All Fields marked with * are mandatory.

Change Request #:	**235**
Assigned OGC Document #:	**12-106r1**
Name:	*Paul Scarponcini
Organization:	*Bentley Systems, Inc.
Email:	*paul.scarponcini@bentley.com
Document Name/Version:	*Geography Markup Language (GML) Encoding Standard / 3.2.1
OGC Project Document:	*07-036
If this is a revision of a previous submission and you have a Change Request Number, then check here: ☐ Enter the CR number here: Enter the Revsion Number that you are revising here:	
Title:	* GML Harmonization with SF, SQL/MM and ISO TC211
Source:	*Paul Scarponcini
Work item code:	
Category:	* C (Functional modification of feature)
Reason for change:	* It has been a long time goal to keep SQL/MM, GML, OGC Simple Features and ISO TC211 harmonized. As SQL/MM and TC211 19107 are currently undergoing revision, it is the appropriate time to attempt to fix all known differences.
Summary of change:	* 1) GML BSpline is in error in associating weights with knots instead of with control points so this must be corrected. 2) GML needs to be corrected to state that the start and end angles of a Circle should not be identical and that they should differ by 360 degrees so that the direction of the curve is determinate

	3) GML EllipticalCurve is proposed as a normative replacement for the Ellipse informative type. 4) GML supports clothoids for roadway design but a more complete list of spiral curves for rail design is needed. 5) GML Circle does not support measure (m) coordinate value at the end of the curve. 6) GML Bspline and clothoid do not support measure (m) coordinate values at the start or end of the curve.
Consequences if not approved:	Without these changes, GML will not be in harmony with the new revisions of SQL/MM, OGC Simple Features and ISO TC211 19107: 1) and 2) above are errors in GML which need to be corrected; 3) GML only provides an example for closed ellipses which is insufficient for partial elliptical curves, only informative, and inconsistent with the proposed changes to SQL/MM and OGC SF / 19107 4) GML does not support the spiral curve type proposed for SQL/MM and OGC SF / 19107 5) and 6) are deficiencies being corrected in the proposed SQL/MM and OGC SF revisions
Clauses affected:	* 10.4.7.7 CircleType, Circle 10.4.7.13 BSplineType, BSpline 10.4.7.14 KnotType, KnotPropertyType 10.4.7.16 BezierType, Bezier 10.4.7.19 ClothoidType, Clothoid E.2.4.7 UML classes (object types) Ellipse example
Additional Documents affected:	ISO 19107:2003, Geographic information — Spatial schema OpenGIS® Implementation Specification for Geographic information - Simple feature access - Part 1: Common Architecture, OGC-06-103r3
Supporting Documentation:	GML Harmonization Items, ISO/IEC JTC 1/SC 32/WG 4:KRK-024. See attached.
Comments:	
Status:	Assigned
Assigned To:	GML 3.3 SWG
Disposition:	Referred and Posted

Detailed Change Proposal

It has been a long time goal to keep SQL/MM, GML, OGC Simple Features and ISO TC211 harmonized. As SQL/MM, OGC Simple Features and TC211 19107 are currently undergoing revision, it is the appropriate time to attempt to fix all known differences in these as well as the GML standard.

The six opportunities identified for change in GML 3.2.1 are:

1) **BSpline Weights:** GML BSpline is in error in associating weights with knots instead of with control points so this must be corrected.
2) **Circle Start and End Angles:** GML needs to be corrected to state that the start and end angles of a Circle should not be identical and that they should differ by 360 degrees so that the direction of the curve is determinate
3) **Elliptical Curve:** GML EllipticalCurve is proposed as a normative replacement for the Ellipse informative type.
4) **Spiral Curves:** GML supports clothoids for roadway design but a more complete list of spiral curves for rail design is needed.
5) **Circle End Measure:** GML Circle does not support measure (m) coordinate value at the end of the curve.
6) **BSpline, Bezier and Clothoid Measures:** GML Bspline, Bezier and Clothoid do not support measure (m) coordinate values at the start or end of the curve.

For each of the six problems identified, a problem statement, proposed solution and proposed revision text are supplied. For the latter, the existing GML 3.2.1 text is included. Any of this text which is proposed to be deleted in shown as crossed out ~~blue~~ text. Any new text is shown as bold, underlined red **<u>text</u>**. Proposed revision text is independent for each of the six problems and may therefore apply to the same GML 3.2.1 clause. If overlapping solutions are accepted, the proposed revisions will need to be merged.

