Style Definition: Heading 3;h3;sub-clause 3;H3;hd3: Space Before: 12 pt, After: 6 pt

Open Geospatial Consortium Inc.

Date: 14 June 2011

Reference number of this document: OGC 11-060

Editors: **OGC Aviation Domain Working Group**

Use of GML for aviation data

Warning

This document is not an OGC Standard. This is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

i. Preface

This paper provides guidelines for some aspects of the GML encoding that are specific to the aviation data domain.

ii. Document terms and definitions

This document uses the standard terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

iii. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Role	Organization
David Burggraf	Contributor	Galdos, Inc.
Warwick Dufour	Contributor	Avitech AG
Johannes Echterhoff	Contributor	iGSI
Benoit Geffroy	Contributor	EgisAvia
Razvan Guleac	Contributor	EUROCONTROL
Alain Kabamba	Contributor	Erdas
Michal Kadlec	Contributor	Avitech AG
Eduard Porosnicu	Editor	EUROCONTROL
Bert Robben	Contributor	Luciad
Scott Wilson	Contributor	EUROCONTROL
Anyone missing??	**???**	**???**

1 Introduction

1.1 Objective

The AIXM 5.1 schema uses Geographical Markup Language (GML) version 3.2.1 for the encoding of positional and shape data of aeronautical information items, such as airspace, runway thresholds, navaids, etc.

The ISO 19107 spatial schema, which is implemented by GML, is very complex. It contains an extensive list of geometries, geometric properties and operations – many of which are not necessary for aeronautical information applications. In addition, the ISO 19107 contains an exhaustive 3D geometry model that is probably not needed in its entirety for AIXM either. Therefore, a GML profile for AIXM needs to be defined. The objective of this document is to identify the elements of the AIXM-GML profile and also to provide guidelines for the use of GML constructs in AIXM data sets.

1.2 Scope – Applicability

The discussion paper focuses on guidelines for a aeronautical data encoding using the Aeronautical Information Exchange Model (AIXM).

1.3 References

[1] ICAO Annex 15 (13th Edition) – Aeronautical Information Services

[2] ISO 19107:2003 – Geographic information — Spatial schema

[3] ISO 19136:2007 – Geography Markup Language

[4] OGC 07-092r3 - Definition identifier URNs in OGC namespace

[5] OGC 06-042 - OpenGIS® Web Map Server Implementation Specification

[6] EAD AIXM4.5-to-5.1_Mapping.doc

[7] EAD AIXM 5.1 Conceptual Manual 0.2.doc

Deleted: 05-010 – URNs of definitions in OGC namespace

Deleted: OGC

Deleted: 04-024 (ISO DIS 19128) – Web Map Service

1.4 Interpretation

The following definitions shall apply:

– ‘data set’ means identifiable collection of data

2 Geographical data in Aeronautical Information

Aeronautical Information (AI) was traditionally presented and communicated by States using paper documents, such as AIP, charts, manuals. This data frequently includes geographically related information items, such as:

- Positions expressed in latitude/longitude, which according to ICAO Annex 15 shall be with reference to the WGS-84 datum;
- Shapes of airspace, expressed as series of positions in combination with arcs of circles or as full circles; sometimes, these contain references to national borders, water courses, etc., which are not provided explicitly in the AIP;
- More recently, shapes of obstacles, provided as point, line or polygon, again using series of positions and arcs of circle.

The Aeronautical Information Exchange Model (AIXM) is used since 2003 for the provision by States of AI in digital format. The AIXM versions up to 4.5 used a custom XML encoding, not based on OGC standards. Since 2008, with the publication of version 5, the AIXM specification has embraced GML as data encoding format.

There are a number of specificities in the AI Domain that concern the provision of geographical and geometrical information. These are discussed in this document, which also includes recommendations for data encoding in GML.

Deleted: modeled in AIXM

Deleted: and still is

Deleted: the

Deleted: the way that such

Deleted: is currently communicated in the AI domain

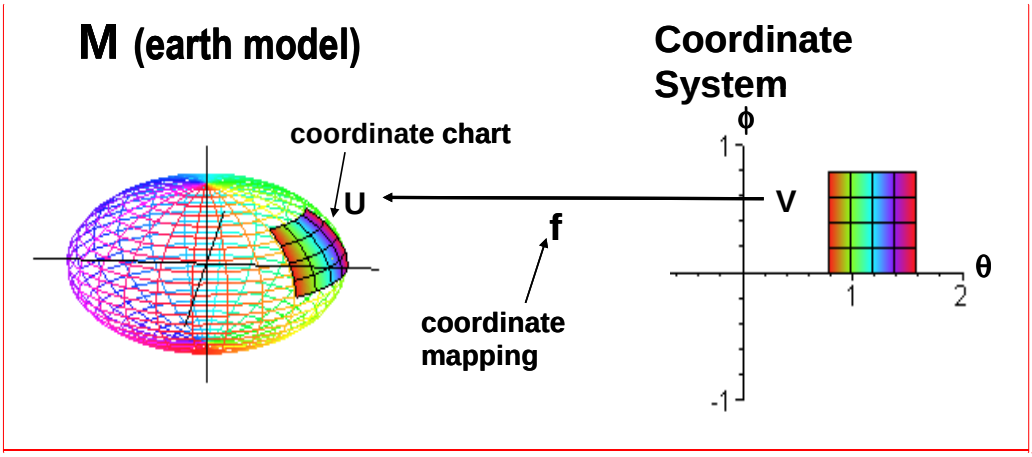
Deleted: the following sub-sections

3 WGS-84

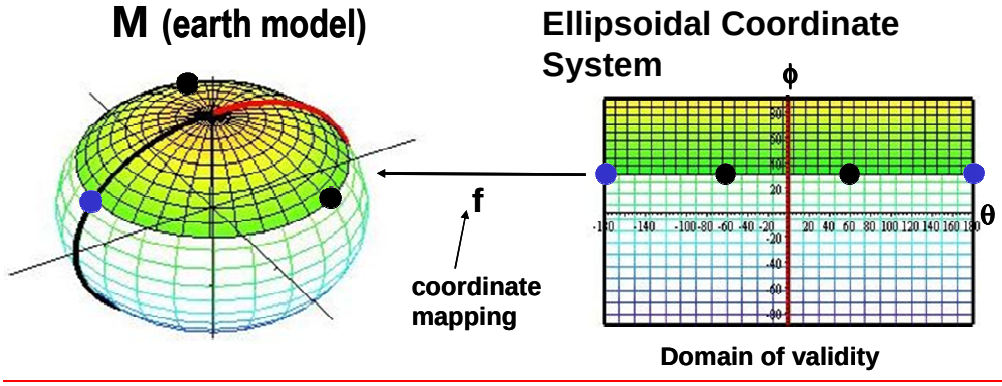
According to ICAO Annex 15, all "*published aeronautical geographical coordinates (indicating latitude and longitude) shall be expressed in terms of the WGS-84 geodetic reference datum*".

3.1 Use of srsName

In GML, the geodetic datum is specified by reference to a Coordinate Reference System (CRS). A CRS relates a coordinate system to the earth by a datum. A geodetic datum consists of an ellipsoid model and a prime meridian. The intersection of the equator and prime meridian is the origin of the CRS.



A geodetic CRS (e.g. EPSG 4326) relates a (lat,lon) ellipsoidal coordinate system to the Earth.



The ***CRS reference is critical for the correct encoding and processing*** of the geographical data contained in AIXM/GML files. It indicates not only the geodetic

Comment: Images copied from presentation of David Burggraf in Australia TC meeting.

Deleted: an

reference datum, but also the order of the coordinate axes (latitude/longitude or longitude/latitude) and has important implications for the convention used for measuring angles (from the North, clockwise, for example). This will be discussed in more detail, later in this document.

Deleted: further

There are two main CRS definition authorities that are relevant for the AI domain: Oil and Gas Producers (OGP), formerly known as the European Petroleum Survey Group (EPSG) and OGC themselves. The two sets of CRS definitions have many common points. For example, the OGC:CRS84 is a variant of EPSG:4326 (differing only in its coordinate order: longitude/latitude) and is defined in the ISO 19128 Geographic information — Web Map Server standard. The EPSG CRS database is available at http://www.epsg.org - European Petroleum Survey Group Geodesy Parameters [EPSG CRS].

Because of the way that angle directions are measured in the AI domain (North corresponds to 0°, East to 90°, etc.), the OGC:CRS 84 is not appropriate for AIXM 5.1/GML data sets that contain arcs of circle defined by start angle/end angle. This will be explained in section 5.2.4.1 of this document. Therefore, it is generally recommended that the EPSG:4326 CRS is used in AIXM 5.1 data sets, which use the WGS-84 reference datum required by ICAO. However, this does not exclude the use of other CRS when appropriate.

Recommendations for the use of CRS references in GML data sets are provided in the OGC Recommendation Paper “URNs of definitions in ogc namespace” [OGC URN], which provides the following URN identifier for the EPSG:4326 CRS: ***urn:ogc:def:crs:EPSG::4326***.

When applied to the encoding of a surface in AIXM, this will give the following GML element:

```
<aixm:Surface gml:id="S01" srsName="urn:ogc:def:crs:EPSG::4326">
```

Note that it is not required to also specify the srsDimension attribute, because it is implicit from the srsName. Specifying the srsDimension could lead to discrepancies, such as using srsName=”epsg:4326” and srsDimension=”3”. People might think that this is a good way to describe 3D WGS84 coordinates. But this is wrong and an appropriate 3D srsName should be used in that case, such as EPSG::4979.

3.2 Use of global srsName

For all geometries (Surface, ElevatedPoint, etc.) a CRS must be specified. The CRS is either defined directly on the geometry element using the srsName attribute or it is derived from the larger context this geometry is part of.

For convenience in constructing feature and feature collection instances, the value of the srsName attribute on the gml:Envelope shall be inherited by all directly expressed geometries in all properties of the feature or members of the collection, unless overruled by the presence of a local srsName. The gml:Envelope is a child element of the

gml:boundedBy property of the feature, as indicated in Figure 1. Thus it is not necessary for a geometry to carry a srsName attribute, if it uses the same coordinate reference system as given on the gml:boundedBy property of its parent feature.

Inheritance of the coordinate reference system continues to any depth of nesting, but if overruled by a local srsName declaration, then the new coordinate reference system is inherited by all its children in turn.

Notwithstanding this rule, all the geometries used in a feature or feature collection may carry srsName attributes, in order to indicate the reference system used locally, even if they are the same as the parent. A geometry without srsName derives its CRS from its closest ancestor that has an srsName. Because of the way that AIXM is built, this can only be one of the following:

- aixm:Surface
- aixm:Curve
- aixm:Point

If no such ancestor is found, it derives its CRS from the srsName of the gml:Envelope in the boundedBy element of the feature or feature collection in which the geometry is contained. Geometries for which no CRS can be derived are invalid.

Formatted: Space After: 0 pt

Comment: Warwick: It would be nice to have a list of the 'highest' children of TimeSlice that have srsName attributes for all corresponding AbstractAIXMFeature subsitutions.

Formatted: Bullets and Numbering

Deleted:

Formatted: Space Before: 12 pt, After: 12 pt

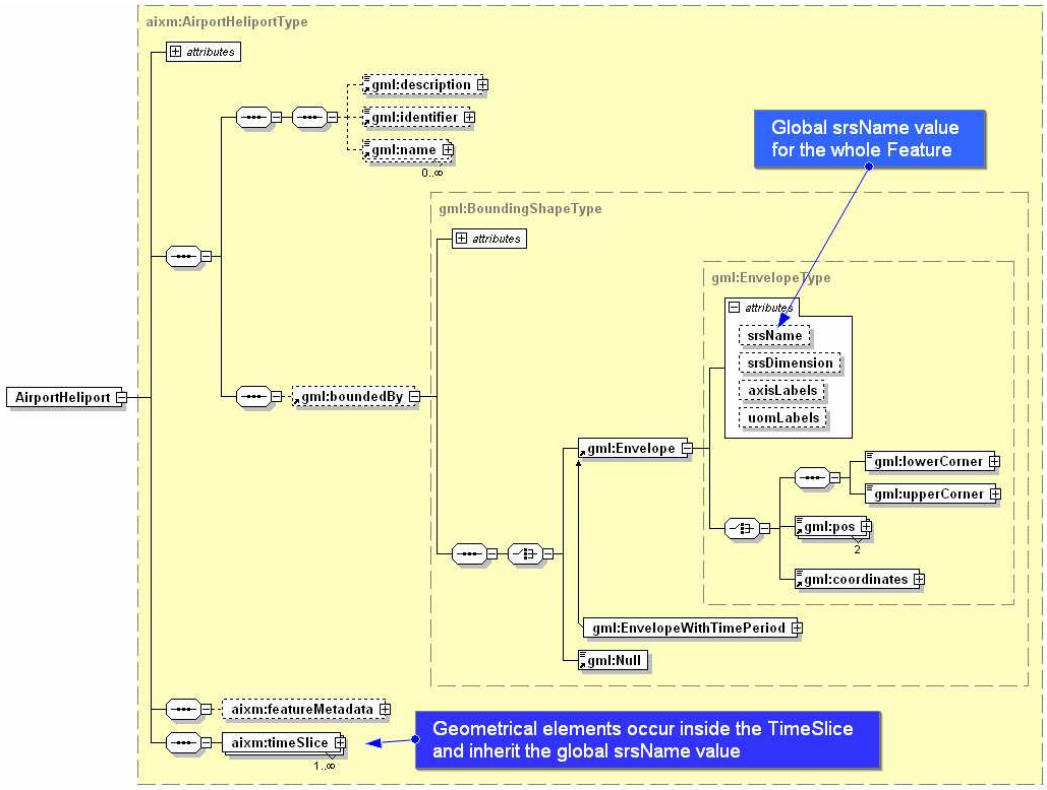


Figure 1 - Global srsName specified using gml:Envelope

4 Positions

4.1 Background

Simple positions are used in order to indicate the geographical location of an airport reference point, navaid, waypoint, runway threshold, etc. In AI publications, they are expressed as a pair of latitude/longitude coordinates. The information about the geodetic reference datum is usually not provided for each individual position, being specified once for the whole AIP.

Formatted: Heading 2;h2;sub-clause 2;H2

4.2 GML encoding

Formatted: Heading 2;h2;sub-clause 2;H2

In AIXM 5.1, simple positions are encoded using the aixm:Point or aixm:ElevatedPoint elements, which are extensions of the gml:Point, as in the following example:

```
...
  <aixm:ElevatedPoint srsName="urn:ogc:def:crs:EPSG::4326" gml:id="ID55">
      <gml:pos>52.2889 -32.0350</gml:pos>
  </aixm:ElevatedPoint>
...
```

Note that the EPSG:4326 CRS has latitude as the primary axis, which indicates that the order of the values in the gml:pos element is ***first latitude, then longitude***. This convention is very important for the correct interpretation of the data and it is not the same for all CRS! For example, the OGC:CRS84 has longitude as the primary axis. Therefore a GML data set that use the OGC:CRS84, the first value in the gml:pos element will indicate the longitude!

