoperability, and the recommendation is made to adopt similar restrictions for mandatory implementable components of GeoPriv.

8. Security Considerations

The primary security considerations relate to how location information is conveyed and used, which are outside the scope of this document. This document is intended to serve only as a set of guidelines as to which elements MUST or SHOULD be implemented by systems wishing to perform location dependent routing. The ramification of such recommendations is that they extend to devices and clients that wish to make use of such services.

9. IANA Considerations

This document does not introduce any IANA considerations.

10. Acknowledgments

The authors would like to thank the GEOPRIV working group for their discussions in the context of PIDF-LO. Furthermore, we would like to thanks Jon Peterson as the author of PIDF-LO and Nadine Abbott for her constructive comments in clarifying some aspects of the document.

11. References

11.1 Normative references

[1] Peterson, J., "A Presence-based GEOPRIV Location Object Format", Internet-Draft draft-ietf-geopriv-pidf-lo-03, September 2004.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", March 1997.

11.2 Informative References

[3] Peterson, J., "A Presence Architecture for the Distribution of GEOPRIV Location Objects", Internet-Draft draft-ietf-geopriv-pres-02, September 2004.

[4] "Mobile Location Protocol (MLP), OMA, Candidate Version 3.1", March 2004.

[5] Schulzrinne, H., "Dynamic Host Configuration Protocol (DHCPv4 and DHCPv6) Option for Civic Addresses Configuration Information", Internet-Draft draft-ietf-geopriv-dhcp-civil-04, October 2004.

[6] Polk, J., Schnizlein, J. and M. Linsner, "Dynamic Host Configuration Protocol Option for Coordinate-based Location Configuration Information", RFC 3825, July 2004.

[7] "TR-45 J-STD-036-AD-2 Enhanced Wireless 9-1-1 Phase 2".

[8] "3GPP TS 23.032 V6.0.0 3rd Generation Partnership Project; Technical Specification Group Code Network; Universal Geographic Area Description (GAD)".

[9] "NENA Standard for the Implementation of the Wireless Emergency Service Protocol E2 Interface".

Authors' Addresses

James Winterbottom
Nortel
Wollongong
NSW Australia

Email: winterb@nortel.com

Martin Thomson
Nortel
Wollongong
NSW Australia

Email: martin.thomson@nortel.com

Hannes Tschofenig
Siemens
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany

Email: Hannes.Tschofenig@siemens.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.