The following GML 3.2.1 clauses are affected (the red ones were not included in the original change proposal):

<u>10.4.7.3 CurveInterpolationType</u>
10.4.7.7 CircleType, Circle
<u>10.4.7.10 ArcByCenterPointType, ArcByCenterPoint</u>
<u>10.4.7.11 CircleByCenterPointType, CircleByCenterPoint</u>
10.4.7.13 BSplineType, BSpline
10.4.7.14 KnotType, KnotPropertyType
10.4.7.16 BezierType, Bezier
10.4.7.19 ClothoidType, Clothoid
E.2.4.7 UML classes (object types) Ellipse example

1. BSpline Weights

1.1 Problem Statement:

GML BSpline is in error in that it associates weights with knots instead of with control points.

1.2 Proposed Solution:

The first choice compositor for the GML BSplineType allows for the specification of un-weighted control points. This should be changed to use weighted control points instead, that is, points of type WeightedControlPointType (Approach A).

Create a new WeightedControlPointType for WeightedControlPoints. A WeightedControlPoint is comprised of a control point and a weight value, where the coordinates of the control point already have the weight applied. For (wx,wy,wz) values stored as the control point coordinates, the wx, for example, is usually the weight times the Cartesian x coordinate value (not the original un-weighted Cartesian x coordinate value).

An alternative Approach B would be to continue to use pos as currently defined, but carrying the weight as the final coordinate for each control point. This would obviate the need for a new WeightedControlPointType. However, it first must be agreed that GML, OGC SF, 19107 and SQL/MM would not preclude the BSpline geometry from having an SRS that is different from the SRS of the control points.

Remove weight from KnotType.

Propose similar changes to ISO 19107.

Note: Does this affect GML Bezier as well?

1.3 Proposed Revision Text:

10.4.7.13 BSplineType, BSpline [Approach A]

```
    <complexType name="BSplineType">
        <complexContent>
            <extension base="gml:AbstractCurveSegmentType">
                <sequence>
~~                    <choice>~~
~~                        <choice minOccurs="0" maxOccurs="unbounded">~~
~~                            <element ref="gml:pos" />~~
~~                            <element ref="gml:pointProperty" />~~
~~                            <element ref="gml:pointRep" />~~
~~                        </choice>~~
~~                        <element ref="gml:posList" />~~
~~                        <element ref="gml:coordinates" />~~
~~                    </choice>~~
```

```
                <element name="weightedControlPoint" type="gml:WeightedControlPointType"
                  minOccurs="0" maxOccurs="unbounded" />
                <element name="degree" type="nonNegativeInteger" />
                <element name="knot" type="gml:KnotPropertyType" minOccurs="2"
                  maxOccurs="unbounded" />
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType" default="polynomialSpline" />
            <attribute name="isPolynomial" type="boolean" />
            <attribute name="knotType" type="gml:KnotTypesType" />
        </extension>
    </complexContent>
</complexType>

<element name="BSpline" type="gml:BSplineType" substitutionGroup="gml:AbstractCurveSegment" />
```

`gml:BSpline` implements ISO 19107 GM_BSplineCurve (see **Error! Reference source not found.** and ISO 19107:2003, 6.4.30).

A B-Spline is a piecewise parametric polynomial or rational curve described in terms of control points and basis functions as specified in ISO 19107:2003, 6.4.30. Therefore, `interpolation` may be either "polynomialSpline" or "rationalSpline" depending on the interpolation type; default is "polynomialSpline".

`degree` shall be the degree of the polynomial used for interpolation in this spline.

`gml:knot` shall be the sequence of distinct knots used to define the spline basis functions (see ISO 19107:2003, 6.4.26.2).

`gml:weightedControlPoint` shall be the sequence of weighted control points used to define the shape of the curve (see ISO 19107:20nn, tbd).

The attribute `isPolynomial` shall be set to “true” if this is a polynomial spline (see ISO 19107:2003, 6.4.30.5).

The attribute `knotType` shall provide the type of knot distribution used in defining this spline (see ISO 19107:2003, 6.4.30.4).