The convention "first latitude, then longitude" is also widely observed in the AI domain, as it can be seen in the example of the prohibited area definition from section 0 below. Not surprisingly, the WMS specification has a note that reads: "*users in the international aviation and marine sectors may expect latitude to be before longitude, and a different coordinate display may have safety implications, especially in an emergency response situation*" and "*developers of user interfaces for WMSs are cautioned that all references to latitude and longitude, for example user input of bounding box or readout of cursor coordinates, should show latitude before longitude*".

Deleted: *Web Map Services*

For GML encoding, the latitude/longitude data needs to be expressed in numerical (degrees with decimals) format. If this is not the case with the originator data, a data format conversion should take place. This conversion should be done with a sufficient number of decimal values in order to preserve the original precision of the data. In order to avoid the introduction of small imprecision due to such format conversions, data originators shall be asked to send the data in the raw numerical format (degrees with decimals) that is typically used for survey and geodetic calculations.

For information, note that in the previous AIXM 4.5 version (not based on GML) positional data was encoded using AIXM-defined XML elements, where WGE designates the WGS-84 reference datum:

```
…
   <geoLat>52.2889</geoLat>
   <geoLong>-32.0350</geoLong>
   <codeDatum>WGE</codeDatum>
…
```

Formatted: Left

Formatted: Normal

5 Lines and Surfaces

5.1 Background

Formatted: Heading 2;h2;sub-clause 2;H2

Certain features in the AI domain have polygonal shape (such as Airspace) or linear shape (such as a power line VerticalStructure). They are typically published (for example, in Aeronautical Information Publications – AIP) as a series of latitude/longitude positions, such as in the following example:

EAP 25 (The Castle)
52°11'08.00"N 005°12'30.00"E;
52°12'22.00"N 005°17'15.00"E;
52°11'21.00"N 005°17'56.00"E;
52°10'09.00"N 005°17'56.00"E;
(then along the parallel to) 52°10'09.00"N 005°13'11.00"E;
to point of origin.

Deleted: *EHP*

Deleted: *Drakensteijn*

Usually, it is not indicated in the AI source documents what kind of interpolation is used for the curve between the consecutive points, but it is generally assumed that:

- If two consecutive points have the same latitude value, then the line connecting the two points is a parallel on the surface of the Earth; this may be explicitly stated using words such as “along the parallel to”;
- Otherwise, it is considered a “straight line on the map”.

In addition, the map used when the airspace was designed is typically unknown.

Arcs of circle are also used in the definition of airspace borders, such as in the following examples:

Deleted: typically

EHR 4A (VLIEHORS) TSA
53°10'12.59"N 004°46'21.14"E; along clockwise arc (radius 8 NM, centre 53°15'00.00"N 004°57'00.00"E) to 53°07'01.98"N 004°56'02.41"E; 53°11'00.00"N 004°51'24.00"E; to point of origin.

EHR 4B (VLIEHORS)
53°09'43.06"N 005°06'58.79"E; 53°02'40.00"N 005°15'00.00"E; 52°58'09.00"N 005°06'22.00"E; 53°07'01.98"N 004°56'02.41"E; along anti-clockwise arc (radius 8 NM, centre 53°15'00.00"N 004°57'00.00"E) to point of origin.

Arcs may also be used in the definition of approach/departure trajectories. However, specific “path and terminator” codes are used to encode such arcs and the use of GML in this case is limited to providing a curve for printing the procedure on a map. GML is not used to encode the real flight trajectory of an aircraft, as stored in the Flight Management System (FMS).

5.2 GML encoding

5.2.1 Straight lines

It is recommended that gml:GeodesicString is used as default encoding for straight lines, for the following reasons:

- a geodesic interpolation is the mathematical generalization of the notion of "straight line" on the surface of the Earth;
- the result of the interpolation on the surface of the Earth does not depend on the CRS used; by contrast, a linear interpolation would result on different curves on the surface of the Earth, depending on the CRS used (in fact, this will be exploited in order to encode lines along a parallel, see further down).

Surfaces are encoded in GML using gml:PolygonPatch elements. The pairs of lat/long coordinates can be encoded as either a sequence of gml:pos or, more compact, using a gml:posList element:

```
...
<aixm:Surface gml:id="S01" srsName="urn:ogc:def:crs:EPSG::4326">
  <gml:patches>
    <gml:PolygonPatch gml:id="PP01">
      <gml:exterior>
        <gml:Ring>
          <gml:curveMember>
            <gml:Curve gml:id="C001">
              <gml:segments>
                <gml:GeodesicString>
                  <gml:posList>52.18556 5.20833 52.20611 5.2875 52.18917 5.29889 52.16917 5.29889 52.18556 5.20833</gml:posList>
                </gml:GeodesicString>
              </gml:segments>
            </gml:Curve>
          </gml:curveMember>
        </gml:Ring>
      </gml:exterior>
    </gml:PolygonPatch>
  </gml:patches>
</aixm:Surface>
...
```

Note that the first latitude/longitude pair in a posList is equal to the last one, which is mandatory[1] for a closed LinearRing in GML. Also note that the same separator (space) is used both between the latitude and longitude values (coordinate separator) and also between the latitude/longitude groups (tuple separator).

[1] Note that in GML 3.3 there are new compact encodings for geometry primitives, such as SimplePolygon that do not require the repeated last coordinate. However, these are not available yet in AIXM 5.1, which uses GML 3.2.1.

Formatted: Heading 2;h2;sub-clause 2;H2

Formatted: Heading 3;h3;sub-clause 3;H3;hd3

Deleted: To keep it simple, it is recommended that linear interpolations are used by default for straight lines that are part of gml:Surface or gml:Curve elements. This results in a curve that is neither a geodesic on the ellipsoid nor a great circle (because the earth model in this case is an ellipsoid not a sphere). If a precise interpolation needs to be specified, GML provides dedicated elements, such as gml:Geodesic. However, for short Airspace border segments (less than 50 km), the difference between a linear interpolation and a geodesic is not operationally significant. Is this correct?

Formatted: Bullets and Numbering

Deleted: <gml:LinearRing gml:id="LR01">¶
<gml:posList>56.007778 11.282222 56.111944 11.173889 56.25 11.4 56.25 11.6 56.02 11.793333 56.958333 11.793333 56.007778 11.282222</gml:posList>¶
</gml:LinearRing>¶

Formatted: Space After: 0 pt

Formatted: Normal, Space Before: 12 pt

Deleted: The srsDimension attribute of the parent aixm:Surface element is very important from this point of view because it indicates explicitly the number of values (i.e. equal to the number of coordinate axes) that define the position. For a 2D CRS, such as EPSG:4326, this shall be 2 – latitude and longitude only. In a 3D CRS, a posList could have 3 values for each group: latitude, longitude and elevation.¶
If a precise interpolation needs to be specified, GML provides dedicated elements, such as in the example below, where a geodesic interpolation is used for the curve.¶

```
....¶
<gml:Curve>¶
<gml:segments>¶
<gml:Geodesic interpolation="geodesic">¶
<gml:posList>11.282222 56.007778 11.173889 56.111944</gml:posList>¶
</gml:Geodesic>¶
</gml:segments>¶
```

... [1]

5.2.2 Parallels

In the AI domain, if an airspace border has two consecutive points at the same geographical latitude, it is assumed that the line between the two points is "along the parallel". This can be encoded in AIXM/GML using "linear" GML elements in combination with a geodetic CRS, such as EPSG:4326. The linear interpolation in a 2D geodetic CRS between two points that have the same latitude corresponds to a parallel on the Earth's surface. This is shown graphically in Figure 2.

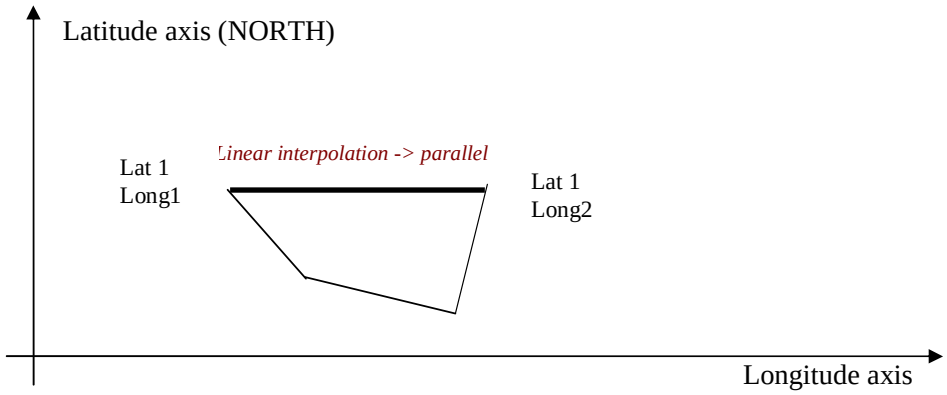


Figure 2 - Linear interpolation in a geodetic CRS

It shall be noted that the current GML 3.2.1 does not allow general "rhumbline" (constant angle with the meridians) interpolations to be specified directly (e.g. using a "RhumbLine" element or interpolation. However, linearly interpolated lines in certain conformal projections are realized as rhumblines on the ellipsoid Earth model.

For example, a LineString with two coordinates (i.e. a line segment with begin and end point) may be used with a srsName that references a Mercator projection (e.g. EPSG:3395), which is a well supported conformal projection. Note that the LineString element implies that linear interpolation must be used and srsName="urn:ogc:def:crs:EPSG::3395" implies that the interpolation is done in the Mercator projection plane. Hence this geometry gets realized as a rhumbline on the WGS84 ellipsoid Earth model. Other conformal stereographic and conical projections can also be used to represent rhumblines on the earth ellipsoid model of WGS84 (this may be useful for regions of interest near the poles).

A change request has been raised with OGC (see Annex B) in order to enable specifying additional interpolations, such as rhumbline. However, this is mostly a theoretical issue since parallels are the only type of rhumbline extensively used in the AI domain. Therefore, it is not considered necessary to support the encoding of rhumblines, other than parallels, in AIXM 5.1.

Deleted: Encoding p

Formatted: Heading 3;h3;sub-clause 3;H3;hd3

Formatted: Bullets and Numbering

Deleted: when

Deleted: with

Deleted: drawn

Deleted: does not require any special annotation

Deleted: , when

Deleted: represented

Comment: Warwick: Do you mean 'straight' lines (ie linearly interpolated)? If so then this is not true for the conformal Azimuthal Stereographic Projection. As far as I know Mercator is the only projection where straight lines are loxodromes.

Deleted: all

Deleted: a conformal projection

Comment: Warwick: Please see previous comment

Comment: I think that this was answered by David Burggraf, who confirmed that the text is correct.

Deleted: copied below

Formatted: Space Before: 0 pt

5.2.3 Arc by edge

This is a relatively simple case as it is represented by the element gml:Arc in GML, which does not have any ambiguities: 3 points always define a single arc. Unfortunately, this is rarely used in the AI domain. When moving towards a fully digital AI chain, the use of this type of arc information should be encouraged and eventually imposed as the unique way for defining arcs. However, this is not likely to be achieved on short term.

A border that uses arcs by 3 points looks like in Figure 3 - Arc by edge point:

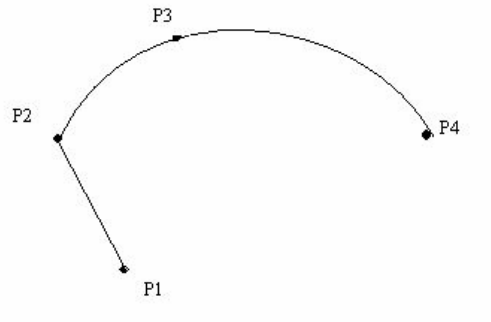


Figure 3 - Arc by edge point

A GML encoding example for this type of arcs is provided below.

```
...
<gml:PolygonPatch>
  <gml:exterior>
    <gml:Ring gml:id="…">
     …
      <gml:curveMember>
        <gml:Curve gml:id="…">
          <gml:segments>
            <gml:Arc gml:id="…">
              <gml:pos>P2</gml:pos>
              <gml:pos>P3</gml:pos>
              <gml:pos>P4</gml:pos>
            </gml:Arc>
          </gml:segments>
        </gml:Curve>
      </gml:curveMember>
    ...
```

In fact, this is a particular case of the more general GML concept of "ArcString", which is a curve segment that uses three-point circular arc interpolation in a piecewise fashion to "string" the arc segments together. The number of control points in the string is *(2 x numArc)+1*, where numArc is the property defining the number of the arcs in the string, such as in Figure 4 - ArcString in ISO 19107

Formatted: Bullets and Numbering

Deleted: and

Formatted: Left

Deleted: *x*

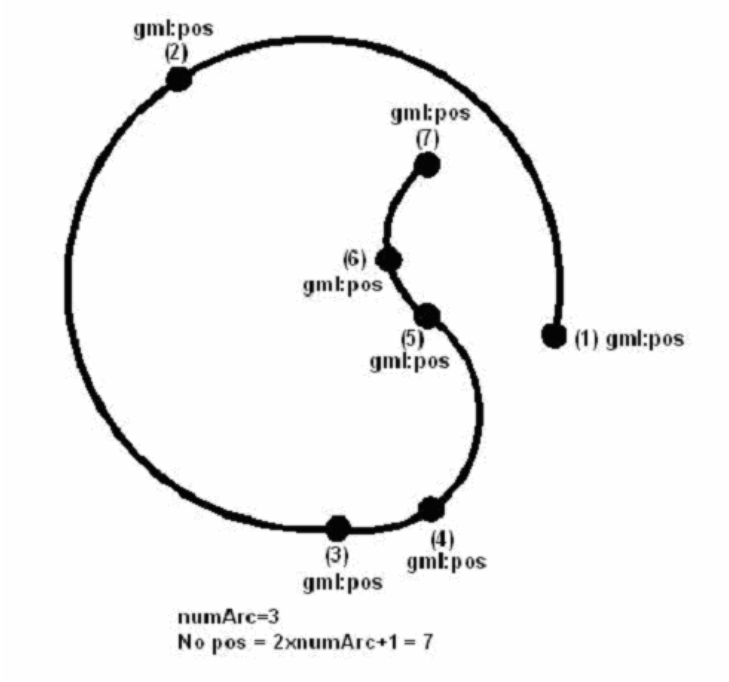


Figure 4 - ArcString in ISO 19107

For information, the AIXM 4.5 way of encoding the arc by edge point using AIXM-defined elements is provided below:

```
...
<Avx>
   <codeType>ABE</codeType>
   <geoLat>lat-P2</geoLat>
   <geoLong>long-P2</geoLong>
   <codeDatum>WGE</codeDatum>
   <geoLatArc>lat-P3</geoLatArc>
   <geoLongArc>long-P3</geoLongArc>
</Avx>
...
```

5.2.4 Arc by centre point

Currently, this is the typical construct for arcs used in the definition of airspace borders in the AI domain. It comes with two problems:

- The arc is over specified, as the start/end points, the centre and the radius are all provided. Typically, the calculated distance from the centre to the start and end point is not quite the same due to round-off error and is also usually different from the radius;
- This construction of arcs is not supported exactly this way in GML and in GIS systems in general.