The content model follows the general pattern for the encoding of curve segments (see 10.4.7).

10.4.7.13 BSplineType, BSpline [Approach B]

```
<complexType name="BSplineType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>
                <choice>
                    <choice minOccurs="0" maxOccurs="unbounded">
                        <element ref="gml:pos" />
                        <element ref="gml:pointProperty" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:posList" />
                    <element ref="gml:coordinates" />
                </choice>
                <element name="degree" type="nonNegativeInteger" />
                <element name="knot" type="gml:KnotPropertyType" minOccurs="2"
                  maxOccurs="unbounded" />
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType" default="polynomialSpline" />
            <attribute name="isPolynomial" type="boolean" />
            <attribute name="knotType" type="gml:KnotTypesType" />
        </extension>
    </complexContent>
</complexType>

<element name="BSpline" type="gml:BSplineType" substitutionGroup="gml:AbstractCurveSegment" />
```

`gml:BSpline` implements ISO 19107 GM_BSplineCurve (see D.2.3.4 and ISO 19107:2003, 6.4.30).
A B-Spline is a piecewise parametric polynomial or rational curve described in terms of **weighted** control points and basis functions as specified in ISO 19107:2003, 6.4.30. Therefore, `interpolation` may be either "polynomialSpline" or "rationalSpline" depending on the interpolation type; default is "polynomialSpline".
Weighted control points are comprised of control point coordinates and a weight value, where the coordinates of the control point already have the weight applied. For (wx,wy,wz) values stored as the control point coordinates, the wx, for example, is usually the weight times the cartesian x coordinate value (not the original unweighted cartesian x coordinate value). The weight value is the final coordinate appended to the control point, resulting in (wx,wy,wz,w) as the weighted control point.
`degree` shall be the degree of the polynomial used for interpolation in this spline.
`gml:knot` shall be the sequence of distinct knots used to define the spline basis functions (see ISO 19107:2003, 6.4.26.2).
The attribute `isPolynomial` shall be set to “true” if this is a polynomial spline (see ISO 19107:2003, 6.4.30.5).
The attribute `knotType` shall provide the type of knot distribution used in defining this spline (see ISO 19107:2003, 6.4.30.4).
The content model follows the general pattern for the encoding of curve segments (see 10.4.7).

10.4.7.14 KnotType, KnotPropertyType [Approach A and B]

```
<complexType name="KnotType">
    <sequence>
        <element name="value" type="double" />
        <element name="multiplicity" type="nonNegativeInteger" />
        ~~<element name="weight" type="double" />~~
    </sequence>
</complexType>
```

`gml:Knot` implements ISO 19107 GM_Knot (see D.2.3.4 and ISO 19107:2003, 6.4.24).
A knot is a breakpoint on a piecewise spline curve.
`gml:value` is the value of the parameter at the knot of the spline (see ISO 19107:2003, 6.4.24.2).
`gml:multiplicity` is the multiplicity of this knot used in the definition of the spline (with the same weight).
~~`gml:weight` is the value of the averaging weight used for this knot of the spline.~~

```
<complexType name="KnotPropertyType">
    <sequence>
        <element name="Knot" type="gml:KnotType" />
```

```
		</sequence>
	</complexType>
```

gml:KnotPropertyType encapsulates a knot to use it in a geometric type.

10.4.7.15 KnotTypesType [Approach A and B]

```
	<simpleType name="KnotTypesType">
		<restriction base="string">
			<enumeration value="uniform" />
			<enumeration value="quasiUniform" />
			<enumeration value="piecewiseBezier" />
		</restriction>
	</simpleType>
```

gml:KnotTypesType implements ISO 19107 GM_KnotType (see D.2.3.4 and ISO 19107:2003, 6.4.25).
This enumeration type specifies values for the knots' type (see ISO 19107:2003, 6.4.25).