Formatted: Heading 3;h3;sub-clause 3;H3;hd3

Formatted: Bullets and Numbering

Deleted: T

The closest GML construct that can be used for encoding this type of arcs is ArcByCenterPoint. This requires calculating the start/end angles from the centre to the start/end points. Before calculating these angles, it is important to specify the angle measuring convention in AIXM, as GML seems to leave some degree of interpretation for this aspect.

Formatted: Bullets and Numbering

5.2.4.1 Measuring angles in GML

GML explicitly implements[2] the semantics of ISO 19107. The "startOfArc" (6.4.15.5) and "endOfArc" (6.4.15.6) are defined in terms of bearings. The definition of "bearing" is provided in Section 6.3.12, which states that "*Bearing is a data type used to represent direction in the coordinate reference system. In a 2D coordinate reference system, this can be accomplished using a "angle measured from true north" or a 2D vector point in that direction.*"

There are two variants mentioned in ISO 19107 for expressing bearings: angle and direction. The semantics for angle is given in 6.3.12.2 of the same document: "*In this variant of Bearing usually used for 2D coordinate systems, the first angle (azimuth) is measured from the first coordinate axis (usually north) in a counterclockwise fashion parallel to the reference surface tangent plane.*"

Although it may not be obvious, this definition matches the needs of the AI domain, as angles are usually expressed in degrees measured clockwise from the True North. The diagrams below explain why the "counter clockwise" convention stated in the ISO 19107 standard, when combined with the EPSG:4326 geodetic CRS actually corresponds to a clockwise rotation in the AI domain.

Deleted: s

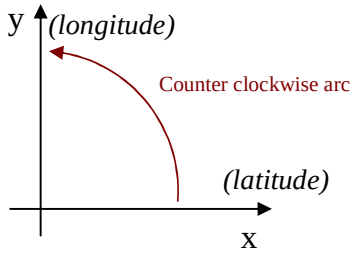


Figure 5 - Angle measured in a 2D geodetic CRS that has latitude as first axis

In the EPSG:4326 CRS, the first (x) axis is latitude and the second (y) axis is longitude. A counter clockwise angle means measuring it from the first axis towards the second

[2] Note that this is true for classes of spatial primitives (e.g. GM_Arc) but in general not their operations. In the ISO 19107, startOfArc and endOfArc are operations, aka constructors which derive other values from the GM_Arc. However, GML/ISO 19136 defines the UML Class ArcByCenterPoint (Figure D.48) in a GML profile of ISO 19107 where startOfArc and endOfArc are attributes. Although not explicitly stated in the GML standard, we can infer that the ISO 19107 semantics for operations carry over to the identical semantics for the corresponding attributes with the same name in the GML profile of ISO 19107 (ISO 19136, Annex D).

axis. When transposing this coordinate system on the surface of the Earth, this corresponds to a clockwise rotation from the first axis (North) in order to measure angles, as shown in Figure 6. Practically, the x/y reference system is mirrored and rotated to be aligned with the meridians and the parallels.

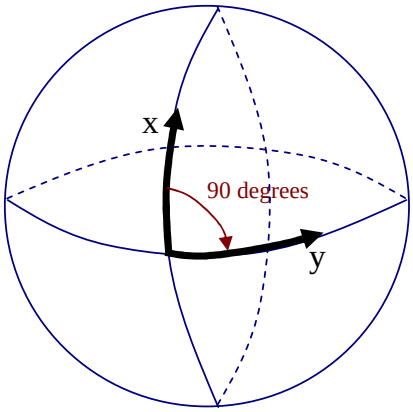
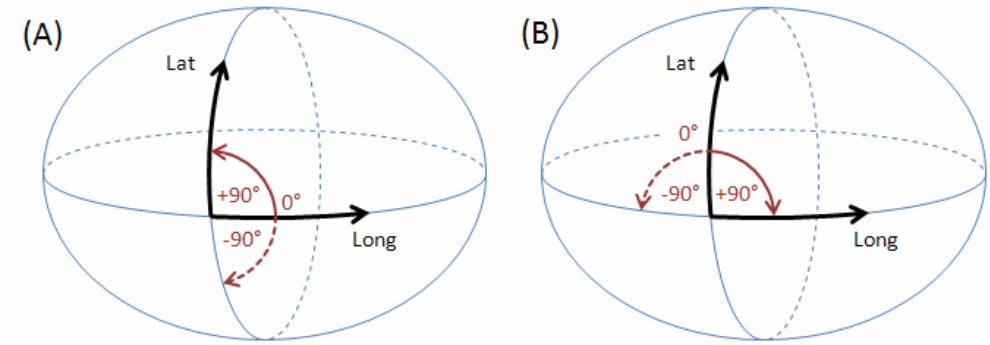


Figure 6 - Coordinate System Axes of EPSG:4326 CRS mapped on the Earth's surface

Therefore, when the EPSG:4326 CRS (or another 2D geodetic CRS that has latitude as first axis) is used, this translates to angles that are measured clockwise starting from the True North in the AI Domain. East is at 90 degrees from North, South at 180 degrees from North, etc. This convention is important for defining the startAngle and endAngle of the ArcByCenterPoint. Note that it is also possible to use negative values for angles. Negative angles are measured from the first axis through rotation in the direction opposite to the second axis.

The following diagram shows how angles are measured in WGS 84 2D with different coordinate systems:



(A) GeodeticCRS: urn:ogc:def:crs:OGC:1.3:CRS84

- Datum: WGS84
- Ellipsoidal 2D CS.

- Axes: (1st) longitude, (2nd) latitude.
- Orientations: east, north. UoM: degree

(B) GeodeticCRS: urn:ogc:def:crs:EPSG::4326

- Datum: WGS84
- Ellipsoidal 2D CS.
  - Axes: (1st) latitude, (2nd) longitude.
  - Orientations: north, east. UoM: degree

The order of the axes determines where 0° is located (on the positive part of the coordinate system's first axis).

Formatted: Bullets and Numbering

5.2.4.2 Arc direction

Once the startAngle and endAngle are known, it is still necessary to establish how the arcs are interpolated/drawn. The semantics of the words "start" and "end" indicate that arcs shall be interpolated/drawn from the start angle to the end angle, similarly to a line that is always interpolated/drawn from its start to its end. However, this still leaves some room for interpretation, e.g. an arc that has startAngle=90 (East) and endAngle=180 (West) should be interpolated/drawn through South or through North?

The following convention shall apply in the aviation domain***: if the start angle is smaller than the end angle then the arc direction is the direction in which the angle values increase***. If the opposite is true, than the arc direction is the one in which the angle values decrease. Depending upon the coordinate system that applies to a given ArcByCenterPoint, this results in a clockwise (left-handed system) or counter-clockwise (right-handed system) directed arc. The arguments for this convention are detailed in Annex A.

Applied to the EPSG:4326 CRS, this means that ***arcs shall always be drawn:***

- ***clockwise on the surface of the Earth when the startAngle is lower than the endAngle;***
- ***counter-clockwise on the surface of the Earth when the startAngle is higher than the endAngle***

The same convention applies to any other geodetic CRS that has latitude as first (x) axis. This is exemplified in Figure 7. It is therefore possible to define arcs in both clockwise and counter-clockwise direction by choosing the right startAngle and endAngle values. This also requires the use of angle values between -360 and 360, both values included.

Deleted: Figure 1

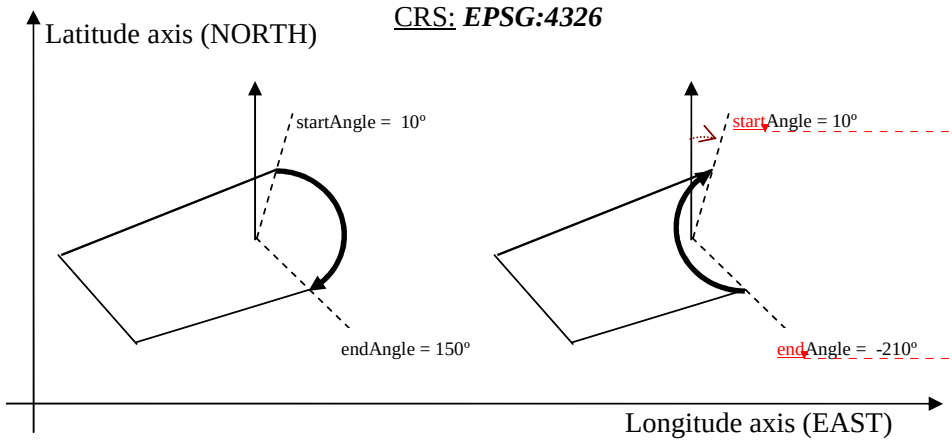


Figure 7 - Arcs are drawn clockwise, from startAngle to endAngle in the AI convention (as illustrated in Figure 6)

5.2.4.3 Arc interpolation

Typically, there are no statements in AI sources, such as State AIP, about how arcs and circles should be interpolated. Such arcs are usually designed using a map (into a specific projection, which is not communicated). Thus, they would represent points of equal distance from the arc centre, on that specific projection. Considering that the airspace design processes always include buffers in the definition of an area that contains a dangerous activity, the relative imprecision of an arc or circle interpolation is not operationally significant. However, it is important to have a common interpretation of how arcs and circles should be interpolated when defined in GML "by center point".

Therefore, it is recommended to interpolate ArcByCentrePoint through points situated at equal geodesic distance (the radius if the arc) from the arc centre. This makes the interpolation independent of the CRS used.

5.2.4.4 Mapping for ArcByCenterPoint

This section indicates how to calculate the start/end angles when the information provided by the (official) source is in the form of start/end positions, centre and radius. The situation presented in Figure 8 exemplifies a part of an Airspace boundary that is a sequence of:

- a straight segment (P1-P2)
- followed by a clockwise arc from P2 to P4 with the centre in P3
- followed again by a straight line to P5.

Deleted: end

Deleted: start

Formatted: Bullets and Numbering

Formatted: Normal

Formatted: Underline

Comment: Warwick: I'm bothered by the choice of CRS used when arc drawing. My understanding is that the Arc is interpolated in 2D Euclidean space and then interpolated points are mapped to the surface of the Ellipsoid according to the CRS. If this is the case then I would have thought you have to be very careful about the choice of CRS for the arc to be meaningful on the surface. Beware that if we have to specify a different CRS than EPSG:4326 to get a meaningful interpolation then the start/end angle semantics may change so that when going from 10 deg to 20 deg it may be anti-clockwise!

Comment: Warwick, does the new section about Arc Interpolation answer your concern? It says that the arc interpolation is done by finding points of equal geodesic distance from the centre, therefore it's independent from the CRS that is use.

Formatted: Bullets and Numbering

Deleted:

Formatted: Indent: Before: 0,74 cm, Hanging: 0,63 cm, Space Before: 3 pt, After: 0 pt, Bulleted + Level: 1 + Aligned at: 0,74 cm + Tab after: 1,38 cm + Indent at: 1,38 cm

Deleted: ,

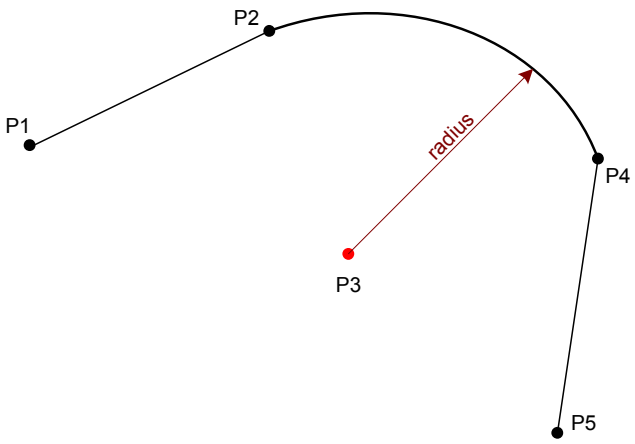


Figure 8 - Mapping arc by start/end point into ArcByCenterPoint

Sometimes, the arc drawn using the exact centre points and radius value provided by the originator does not exactly match. The radius is usually a rounded value, which may be different up to 10% from the calculated distances from the centre P3 to points P2 or P4. However, as one objective of the GML encoding is to preserve as much as possible the original (official) information, it is recommended that the original centre and radius data is used directly for the ArcByCenterPoint in the AIXM/GML encoding. The startAngle shall be calculated using the vector P3-P2 and the endAngle shall be calculated using the vector P3-P4. The resulting arc is represented in Figure 9. This shows that it is possible that the resulting GML arc does not align at points P2 and P4, because of the imprecision of the original centre and/or radius data.

Deleted: the

Deleted: ,

Deleted: which also

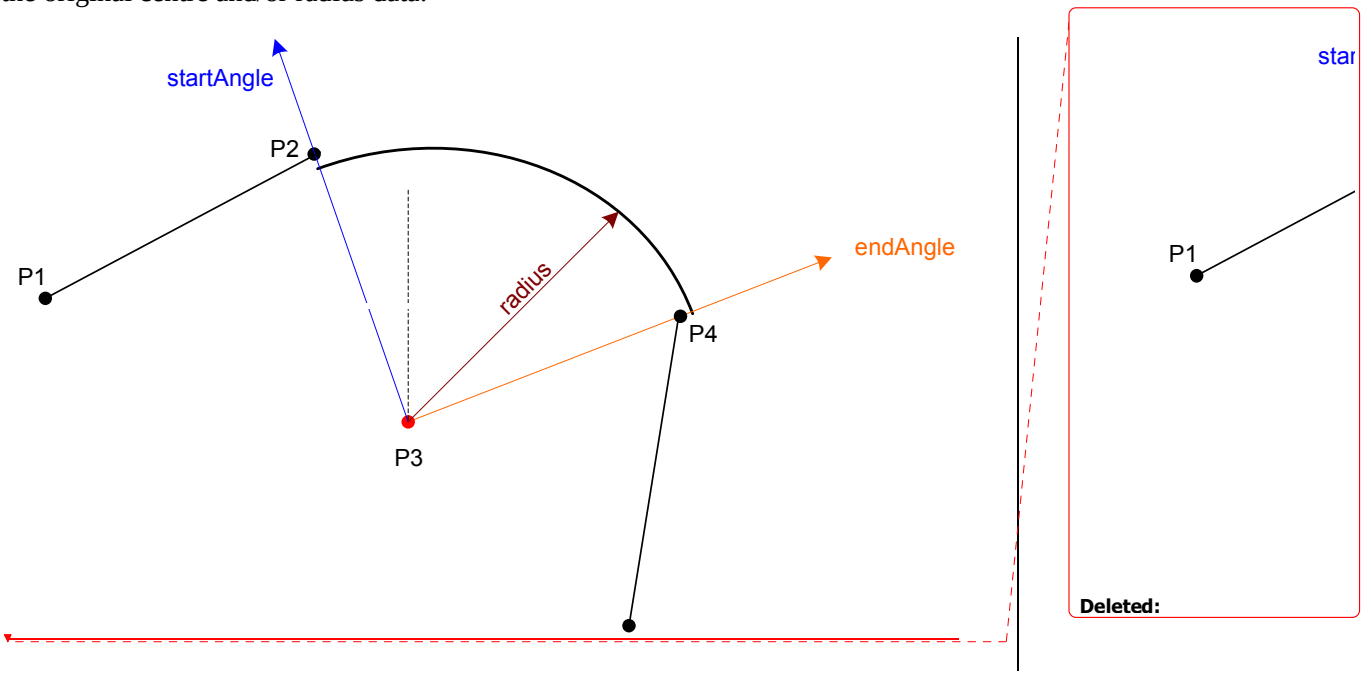


Figure 9 - Calculating startAngle and endAngle for ArcByCenterPoint

If the potential misalignment of the arc (as indicated in Figure 9) is a problem for an application, then it is suggested that, for example, the original centre and radius data are replaced with calculated values, by that application, as indicated in Figure 10.