10.4.7.15+1 WeightedControlPointType, WeightedControlPoint [Approach A]

```
	<complexType name="WeightedControlPointType">
		<complexContent>
			<extension base="gml:AbstractGMLType">
				<sequence>
					<choice>
						<element ref="gml:pos"/>
						<element ref="gml:pointProperty"/>
					</choice>
					<element name="weight" type="double"/>
				</sequence>
			</extension>
		</complexContent>
	</complexType>

	<element name="WeightedControlPoint" type="gml:WeightedControlPointType"
	  substitutionGroup="gml:AbstractGML" />
```

gml:WeightedControlPoint implements ISO 19107 GM_WeightedControlPoint (see ISO 19107:20nn, tbd).
Weighted control points are comprised of a control point and a weight value, where the coordinates of the control point already have the weight applied. For (wx,wy,wz) values stored as the control point coordinates, the wx, for example, is usually the weight times the cartesian x coordinate value (not the original unweighted cartesian x coordinate value).
gml:weight is the value of the weight used for this control point of the spline.

2. Circle Start and End Angles

2.1 Problem Statement:

When specifying a Circle using the GML 3.2.1 CircleByCenterPointType, the result can be nondeterministic. The Circle is specified by a center point, a radius, and optional start and end angles, the angles being inherited from the parent ArcByCenterPointType. The accompanying text states that the start and end angles are equal, even though these are not made mandatory. The problem is that if they are not specified or are specified as being equal, there is no way to distinguish a clockwise vs. anticlockwise resultant circle.

A similar problem results for the parent ArcByCenterPointType if equal start and end angles are specified.

Another observation is that GML is not clear about when and how an angle represents a bearing. In **16.3.4 angle**, GML merely states that the "`gml:angle` property element is used to record the value of an angle quantity as a single number, with its units. It is declared as follows:

```
<element name="angle" type="gml:AngleType"/>
```

We need to go back to ISO 19107, **6.3.12.2 angle** to see that:

> 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[1] fashion parallel to the reference surface tangent plane.

This needs to be clarified in ArcByCenterPointType, where angle is used as a variant of Bearing.

2.2 Proposed Solution:

For the CircleByCenterPointType, the start and end angles should be made mandatory and should be specified as differing by 360 degrees. This would eliminate the ambiguity and would be consistent with GML 3.3 SimpleCircleByCenterPoint.
Ref: (10-129r1_Geography_Markup_Language_GML_Version_3.3).

For the ArcByCenterPointType, the start and end angles, if present, should not be allowed to be equal.

Clarify what is meant by "angle".

[1] ISO/IEC 13249-3, SQL/MM Part 3: Spatial measures angles clockwise from true North.

2.3 Proposed Revision Text:

10.4.7.10 ArcByCenterPointType, ArcByCenterPoint

```
<complexType name="ArcByCenterPointType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>
                <choice>
                    <choice>
                        <element ref="gml:pos" />
                        <element ref="gml:pointProperty" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:posList" />
                    <element ref="gml:coordinates" />
                </choice>
                <element name="radius" type="gml:LengthType" />
                <element name="startAngle" type="gml:AngleType" minOccurs="0" />
                <element name="endAngle" type="gml:AngleType" minOccurs="0" />
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType"
                fixed="circularArcCenterPointWithRadius" />
            <attribute name="numArc" type="integer" use="required" fixed="1" />
        </extension>
    </complexContent>
    </complexType>

    <element name="ArcByCenterPoint" type="gml:ArcByCenterPointType"
     substitutionGroup="gml:AbstractCurveSegment" />
```

This variant of the arc requires that the points on the arc shall be computed instead of storing the coordinates directly. The single control point is the center point of the arc plus the radius and the **optional** bearing at start and end. **If present, the start and end bearing angles shall not be equal, in order to prescribe a non-ambiguous direction for the arc.** This representation can be used only in 2D.
The element `gml:radius` specifies the radius of the arc.
The element `gml:startAngle` specifies the bearing of the arc at the start.
The element `gml:endAngle` specifies the bearing of the arc at the end.
For `gml:startAngle` and `gml:endAngle`, the bearing is defined by an angle (azimuth) measured from the first coordinate axis (usually north) in a counterclockwise fashion parallel to the reference surface tangent plane, as per ISO 19107:2003, 6.3.12.2.
The `interpolation` is fixed as "circularArcCenterPointWithRadius".
Since this type describes always a single arc, the attribute “numArc” is fixed to "1".
The content model follows the general pattern for the encoding of curve segments (see 10.4.7).

10.4.7.11 CircleByCenterPointType, CircleByCenterPoint

```
<complexType name="CircleByCenterPointType">
    <complexContent>
        <restriction base="gml:ArcByCenterPointType">
            <sequence>
                <choice>
                    <choice>
                        <element ref="gml:pos" />
                        <element ref="gml:pointProperty" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:posList" />
                    <element ref="gml:coordinates" />
                </choice>
                <element name="radius" type="gml:LengthType" />
                <element name="startAngle" type="gml:AngleType" minOccurs="1" />
                <element name="endAngle" type="gml:AngleType" minOccurs="1" />
            </sequence>
        </restriction>
    </complexContent>
</complexType>

    <element name="CircleByCenterPoint" type="gml:CircleByCenterPointType"
    substitutionGroup="gml:ArcByCenterPoint" />
```

A `gml:CircleByCenterPoint` is a~~n~~ `gml:ArcByCenterPoint` ~~with identical start and end angle~~ **closed** to form a full circle. **The start and end bearing angles are mandatory and shall differ by exactly 360 degrees in order to prescribe a non-ambiguous direction for the circle.** Again, this representation can be used only in 2D.

3. Elliptical Curve

3.1 Problem Statement:

There is no normative GML EllipticalCurve type. There is an example of how one might specify a complete ellipse, but, because it is an example, it is assumed to be informative (See E.2.4.7.)

3.2 Proposed Solution:

GML EllipticalCurve is proposed as a normative replacement for the Ellipse informative example in E.2.4.7.

Propose similar addition to ISO 19107.

3.3 Proposed Revision Text:

10.4.7.11+1 EllipticalCurveType, EllipticalCurve

```
    <complexType name="EllipticalCurveType">
        <complexContent>
            <extension base="gml:AbstractCurveSegmentType">
                <sequence>
                    <element name="refLocation">
                        <complexType>
                            <sequence>
                                <element ref="gml:AffinePlacement"/>
                            </sequence>
                        </complexType>
                    </element>
                    <element name="uAxisLength" type="double"/>
                    <element name="vAxisLength" type="double"/>
                    <element name="startAngle" type="gml:AngleType"/>
                    <element name="endAngle" type="gml:AngleType"/>
                </sequence>
                <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="elliptical"/>
                <attribute name="startMeasure" type="double"/>
                <attribute name="endMeasure" type="double"/>
            </extension>
        </complexContent>
    </complexType>

    <element name="EllipticalCurve" type="gml:EllipticalCurveType"
     substitutionGroup="gml:AbstractCurveSegment"/>
```

`gml:EllipticalCurve` implements ISO 19107 GM_EllipticalCurve (see ISO 19107:20nn, tbd). It defines a single elliptical curve segment. `gml:refLocation` is an affine mapping of type `gml:AffinePlacement` that describes the local coordinate system with origin at the elliptical curve center. `gml:uAxisLength` is the elliptical curve axis length along the u axis of its local 2D coordinate system. This is commonly called the “semimajor axis”, but is not required to be the larger axis.

`gml:vAxisLength` is the elliptical curve axis length along the v axis of its local 2D coordinate system.   This is commonly called the “semiminor axis, but it is not required to be the smaller axis.

`gml:startAngle` is the constructive parameter angle of type `gml:AngleType.`

`gml:endAngle` is the constructive parameter angle of type `gml:AngleType.  A gml:EllipticalCurve` having start and end angles that differ by 360 degrees is a complete ellipse.

`gml:interpolation` is fixed as "elliptical".

`gml:startMeasure` is the start measure of the curve.

`gml:endMeasure` is the end measure of the curve.

The content model follows the general pattern for the encoding of curve segments (see 10.4.7).

4. Spiral Curves

4.1 Problem Statement:

GML supports clothoids for roadway design but a more complete list of spiral curves for rail design is needed.

4.2 Proposed Solution:

GML SpiralCurve is proposed as a new curve type to support spiral curves of various types.

Propose similar addition to ISO 19107.