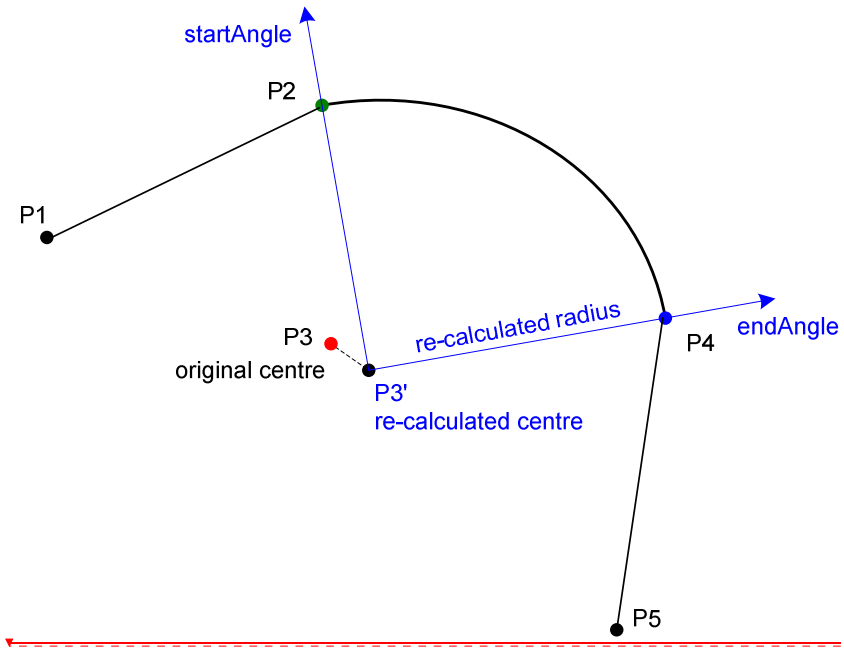


Figure 10 – Re-calculating the whole arc so that it generates a closed surface

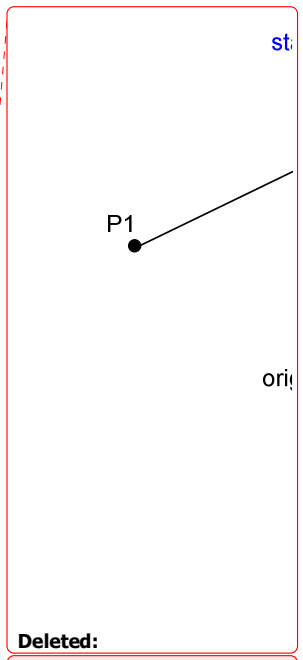
The objective is to find a point that is at equal geodesic distance from the start P1 and end P2 points. This can be done with an algorithm that iteratively tries to find the point of which the distance to both P1 and P2 is the same and which is as close as possible to the original center. However, if the difference between the original distances P3-P2 and P3-P4 is higher than 1%, then the application shall raise an error message and abort the calculation. This rule was also specified in previous AIXM versions (4.5) and it did prevent the provision of geometrically inconsistent arc data.

The calculated point (P3') can then used as the corrected center point. The corrected radius is the distance from the corrected center P3' to P1 (because of the algorithm applied, this distance is also the distance to P2). The start/end angles are the bearings from the corrected center P3' to P1/P2.

There might be situations where the centre P3 is an Aerodrome Reference Point (ARP) or a Navaid. In this situation, it might be more appropriate to re-calculate the points P2 and P4 in order to correctly close the surface. This could be problematic when P2 or P4 are also ends of arcs. The best solution depends on the intended use of the data, therefore this is left for decision by application developers.

An example of the GML encoding for an ArcByCenterPoint is presented below.

...



Deleted:

Comment: New text as proposed by Bert Robben.

Comment: Warwick: I would have thought that he CRS used will affect the result of these calculations and so CRS choice is critical unless geometry construction is performed using 'Geodesic' primitives which effectivel 'bypass' the CRS mapping.

Deleted: The original centre is projected orthogonally on the perpendicular bisector of the chord between the start/end points and then the radius is computed from this new centre. This ensures that the arc actually passes through the start/end points, while not modifying significantly the original data.

Comment: Warwick, I think that this answers your concern.

Comment: is there any guidance available on how to implement this? Can we provide more detail for such an algorithm,?

Deleted: However, if the difference between the distances P3-P2 and P3-P4 is higher than 1%, then the application shall raise an error message and abort the calculation. This rule was also specified in previous AIXM versions (4.5) and it did prevent the provision of geometrically inconsistent arc data. Then, using the position of the P3 and P ... [2]

```
<aixm:Surface gml:id="S01" srsName="urn:ogc:def:crs:EPSG::4326" srsDimension="2">
  <gml:patches>
    <gml:PolygonPatch gml:id="PP01">
      <gml:exterior>
        <gml:Ring gml:id="R01">
          <gml:curveMember>
            <aixm:Curve gml:id="C01">
              <gml:segments>
                <gml:LineStringSegment gml:id="LSS01">
                  <gml:pos>lat_P1 long_P1</gml:pos>
                  <gml:pos>lat_P2 long_P2</gml:pos>
                </gml:LineStringSegment>
                <gml:ArcByCenterPoint gml:id="A01">
                  <gml:pos>lat_P3 long_P3</gml:pos>
                  <gml:radius uom="m">radius</gml:radius>
                  <gml:startAngle uom="deg">calculated_start_angle</gml:startAngle>
                  <gml:endAngle  uom="deg">calculated_end_angle</gml:endAngle>
                </gml:ArcByCenterPoint>
                <gml:LineStringSegment gml:id="LSS02">
                  <gml:pos>lat_P4 long_P4</gml:pos>
                  <gml:pos>lat_P5 long_P5</gml:pos>
                </gml:LineStringSegment>
                …
              </gml:segments>
            </aixm:Curve>
          </gml:curveMember>
        </gml:Ring>
      </gml:exterior>
    </gml:PolygonPatch>
  </gml:patches>
</aixm:Surface>
…
```

Formatted: Bullets and Numbering

5.2.4.5 Units of measurement

The previous XML encoding example also shows recommended values for the uom attributes of gml:radius and gml:start(end)Angle. According to GML section 8.2.3.7, "*in an instance document, on elements of type gml:MeasureType the mandatory uom attribute shall carry a value corresponding to either:*

Deleted: snd

- *a conventional unit of measure symbol,*
- *a link to a definition of a unit of measure that does not have a conventional symbol, or when it is desired to indicate a precise or variant definition.*

For common units of measurement, such as meter for distances and degrees for angles, it is recommended that conventional units of measure symbols are used. A Note in the [GML] standard suggests the use of UCUM symbols: "*It is recommended that the symbol be an identifier for a unit of measure as specified in the Unified Code of Units of Measure' (UCUM) (http://aurora.regenstrief.org/UCUM). This provides a set of symbols and a grammar for constructing identifiers for units of measure that are unique, and may*

be easily entered with a keyboard supporting the limited character set known as 7-bit ASCII."

The following UCUM "c/s" (case sensitive) values are recommended to be used for gml:radius in AIXM/GML data sets:

- **m** – when the radius is expressed in meters
- **km** – when the radius is expressed in kilometers
- **[nmi_i]** – when the radius is expressed in Nautical Miles

The symbol "**deg**" is recommended to be used for the uom attribute of gml:startAngle and gml:endAngle elements.

5.2.4.6 Old AIXM 4.5 encoding

For information, the same border using a clockwise arc was encoded in AIXM 4.5 as in the example below:

```
...
<Avx>
   <codeType>GRC</codeType>
   <geoLat>lat-P1</geoLat>
   <geoLong>long-P1</geoLong>
   <codeDatum>WGE</codeDatum>
</Avx>
<Avx>
   <codeType>CWA</codeType>
   <geoLat>lat-P2</geoLat>
   <geoLong>long-P2</geoLong>
   <codeDatum>WGE</codeDatum>
   <geoLatArc>lat-P3</geoLatArc>
   <geoLongArc>long-P3</geoLongArc>
   <valRadiusArc>Radius</valRadiusArc>
   <uomRadiusArc>NM</uomRadiusArc>
</Avx>
<Avx>
   <codeType>GRC</codeType>
   <geoLat>lat-P4</geoLat>
   <geoLong>long-P4</geoLong>
   <codeDatum>WGE</codeDatum>
</Avx>
...
```

5.2.5 Circle by center point

Full circles are also used in order to define the geometry of certain airspace in the AI domain. They can be directly encoded using the gml:CircleByCenterPoint. It is quite deep in the structure, as presented in the example below.

```
            ...
                <aixm:Surface gml:id="S001" srsName="urn:ogc:def:crs:EPSG::4326">
```

Formatted: Bullets and Numbering

Formatted: XML code

Formatted: Heading 3;h3;sub-clause 3;H3;hd3

Formatted: Bullets and Numbering

Formatted: XML code, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers, Border: Top: (No border), Bottom: (No border), Left: (No border), Right: (No border)

```
            <gml:polygonPatches>
              <gml:PolygonPatch>
                <gml:exterior>
                  <gml:Ring>
                    <gml:curveMember>
                      <gml:Curve gml:id="CUR001">
                        <gml:segments>
                          <gml:CircleByCenterPoint numArc="1">
                            <gml:pos>51.01555556 2.57138889</gml:pos>
                            <gml:radius uom="[nmi_i]">12</gml:radius>
                          </gml:CircleByCenterPoint>
                        </gml:segments>
                      </gml:Curve>
                    </gml:curveMember>
                  </gml:Ring>
                </gml:exterior>
              </gml:PolygonPatch>
            </gml:polygonPatches>
          </aixm:Surface>
        ...
```

5.2.6 Corridor

Corridors are sometimes used in the definition of airspace geometries. This is done by specifying a centerline and a width or half-width. Such airspace can be encoded using gml:OffsetCurve, which is an implementation of the GM_OffsetCurve.

An offset curve is a curve at a constant distance from the basis curve. It is specified by providing:

- a "baseCurve", which is a reference to the curve from which this curve is defined as an offset;
- a "distance", which is the distance at which the offset curve is generated from the basis curve; in a 2D system, positive distances are to be left of the basis curve, and negative distances are right of the basis curve;
- a "refDirection" (optional), which is used to define the vector direction of the offset curve from the basis curve. It shall not be used in AIXM/GML encodings, where all offset curve are used in 2D cases. In that case, distance defines left side (positive distance) or right side (negative distance) with respect to the tangent to the basis curve.

In fact, the encoding of an airspace corridor requires the use of two gml:OffsetCurve elements and two gml:ArcByEdge, which connect the ends of the two offset curves, as shown in Figure 11. The points P1', P2", P1''' need to be calculated in order to define the closing arc, and similar for the points P2', P2'', P2'''.

Formatted: Default Paragraph Font, Complex Script Font: 10 pt

Formatted: Heading 3;h3;sub-clause 3;H3;hd3

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Normal, Bulleted + Level: 1 + Aligned at: 0,63 cm + Tab after: 1,27 cm + Indent at: 1,27 cm

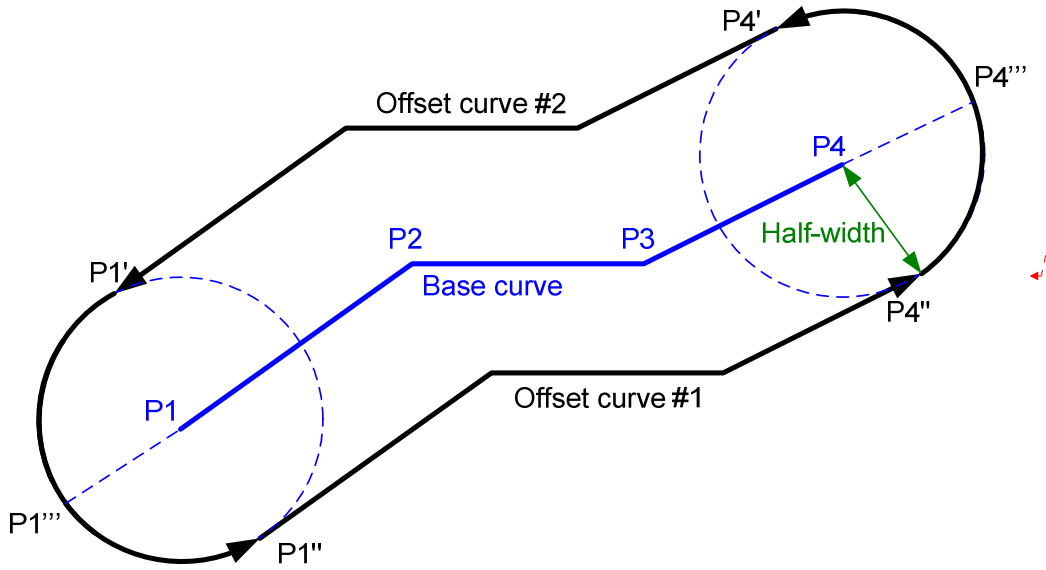


Figure 11 - Airspace corridor encoding

The following GML code sample provides the encoding of the airspace corridor shown in Figure 11.

```
...
To be added
...
```

5.2.7 Perimeter encoding direction

The GML Surface implements ISO 19107 GM_Surface whose exterior boundary must be encoded counter clockwise and any interior boundary encoded clockwise.

For AI Domain applications, this implies that the outside perimeter and any eventual holes shall be encoded as shown in Figure 12. It is unlikely that more than one level of internal patches (rings) would exist for features in the AI Domain.

Formatted: Keep with next

Formatted: Caption, Centered

Formatted: Normal

Formatted: XML code

Formatted: Highlight

Comment: All changes in this section have been proposed by Johannes Echterhoff, who provided the revised text and graphics.