4.3 Proposed Revision Text:

10.4.7.3 CurveInterpolationType

```
<simpleType name="CurveInterpolationType">
    <restriction base="string">
        <enumeration value="linear" />
        <enumeration value="geodesic" />
        <enumeration value="circularArc3Points" />
        <enumeration value="circularArc2PointWithBulge" />
        <enumeration value="circularArcCenterPointWithRadius" />
        <enumeration value="elliptical" />
        <enumeration value="clothoid" />
        <enumeration value="spiral"/>
        <enumeration value="conic" />
        <enumeration value="polynomialSpline" />
        <enumeration value="cubicSpline" />
        <enumeration value="rationalSpline" />
    </restriction>
</simpleType>
```

`gml:CurveInterpolationType` is a list of codes that may be used to identify the interpolation mechanisms specified by an application schema.
This type implements ISO 19107 GM_CurveInterpolation (see D.2.3.4 and ISO 19107:20nn, tbd).

10.4.7.19+1 SpiralCurveType, SpiralCurve

```
<complexType name="SpiralCurveType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>
                <element name="refLocation">
                    <complexType>
                        <sequence>
                            <element ref="gml:AffinePlacement"/>
                        </sequence>
                    </complexType>
                </element>
                <element name="length" type="double"/>
```

```
                    <element name="startCurvature" type="double"/>
                    <element name="endCurvature" type="double"/>
                    <element name="spiralType" type="gml:SpiralTypeType"/>
                </sequence>
                <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="spiral"/>
                <attribute name="startMeasure" type="double"/>
                <attribute name="endMeasure" type="double"/>
            </extension>
        </complexContent>
    </complexType>

    <element name="SpiralCurve" type="gml:SpiralCurveType"
     substitutionGroup="gml:AbstractCurveSegment"></element>
```

`gml:SpiralCurve` implements ISO 19107 GM_SpiralCurve (see ISO 19107:20nn, tbd). It defines a single Spiral curve segment.

`gml:refLocation` is an affine mapping of type `gml:AffinePlacement` that places the spiral into the coordinate reference system of this curve.   The spiral start point is the origin in the placement coordinates, and the initial direction is along the positive x axis.

`gml:length` is the length of the curve.

`gml:startCurvature` is the start curvature value, where curvature is one over the radius of curvature .

`gml:endCurvature` is the end curvature value, where curvature is one over the radius of curvature .

`gml:spiralType` is the type of spiral curve .

`gml:interpolation` is fixed as "spiral".

`gml:startMeasure` is the start measure of the curve.

`gml:endMeasure` is the end measure of the curve.

The content model follows the general pattern for the encoding of curve segments (see 10.4.7).

10.4.7.19+2 SpiralTypeType

```
    <complexType name="SpiralTypeType">
        <annotation>
            <documentation>This codelist type specifies values for the spiral types.</documentation>
            <appinfo>
                <restriction base="string">
                    <enumeration value="clothoid"/>
                    <enumeration value="bloss"/>
                    <enumeration value="biquadratic"/>
                    <enumeration value="sine"/>
                    <enumeration value="cosine"/>
                </restriction>
            </appinfo>
        </annotation>
        <simpleContent>
            <extension base=”gml:CodeType”/>
        </simpleContent>
    </complexType>
```

`gml:SpiralTypeType` implements ISO 19107 GM_SpiralType (see D.2.3.4 and ISO 19107:20nn, tbd).
`gml:SpiralTypeType` is a codelist which includes spiral types and allows for extensions.

5. Circle End Measure

5.1 Problem Statement:

GML Circle does not support a measure (m) coordinate value at the end of the curve because the end of the curve is not an input parameter. The circle is completed by continuing from the last (third) point back to the start point. If the start (first) point has a measure coordinate, it specifies the measure value at the start of the curve. The end of the curve will assume these coordinates, except for the measure coordinate. Therefore the end measure coordinate needs to be specified separately.

5.2 Proposed Solution:

Add measure coordinate as an (optional) attribute to GML Circle.

In similar fashion, add both start and end measures to GML ArcByCenterPoint and GML CircleByCenterPoint.

5.3 Proposed Revision Text:

10.4.7.7 CircleType, Circle

```
<complexType name="CircleType">
    <complexContent>
        <extension base="gml:ArcType" ~~/~~>
            <attribute name="endMeasure" type="double"/>
        </extension>
    </complexContent>
</complexType>

<element name="Circle" type="gml:CircleType" substitutionGroup="gml:Arc" />
```

`gml:Circle` implements ISO 19107 GM_Circle (see D.2.3.4 and ISO 19107:2003, 6.4.16).
A Circle is an arc whose ends coincide to form a simple closed loop. The three control points shall be distinct non-co-linear points for the circle to be unambiguously defined. The arc is simply extended past the third control point until the first control point is encountered. **For measured curves, the last coordinate of each control point is a measure value. The first control point measure value is the start measure for the curve.**
`gml:endMeasure` is the end measure of the curve.