Formatted: Heading 3;h3;sub-clause 3;H3;hd3

Formatted: Bullets and Numbering

Deleted: in the sense of the coordinate system axes of the CRS (as in Figure 4)

Deleted: the

Deleted: The established practice in computer mapping graphics is that, when a 'geographic' CRS is used, the outside perimeter is encoded clockwise, with respect to the Long/Lat ordered axes typically used it this domain. Also, that any eventual inside perimeters are encoded in opposite direction. This is generally consistent with the mathematical definition of inside/outside - draw a line from infinity to point of interest. If number of line crossings is odd, the point is 'inside'.¶
As we recommend the use of the EPSG:4326 CRS for

Deleted: clockwise with respect to the axes shown in Figure 11 and any eventual holes should be encoded counter clockwise.

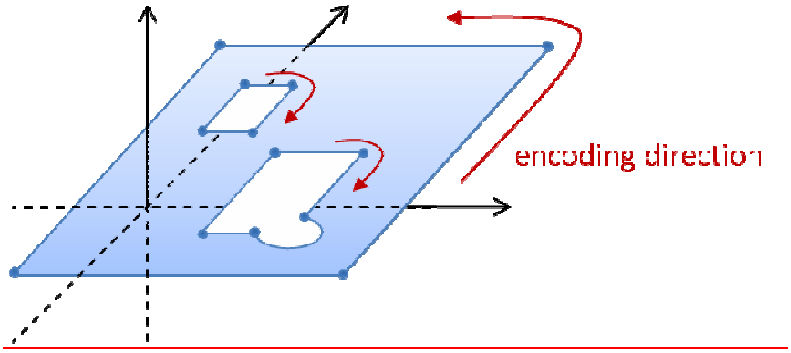


Figure 12 - Encoding direction for surface boundaries

The direction of "clockwise" and "counter-clockwise" do not depend on the axis order of the given CRS. They just depend on where "up" is. In EPSG:4326, "up" is implied and pointing from the center of the earth outwards. When a surface is viewed from the side where "up" is, then – following ISO 19107 - the exterior boundary of the surface shall be encoded counter-clockwise while the interior shall be encoded clockwise.

Accordingly, the encoding of surface boundaries – counter-clockwise for exterior boundary, clockwise for any interior boundary - is the same in both left- and right-handed systems; see the following figure.

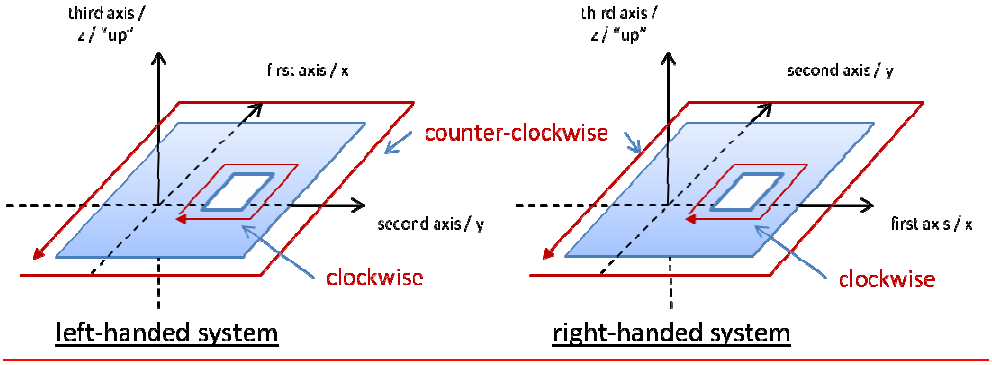


Figure 13 - The boundary encoding is the same in both left- and right-handed systems

Figure 12 shows a surface with two holes, one of which has an arc as part of its boundary. The direction of this arc needs to be in line with the overall orientation of the boundary. This can unambiguously be achieved following the rules laid out in section 5.2.4 and Annex A. The following figure shows some examples where arcs are part of a surface's boundary.

Formatted: Keep with next

Formatted: Caption, Centered

Formatted: Keep with next

Formatted: Caption, Centered

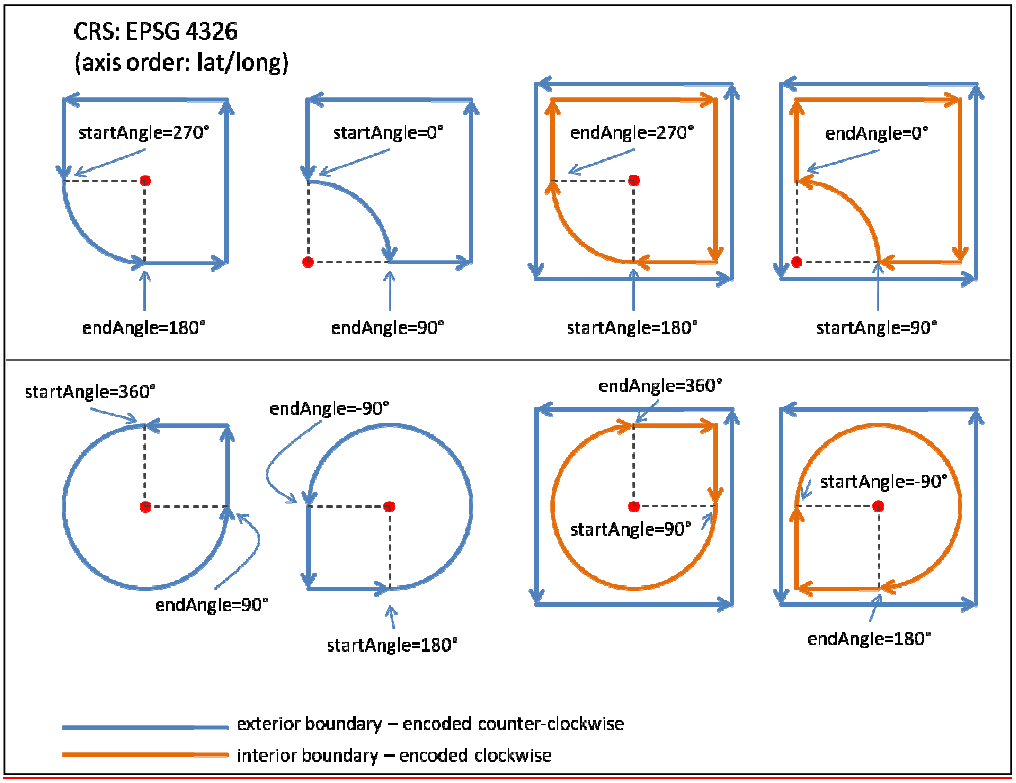


Figure 14 - arcs and their direction when being part of a surface boundary

Note that the gml:OrientableCurve can be used to reuse a curve in its opposite orientation as follows:

```
<gml:Polygon gml:id="PGN01">
<gml:exterior>...</gml:exterior>
<gml:interior>
  <gml:Ring gml:id="RNG01">
    <gml:curveMember>
      <gml:OrientableCurve gml:id="OC01" orientation="-">
        <gml:baseCurve>
          <gml:ArcByCenterPoint gml:id="ACP01">
            <gml:pos>...</gml:pos>
            <gml:radius uom="m">...</gml:radius>
            <gml:startAngle uom="deg">...</gml:startAngle>
            <gml:endAngle  uom="deg">...</gml:endAngle>
          </gml:ArcByCenterPoint>
        </gml:baseCurve>
      </gml:OrientableCurve>
    </gml:curveMember>
    <gml:curveMember>
        <gml:LineString gml:id="LS01">
          ...
```

Formatted: Keep with next

Formatted: Caption, Centered

Deleted: ¶

Deleted: <sp>¶
Figure 11 - Encoding direction for outer patch (ring)¶
In this diagram, it is important to note that the arc of the circle of the inner patch will still have to be encoded clockwise, but used counter-clockwise in the interior ring of a Polygon!

Comment: Is OrientableCurve still necessary? The latest convention for arcs, using angles between -360 and +360 seems to cover all necessary arc direction encodings. I would propose to take this out because supporting OrientableCurve in the aviation profile would mean extra effort for the implementers.

```
            </gml:LineString>
        </gml:curveMember>
        <gml:curveMember>
            <gml:LineString gml:id="LS02">
              ...
            </gml:LineString>
        </gml:curveMember>
      </gml:Ring>

    </gml:interior>
</gml:Polygon>
```

The OrientableCurve is simply a means to invert the direction of a given curve - see ISO 19107. If one or more segments of a curve wrapped in an OrientableCurve element are ArcByCenterPoints, then the direction of each arc is implicitly inverted. For ArcByCenterPoint, this means that the start and end angles are implicitly switched - similar to lines in which the end point becomes the start point and the line is traversed from that point to the original start point.

5.2.8 „Other geometries – for procedure design

"WindSpiral" used in procedure design, but does not have GML correspondent?

"SegmentLocus" -

"ArcLocus"

Deleted: The same would apply in the case of the Airspace border shown in Figure 11.¶
¶
¶
Figure 12 - "Concave" arc drawn conter-clockwise, in the same direction as compared with the orientation of the exterior boundary of the surface

Formatted: Highlight

Formatted: Heading 3;h3;sub-clause 3;H3;hd3

Formatted: Bullets and Numbering

Formatted: Normal, Left

Formatted: Check spelling and grammar

6 Airspace aggregation

6.1 Background

There are two main ways to describe the geometry of an airspace in the AI Domain:

- by specifying a horizontal border and vertical limits;
- by providing a composition rule by which the airspace is defined as a series of unions, intersections, subtractions of other Airspace, such as in the following examples:
  - Airspace of type CTR defined as a "*circle of 50 NM from which the portion of airspace situated in a neighboring FIR is subtracted*";
  - Airspace of type UIR that has "*the same horizontal projection as an FIR*", but different vertical limits;
  - Airspace of type CTA which is the result of aggregating some Airspace of type SECTOR;
  - etc.

The AirspaceVolume class of the AIXM 5 model was designed in order to support the encoding of such aggregated Airspace. Note that the "operation" property of the AirspaceGeometryComponent needs to be used, which steps outside the GML world. Although GML supports the creation of composite surfaces using "patches", it is not possible to leave this aggregation to the GML level because the vertical limits of the different components might be different.

The possibility of using 3D geometries might be considered in future. Until then, using only 2D GML components requires custom processing of AIXM/GML files in order to correctly represent the result of an airspace aggregation, in particular the use of the operation and operationSequence attributes of the AirspaceGeometryComponent class.

6.2 GML encoding

The model gives the possibility for using several approaches for the encoding of airspace aggregations/dependencies. The use of a particular method, from the ones described further in this section, depends on the intended use of the data.

In order to exemplify these methods, the example of an Airspace of type "CTA" will be used. The aggregation hierarchy is presented in .

Formatted: Bullets and Numbering
Formatted: Heading 2;h2;sub-clause 2;H2
Formatted: Normal, Bulleted + Level: 1 + Aligned at: 0,63 cm + Tab after: 1,27 cm + Indent at: 1,27 cm
Formatted: Font: Italic, Complex Script Font: Italic
Formatted: Font: Italic, Complex Script Font: Italic
Formatted: Not Highlight
Formatted: Not Highlight
Formatted: Not Highlight
Formatted: Not Highlight
Formatted: Not Highlight
Formatted: Not Highlight
Formatted: Not Highlight
Formatted: Heading 2;h2;sub-clause 2;H2
Formatted: Bullets and Numbering
Formatted: Not Highlight
Formatted: Normal
Formatted: Not Highlight
Formatted: Not Highlight
Formatted: Highlight

6.2.1 By reference

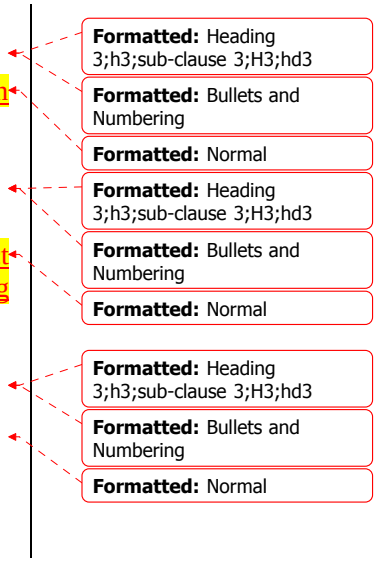
The first method is limited to referring For example, to be used for data provision between synchronized databases, such as between a local and a regional database.

6.2.2 By copying the geometry

To be used for data provision to applications that need full geometrical data for direct consumption. All the tree needs to be copied, as there might exist cascading dependencies.

6.2.3 Combined method

Really needed?



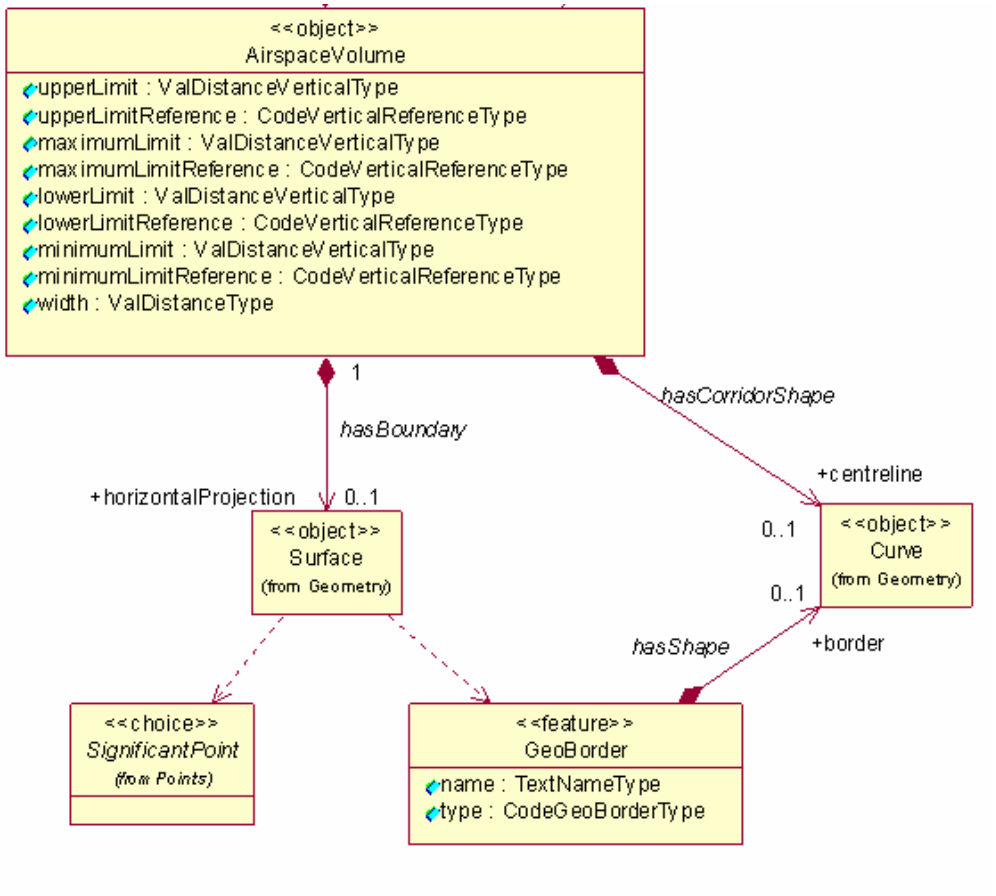
Formatted: Bullets and Numbering

7 Point references and annotations

Formatted: Heading 2;h2;sub-clause 2;H2

7.1 Background

Positions of Navaids or Designated Points are occasionally used in the AI Domain in order to define the shape of an Airspace. They can be used as arc centers or even as boundary points. This is represented in the UML model of AIXM as a "dependency" associations between Surface and SignificantPoint/GeoBorder:



Formatted: Left

Another possibility, particularly used in NOTAM messages, is that coordinates may be followed, when available, by geographical indications between brackets, such as in the example below.

"E) AIR DISPLAY WILL TAKE PLACE WI LATERAL LIMITS: 443838N 0200818E (**NDB OBR**) - 444508N 0201455E (**VILLAGE JAKOVO**) - 443445N 0202447E - 443838N 0200818E (**NDB OBR**).

F) GND G) 3000FT AMSL)"

Formatted: Font: Bold, Complex Script Font: Bold

Formatted: Font: Bold, Complex Script Font: Bold

Formatted: Font: Bold, Complex Script Font: Bold

Formatted: Left

7.2 GML encoding

The encoding of point references and annotations uses the gml:PointProperty element. Note that the pointProperty allows either referring to another gml:Point (by xlink:href) or providing a gml:Point child element.

According to the GML standard, para 10.2.2.2: "*A property that has a point as its value domain may either be an appropriate geometry element encapsulated in an element of this type or an XLink reference to a remote geometry element (where remote includes geometry elements located elsewhere in the same document).* ***Either the reference or the contained element shall be given, but neither both nor none***."

7.2.1 Point annotations

In this case, a gml:pointProperty is used, including an aixm:Point with an annotation (aixm:Note). This encoding has the advantage that the geometry is self-contained (the position of the referenced object is copied as a gml:pos element).

This method should be used whenever the reference information provided is "for human consumption", such as in the case of the NOTAM examples "VILLAGE JAKOVO". Even in the case when an arc center is located on a DME navaid and the distance information provided by the DME can be used to keep the aircraft inside or outside the arc, the provision of a Point annotation could be sufficient for the end user.

An example is provided below:

```
...
                <gml:exterior>
                  <gml:Ring>
                    <gml:curveMember>
                      <gml:Curve gml:id="C001">
                        <gml:segments>
                          <gml:GeodesicString>
                            <gml:posList>52.18556 5.20833 52.20611 5.2875 52.18917 5.29889 52.16917
5.29889</gml:posList>
                          </gml:GeodesicString>
                          <!-- The next segment contains a point annotation encoded as a Note-->
                          <gml:GeodesicString>
                            <gml:pos>52.16917 5.29889</gml:pos>
                            <gml:pointProperty>
                              <aixm:Point gml:id="P001">
                                <gml:pos>52.16917 5.21972</gml:pos>
                                <aixm:annotation>
                                  <aixm:Note gml:id="N001">
                                    <aixm:translatedNote>
                                      <aixm:LinguisticNote gml:id="N002">
                                        <aixm:note lang="ENG">VILLAGE JAKOVO</aixm:note>
                                      </aixm:LinguisticNote>
                                    </aixm:translatedNote>
                                  </aixm:Note>
                                </aixm:annotation>
                              </aixm:Point>
```

Deleted: ¶

Formatted: Bullets and Numbering

Formatted: Heading 3;h3;sub-clause 3;H3;hd3

Formatted: Normal

Formatted: XML code, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers

Formatted: Font: Bold, Complex Script Font: Bold

Formatted: Font: Bold, Complex Script Font: Bold

Formatted: Font: Bold, Complex Script Font: Bold

Formatted: Font: Bold, Complex Script Font: Bold

Formatted: Font: Bold, Complex Script Font: Bold

```
                    </gml:pointProperty>
                  </gml:GeodesicString>
                  <!-- This is the final straight segment encoded as a Geodesic, which closes the surface-->
                  <gml:GeodesicString>
                    <gml:posList>52.16917 5.21972 52.18556 5.20833</gml:posList>
                  </gml:GeodesicString>
                </gml:segments>
              </gml:Curve>
            </gml:curveMember>
          </gml:Ring>
        </gml:exterior>
...
```

7.2.2 Point references

When necessary to present as a true reference (xlink:href) the information that the current position depends on the location of another aeronautical feature, then a gml:PointProperty can again be used, but using an xlink:href instead of a child gml:Point/gml:pos element. There are two ways to encode that: either with a local reference (reference to a gml:id value) or using a remote reference.

7.2.2.1 Local reference to gml:pointProperty

In the example below, the position of the Navaid is used as centre for the circle that defines the horizontal geometry of the Airspace.

```
<aixm:Navaid gml:id="NID2168342">
  …
    <aixm:type>VOR_DME</aixm:type>
    <aixm:name>DONLON</aixm:name>
    <aixm:location>
      <aixm:ElevatedPoint gml:id="P0001" srsName="urn:ogc:def:crs:EPSG::4326">
        <gml:pos>52.2889 -32.0350</gml:pos>
        <aixm:elevation uom="FT">365</aixm:elevation>
      </aixm:ElevatedPoint>
    </aixm:location>
  …
</aixm:Navaid>
…
<aixm:Airspace gml:id="VID2168342">
    …
      <aixm:theAirspaceVolume>
        <aixm:AirspaceVolume gml:id="V001">
          <aixm:horizontalProjection>
            <aixm:Surface gml:id="S001" srsName="urn:ogc:def:crs:EPSG::4326">
              <gml:polygonPatches>
                <gml:PolygonPatch>
                  <gml:exterior>
                    <gml:Ring>
                      <gml:curveMember>
                        <gml:Curve gml:id="CUR001">
                          <gml:segments>
                            <gml:CircleByCenterPoint numArc="1">
                              <gml:pointProperty xlink:href="#P0001" xlink:title="VOR/DME DONLON"/>
                              <gml:radius uom="[nmi_i]">12</gml:radius>
```

Formatted: Heading 3;h3;sub-clause 3;H3;hd3

Formatted: Bullets and Numbering

Deleted: Obviously, the position of the referenced element can be simply copied using a gml:pos element. However, that solution would not

Deleted: rve

Deleted: which is an important operational aspect.

Deleted: For example, if the arc center is located on a DME navaid, then the distance information provided by the DME can be used to keep the aircraft inside or outside the arc, as appropriate.

Formatted: Highlight

Comment: Not sure if this is really possible.

Formatted: Highlight

Deleted: <#>¶

Formatted: Heading 4;h4;sub-clause 4;H4;hd4

Deleted: GML offers a solution to this problem, using the gml:pointProperty element.

Deleted: ¶

Formatted: XML code, Adjust space between Latin and Asian text, Adjust space between Asian text and numbers, Border: Top: (No border), Bottom: (No border), Left: (No border), Right: (No border)

```
                        </gml:CircleByCenterPoint>
                      </gml:segments>
                    </gml:Curve>
                  </gml:curveMember>
                </gml:Ring>
              </gml:exterior>
            </gml:PolygonPatch>
          </gml:polygonPatches>
        </aixm:Surface>
      </aixm:horizontalProjection>
    </aixm:AirspaceVolume>
  ...
</aixm:Airspace>
```

Note also that the xlink:title attribute is used to provide a human readable identification of the Navaid that is referred, which can be used in printed documents.

This solution does not imply the persistence of the gml:id value. It is still just a temporary identifier, which enables linking the gml:PointProperty with the gml:Point inside the file.

This direct link between gml:PointProperty and gml:Point is a deviation from the general AIXM principle of having xlink:href associations towards the feature level only. However, this direct association with the gml:Point property of the aixm:Navaid is the only solution identified for encoding geometry dependencies at the GML level. However, inside a source database, the association can still be towards the Navaid itself. Only for data export/import purpose the reference would be towards the gml:Point directly.

Another aspect to be kept in mind is that this kind of reference is not necessary for all the consumers of the data. Applications should offer the possibility for the recipient of the data to specify if they want such references to be preserved or be replaced with copies of gml:Point elements, eventually with annotations (as explained in 7.2.1).

In particular for Web Feature Server (WFS) applications, the gml:pointProperty cannot be used with a local gml:id, a gml:pos is necessary?

7.2.2.2 Abstract reference to remote feature property

Is this really possible? This would be a deviation from the AIXM xlink:href document, which does not foresee an abstract reference towards a pointProperty.

If the gml:pointProperty had an xlink:href towards the aixm:Navaid, then a native GML tool would identify it as an error, because the aixm:Navaid is not in the substitution group of gml:Point.

Formatted: Highlight

Formatted: Heading 4;h4;sub-clause 4;H4;hd4

Formatted: Highlight

Formatted: Highlight

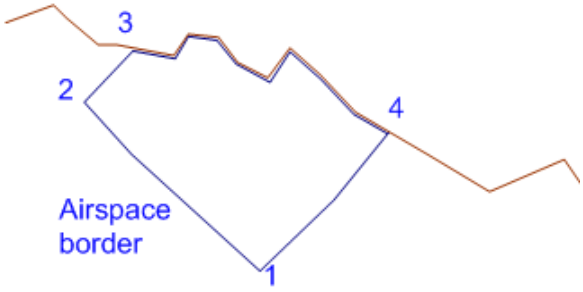
Formatted: Bullets and Numbering

8 Geographical border references

Formatted: Heading 2;h2;sub-clause 2;H2

8.1 Background

Many Airspace boundaries are based on national Country borders or on other geographical features, such as shorelines, rivers, etc.



In such situations, the official definition of the airspace, as provided by Aeronautical Information Publication (AIP) documents, does not contain the geometry of the border. It is left to the end users to derive the actual geometry of the airspace by using a source of geographical border data.

Formatted: Heading 2;h2;sub-clause 2;H2

8.2 GML encoding

A similar solution as for point references is available in GML for re-using a curve into the definition of another curveMember. This is presented in the example below.

Is it possible to use an aixm:Curve with a Note, similar to the Point Reference situation?

Formatted: Highlight

Deleted: ¶

ToDo: replace this with a realistic example; include the actual geo border geometry in the example.

Formatted: Highlight

```
<aixm:Surface gml:id="S001" srsName="urn:ogc:def:crs:EPSG::4326">
  <gml:polygonPatches>
    <gml:PolygonPatch>
      <gml:exterior>
        <gml:Ring>
          <gml:curveMember>
            <gml:Curve gml:id="CUR001">
              <gml:segments>
                <gml:GeodesicString interpolation="geodesic">
                  <gml:posList>52.0 -32.0 53.00 -33.0 54.0 -31.0</gml:posList>
                </gml:GeodesicString>
              </gml:segments>
            </gml:Curve>
          </gml:curveMember>
          <!-- Note that the reference below should resolve to exactly the portion of the France-
Germany border that is actually used in the Airspace Defintion!-->
```

Comment: Warwick: Is it assumed therefore that the first point of the DONLON-ATLANTIS border coincides with the last point of Curve 'CUR001' (ditto for the last point of DONLON-ATLANTIS border and the first point of 'C004')? If not how are they joined?

Also, to traverse the DONLON-ATLANTIS border in reverse (say for a neighbouring FIR) presumably the border reference can be wrapped in an OrientableCurve?

```
                        <gml:curveMember xlink:href="urn:uuid:71537818731853781257235253 7"
xlink:title="follow the DONLON-ATLANTIS border"/>
                        <gml:curveMember>
                          <gml:Curve gml:id="C004">
                            <gml:segments>
                              <gml:GeodesicString interpolation="geodesic">
                                <gml:posList>52.5 -32.5 52.0 -32.0</gml:posList>
                              </gml:GeodesicString>
                            </gml:segments>
                          </gml:Curve>
                        </gml:curveMember>
                      </gml:Ring>
                    </gml:exterior>
                  </gml:PolygonPatch>
                </gml:polygonPatches>
              </aixm:Surface>
```

Geo-border to be a compositeCurve, the exact portion of the curve that is used in the Airspace should be directly identifiable with a gml:id.

See similar usage aspects for point references.

Possibilities:

- reference to geo-border in its entirety (implement as an AIXM association)

In particular for Web Feature Server (WFS) applications, the gml:pointProperty cannot be used with a local gml:id, a gml:pos is necessary.

Formatted: Highlight

Formatted: Normal, Tabs: Not at 0,76 cm

9 GML Profile

To be detailed. For the moment, it is just a list of candidate gml elements.

9.1 GML 3.2 Point

- gml:Point with gml:pos child elemente

9.2 GML 3.2 Curve Types

The GML curve types listed below (selected from the substitution group of AbstractCurveSegment) shall be supported as a minimum by applications in the Aviation domain:

- ArcByCenterPoint
- CircleByCenterPoint
- ArcString
- Arc
- Circle
- CubicSpline
- GeodesicString
- Geodesic
- OffsetCurve
- LineStringSegment

9.3 GML 3.2 Surface Types

The GML surface types listed below (selected from the substitution group of AbstractSurfacePatch) shall be supported as a minimum by applications in the Aviation domain:

GML “surface” type elements are used in AIXM within the following tree structure:

aixm:Surface or aixm:ElevatedSurface (they are in the substitution group of gml:Surface)

…

- gml:patches

Formatted: Bullets and Numbering

Formatted: Highlight

Formatted: Highlight

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

Formatted: Bullets and Numbering

PolygonPatch …GML document, just use gml:patches – the other two are for backwards compatibility reasons!

Formatted: Bullets and Numbering

10 Consistency between GML order of points and the non-GML information

To Do

Formatted: Highlight

Example: RouteSegment also has a gml:curve. How is the curve represented - how do you know directionality. The startPoint and endPoint must be consistent with the direction of the curve.

Annex A - ArcByCenterPoint Interpretation Summary

Introduction

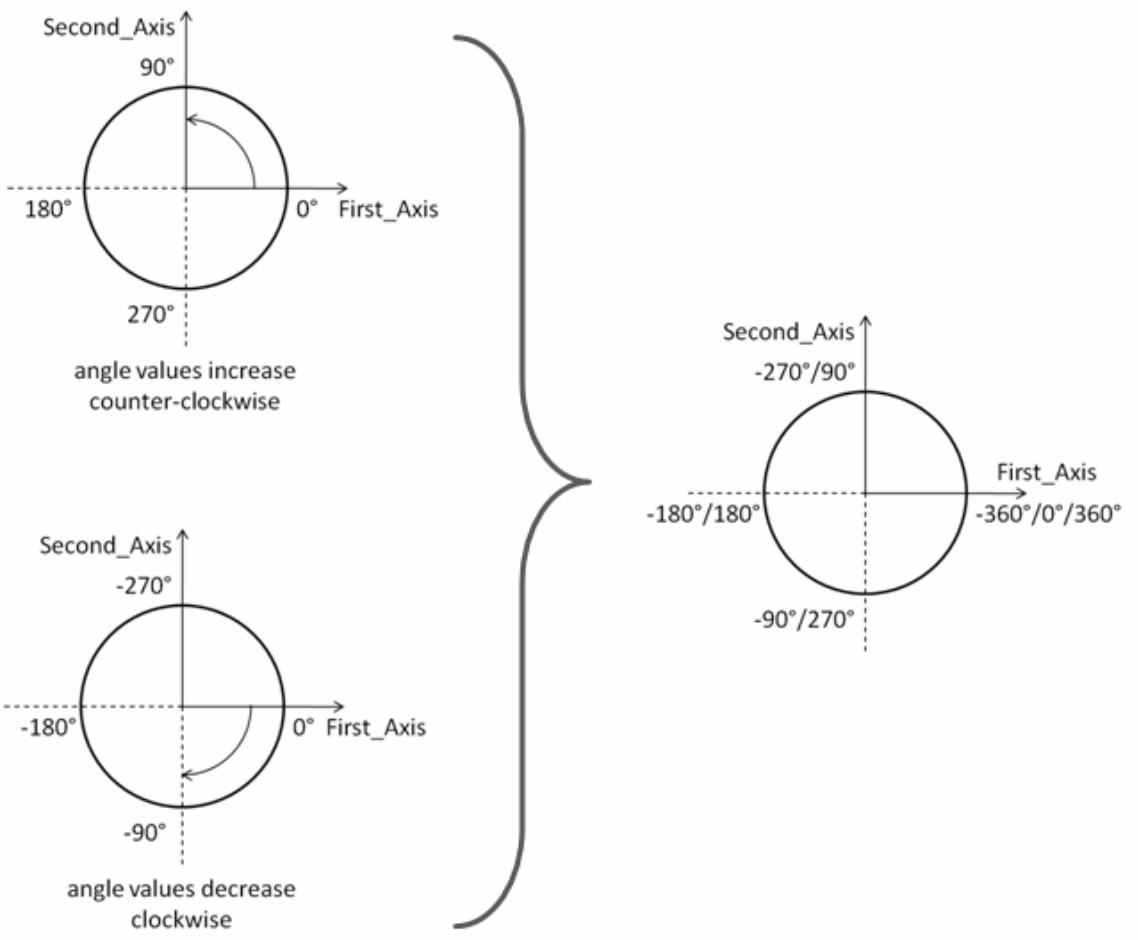
This page contains the summary of the discussions on the correct interpretation of ArcByCenterPoint.

Angle Measuring Convention in GML

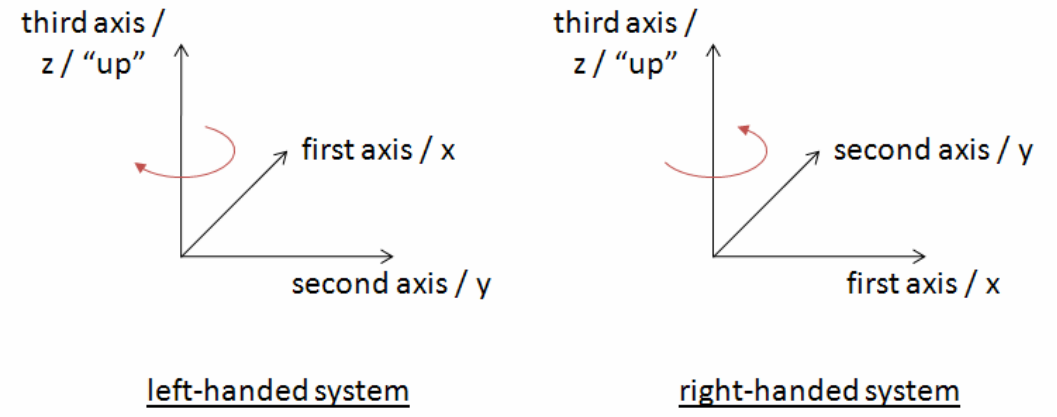
GML implements the semantics of ISO 19107. As such, the semantics for an angle encoded in GML is given by ISO 19107 clause 6.3.12.2: *"In this variant of Bearing usually used for 2D coordinate systems, the first angle (azimuth) is measured from the first coordinate axis (usually north) in a counterclockwise fashion parallel to the reference surface tangent plane."*

General Direction of Increasing and Decreasing Angle Values

ISO 19107 thus defines in which direction angle values increase and in which direction they decrease - see the following diagram:



Depending upon the "up" and the orientation of the first two axes, we have a left-handed or right handed coordinate system (see wikipedia and the following figure).

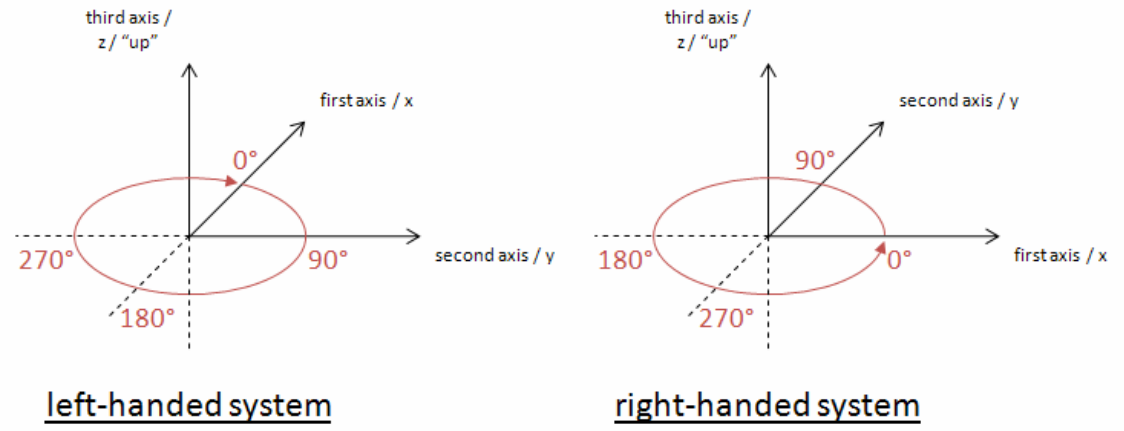


(Note: you can try this yourself with the following convention: thumb=x, index=y, middle=z)

Left-handed systems have a clockwise rotation from first to second axis (positive axis to positive axis). Right-handed systems have a counterclockwise rotation from first to second axis (positive axis to positive axis).

Note: the direction of "clockwise" and "counterclockwise" therefore depends on the "up", not the order of the first two axes in the coordinate system. Swapping the order of the first and second axis does not affect the direction of clockwise, just the direction of the rotation between the two axes.

In a left-handed system, angle values increase in clockwise direction. In a right-handed system, angle values increase in counterclockwise direction.

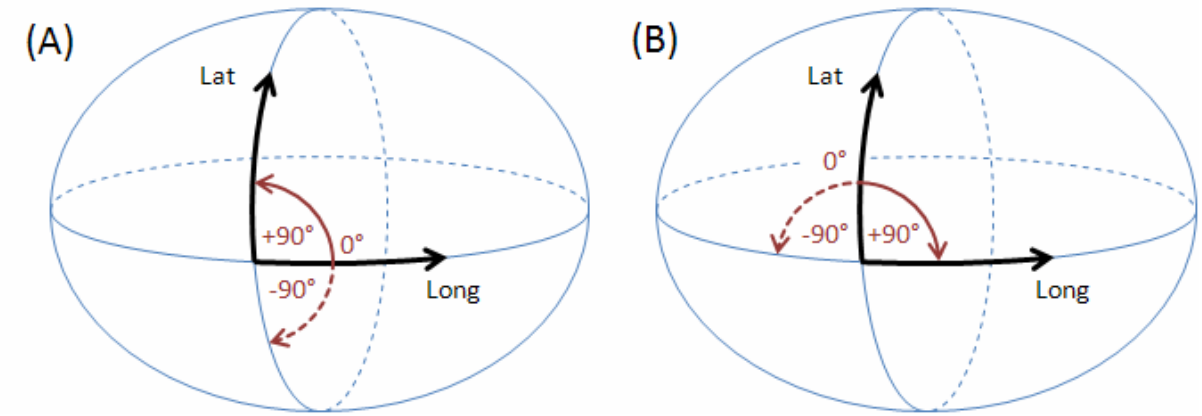


Note: The planes in geographic information are representations of the ground, and as such have a "natural" up direction (usually represented by a normal vector). In a 3D system, "up" is given by the definition of the third axis in the chosen CRS. In a 2D system "up" is usually implied.

Angle Measurement in Different Coordinate Systems

Apparently the way that angle values are expressed heavily depends upon the axis order defined by the coordinate system that is used.

The following diagram shows how angles are measured in WGS 84 2D with different coordinate systems:



(A) GeodeticCRS: urn:ogc:def:crs:OGC:1.3:CRS84

- Datum: WGS84
- Ellipsoidal 2D CS.
    - Axes: (1st) longitude, (2nd) latitude.
    - Orientations: east, north. UoM: degree

(B) GeodeticCRS: urn:ogc:def:crs:EPSG::4326

- Datum: WGS84
- Ellipsoidal 2D CS.
    - Axes: (1st) latitude, (2nd) longitude.
    - Orientations: north, east. UoM: degree

The order of the axes determines where 0° is located.

Definition: The 0° angle is located on the positive part of the coordinate system's first axis.

Note: As explained in the previous section, the order of the first two axes and their location "on the ground" in combination with the "up" direction defines in which directions the angle values increase/decrease.

Direction of an Arc

The direction of a line is always from its start to its end. The same is true for arcs defined by a start and end angle.

Definition: if the start angle is smaller than the end angle then the arc direction is the direction in which the angle values increase.

Note: depending upon the coordinate system that applies to a given ArcByCenterPoint, this results in a clockwise (left-handed system) or counter-clockwise (right-handed system) directed arc.

Example A: CRS urn:ogc:def:crs:OGC:1.3:CRS84 has long/lat axis order. Because the orientation of the axes on the earth surface is the same as that of the first_axis (=x) / second_axis (=y) coordinate system assumed by ISO 19107, angle values increase in counter-clockwise direction in both cases. Thus, if the start angle of an ArcByCenterPoint in this CRS is smaller than its end angle, then the direction of the arc is counter-clockwise; otherwise it is clockwise.

Example B: CRS urn:ogc:def:crs:EPSG::4326 has lat/long axis order. Even though the axes have the same orientation on the earth surface as in example A, the axis order of the coordinate system is different! In the EPSG:4326 CRS, the "counter-clockwise" convention from ISO 19107 actually corresponds to a clockwise rotation, because the first_axis (=x) / second_axis (=y) coordinate system assumed by ISO 19107 is mirrored (reflected) through the x=y diagonal when transposing the coordinate system used by EPSG:4326 on the surface of the Earth. Because of this, if the start angle of an ArcByCenterPoint in EPSG:4326 is smaller than its end angle, then the direction of the arc on the earth surface is clockwise; otherwise it is counter-clockwise.

In other, more mathematical words: angle values increase in counter-clockwise direction if the coordinate system of the CRS used for an ArcByCenterPoint can be transformed into the coordinate system implied by ISO 19107 through a simple rotation. Whenever a reflection across the x=y line is needed (+ any rotation) to achieve the axis orientation of the given CRS's coordinate system then the angle values of an arc increase in clockwise direction.

Examples

The following diagrams exemplify this definition. They show the arc on a line representing a flattened circle first, followed by its representation in the general case.

Notes:

- as angle values have an unlimited range, the flattened circle is a theoretically unlimited line.
- each angle (from [0°,360°[) on the circle has an unlimited number of representations: *angle + X * 360°, X being an integer*
- to define arcs on a circle, it is important that the difference between end and start angle is smaller than 360°: |*end-angle - start-angle*| < *360°* (else the result would be an arc that is greater than or equal to 360° and looks like a circle)
- computing the length of an arc can be performed as follows: |*startAngle – endAngle*|*/180° * radius * pi* - the definition conforms to this formula

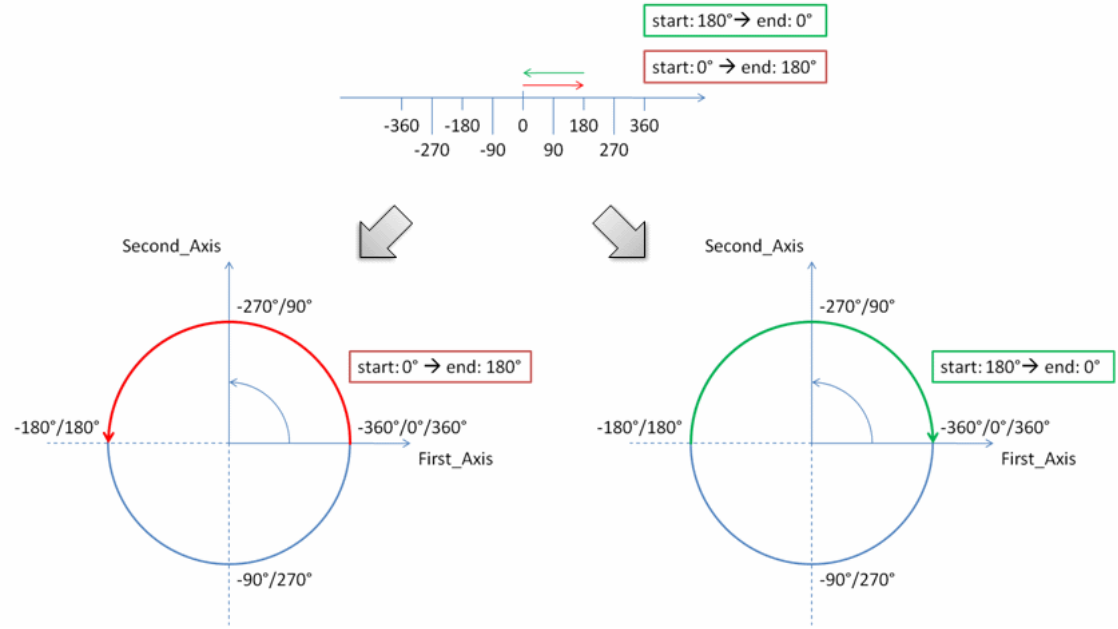
The according ArcByCenterPoint would look like the following (with start and end angles as used in the examples):

```
<gml:ArcByCenterPoint numArc="1"
xsi:schemaLocation="http://www.opengis.net/gml/3.2
http://schemas.opengis.net/gml/3.2.1/gml.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gml="http://www.opengis.net/gml/3.2">
  <gml:pointProperty>
    <gml:Point gml:id="ID1">
      <gml:pos>0 0</gml:pos>
    </gml:Point>
  </gml:pointProperty>
  <gml:radius uom="m">1</gml:radius>
  <gml:startAngle uom="deg">actual_start_value</gml:startAngle>
  <gml:endAngle uom="deg">actual_end_value</gml:endAngle>
</gml:ArcByCenterPoint>
```

Note that the spatial reference system - and thus the coordinate system - to be used cannot be directly defined in the ArcByCenterPoint. It is the same as that of the Curve that the ArcByCenterPoint is a segment of.

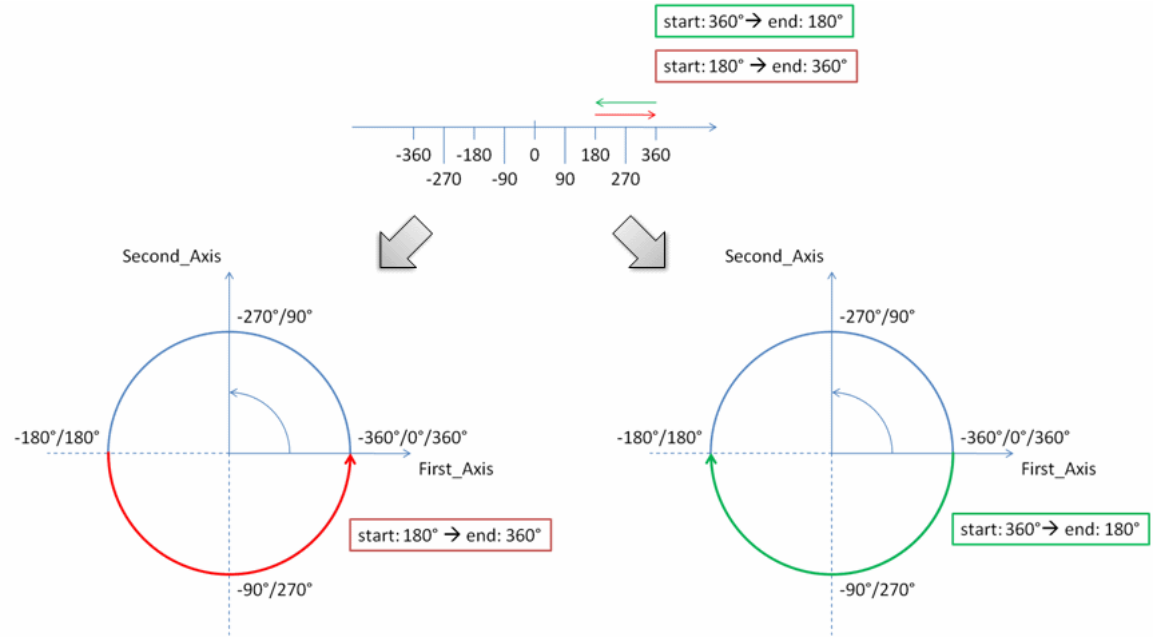
---

example - one

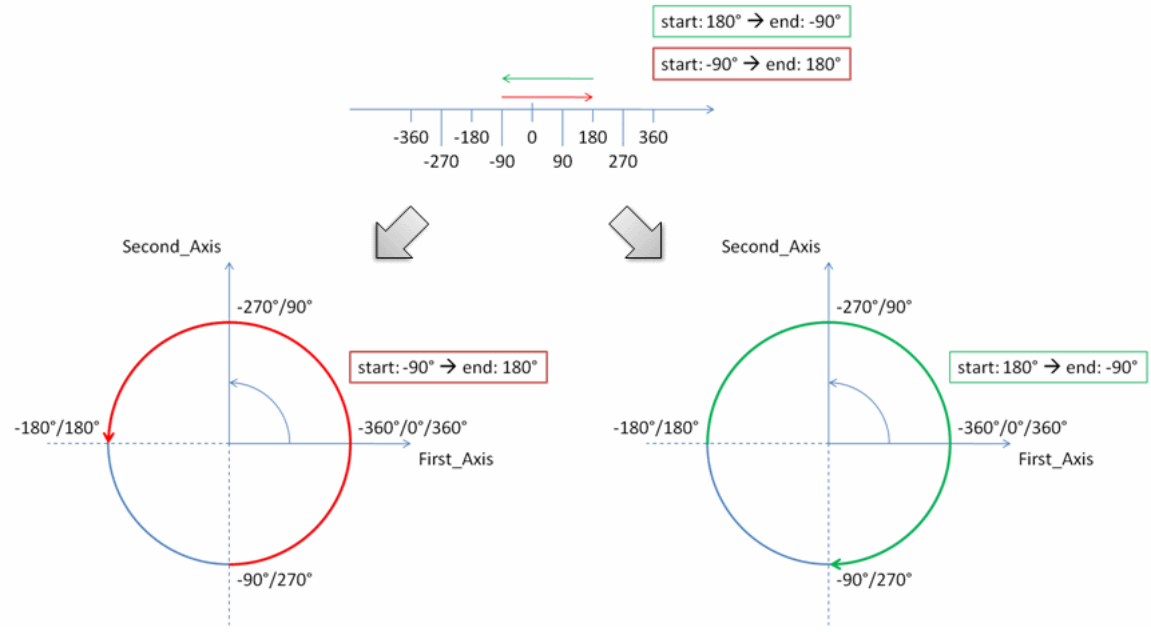


---

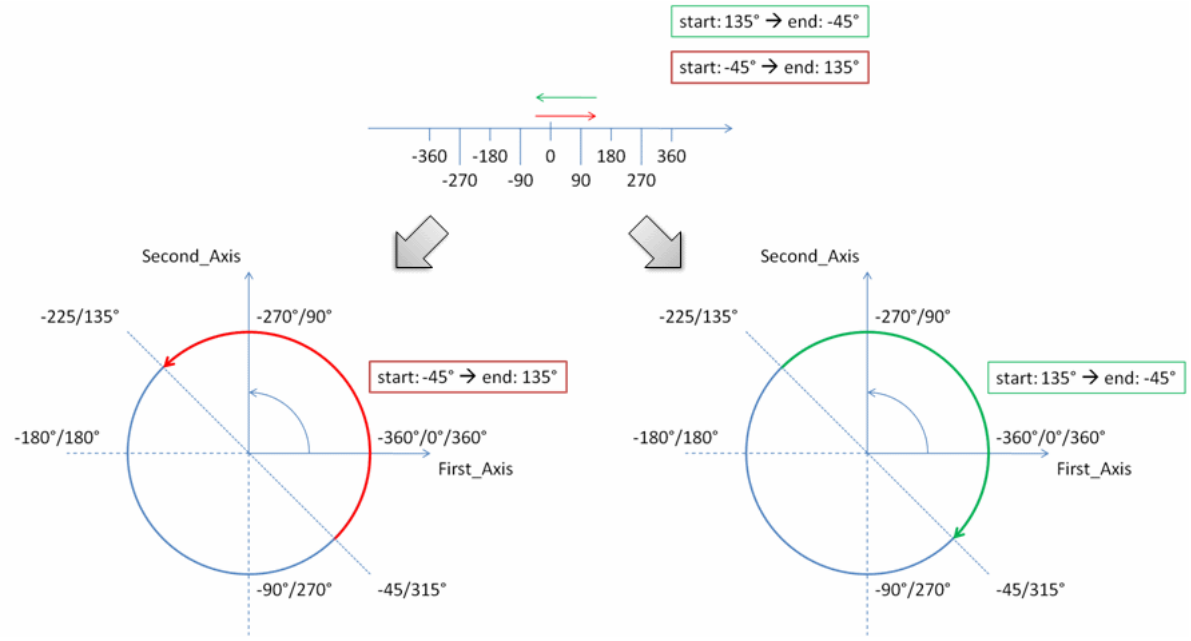
example - two



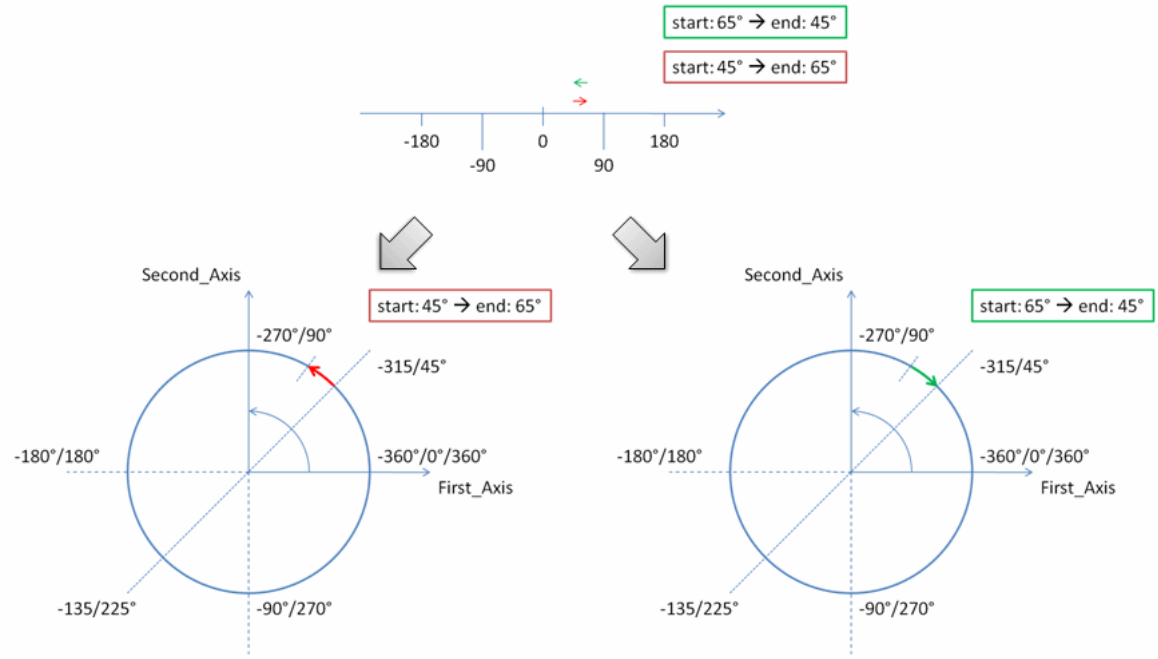
example - three



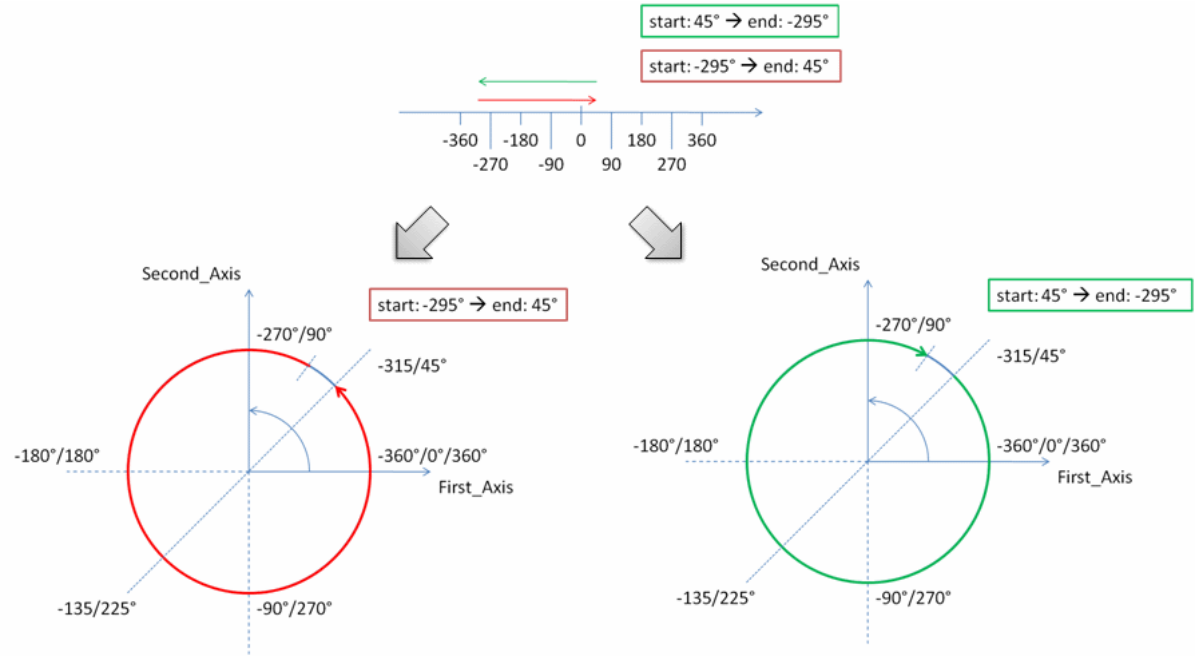
example - four



example - five



example - six



Annex B – GML change request – support arbitrary rhumblines

The following change request has been raised within OGC in order to enable specifying additional interpolations, such as rhumbline.

Title: ⌘	GML: Make CurveInterpolationType and SurfaceInterpolationType expandable codelists.
Source: ⌘	
Work item code: ⌘	***Date:*** ⌘ 2008-11-11
Category: ⌘	**B**

Use one of the following categories:
F *(Critical correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in the TC Policies and Procedures.

Reason for change: ⌘	(1.) Other curve interpolation types needed in applications (e.g. rhumb lines in Aeronautical Information (AIXM), often used in (aerial and naval) navigation). (2.) ISO 19107 GM_CurveInterpolation and GM_SurfaceInterpolation are <<CodeList>>s that are expandable per definition.
Summary of change: ⌘	Change the definition of CurveInterpolationType (10.4.7.3) and SurfaceInterpolationType (10.5.12.3) such that both become extendable codelists in either of the two ways specified in E.2.4.9 to allow the use of other values than the predefined ones. This also brings the implementation in line with ISO 19107, where both types are <<CodeList>>s, which are supposed to be implemented as expandable types.

Page 9: [1] Deleted **POROSNICU Eduard (HBRUPW85X - porosnie)** **28/04/2011 1:54 PM**

The srsDimension attribute of the parent aixm:Surface element is very important from this point of view because it indicates explicitly the number of values (i.e. equal to the number of coordinate axes) that define the position. For a 2D CRS, such as EPSG:4326, this shall be 2 – latitude and longitude only. In a 3D CRS, a posList could have 3 values for each group: latitude, longitude and elevation.

If a precise interpolation needs to be specified, GML provides dedicated elements, such as in the example below, where a geodesic interpolation is used for the curve.

….

```
<gml:Curve>
 <gml:segments>
  <gml:Geodesic interpolation="geodesic">
   <gml:posList>11.282222 56.007778 11.173889 56.111944</gml:posList>
  </gml:Geodesic>
 </gml:segments>
</gml:Curve>
```

….

As long as the data originator does not specify a precise type of interpolation, it is considered that the simple linear interpolation is the most appropriate encoding, therefore the gml:LinearRing will be the typical element used for encoding airspace boundaries.

Page 18: [2] Deleted **POROSNICU Eduard (HBRUPW85X - porosnie)** **14/06/2011 1:30 PM**

However, if the difference between the distances P3-P2 and P3-P4 is higher than 1%, then the application shall raise an error message and abort the calculation. This rule was also specified in previous AIXM versions (4.5) and it did prevent the provision of geometrically inconsistent arc data. Then, using the position of the P3 and P2/P4, the start/end angles shall be calculated, measured clockwise from True North.