10.4.7.10 ArcByCenterPointType, ArcByCenterPoint

```
<complexType name="ArcByCenterPointType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>
                <choice>
                    <choice>
                        <element ref="gml:pos" />
                        <element ref="gml:pointProperty" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:posList" />
                    <element ref="gml:coordinates" />
                </choice>
                <element name="radius" type="gml:LengthType" />
                <element name="startAngle" type="gml:AngleType" minOccurs="0" />
                <element name="endAngle" type="gml:AngleType" minOccurs="0" />
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType"
                fixed="circularArcCenterPointWithRadius" />
            <attribute name="numArc" type="integer" use="required" fixed="1" />
            <attribute name="startMeasure" type="double"/>
            <attribute name="endMeasure" type="double"/>
        </extension>
    </complexContent>
</complexType>

<element name="ArcByCenterPoint" type="gml:ArcByCenterPointType"
  substitutionGroup="gml:AbstractCurveSegment" />
```

This variant of the arc requires that the points on the arc shall be computed instead of storing the coordinates directly. The single control point is the center point of the arc plus the radius and the bearing at start and end. This representation can be used only in 2D.

The element `gml:radius` specifies the radius of the arc.

The element `gml:startAngle` specifies the bearing of the arc at the start.

The element `gml:endAngle` specifies the bearing of the arc at the end.

The `interpolation` is fixed as "circularArcCenterPointWithRadius".

Since this type describes always a single arc, the attribute “numArc” is fixed to "1".

`gml:startMeasure` is the start measure of the curve.

`gml:endMeasure` is the end measure of the curve.

The content model follows the general pattern for the encoding of curve segments (see 10.4.7).

10.4.7.11 CircleByCenterPointType, CircleByCenterPoint

```
<complexType name="CircleByCenterPointType">
    <complexContent>
        <restriction base="gml:ArcByCenterPointType">
            <sequence>
                <choice>
                    <choice>
                        <element ref="gml:pos" />
                        <element ref="gml:pointProperty" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:posList" />
                    <element ref="gml:coordinates" />
                </choice>
                <element name="radius" type="gml:LengthType" />
            </sequence>
            <attribute name="startMeasure" type="double"/>
            <attribute name="endMeasure" type="double"/>
        </restriction>
    </complexContent>
</complexType>

<element name="CircleByCenterPoint" type="gml:CircleByCenterPointType"
  substitutionGroup="gml:ArcByCenterPoint" />
```

A `gml:CircleByCenterPoint` is an `gml:ArcByCenterPoint` with identical start and end angle to form a full circle. Again, this representation can be used only in 2D.
`gml:startMeasure` is the start measure of the curve.
`gml:endMeasure` is the end measure of the curve.

6. BSpline, Bezier and Clothoid Measures

6.1 Problem Statement:

GML Bspline, Bezier and Clothoid do not support measure (m) coordinate values at the start or end of the curve because no points on the curve are specified.

6.2 Proposed Solution:

Add both start and end measure attributes to GML BSpline, GML Bezier and GML Clothoid.

6.3 Proposed Revision Text:

10.4.7.13 BSplineType, BSpline

```
<complexType name="BSplineType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>
                <choice>
                    <choice minOccurs="0" maxOccurs="unbounded">
                        <element ref="gml:pos" />
                        <element ref="gml:pointProperty" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:posList" />
                    <element ref="gml:coordinates" />
                </choice>
                <element name="degree" type="nonNegativeInteger" />
                <element name="knot" type="gml:KnotPropertyType" minOccurs="2"
                  maxOccurs="unbounded" />
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType" default="polynomialSpline" />
            <attribute name="isPolynomial" type="boolean" />
            <attribute name="knotType" type="gml:KnotTypesType" />
            <attribute name="startMeasure" type="double"/>
            <attribute name="endMeasure" type="double"/>
        </extension>
    </complexContent>
</complexType>

<element name="BSpline" type="gml:BSplineType" substitutionGroup="gml:AbstractCurveSegment" />
```

`gml:BSpline` implements ISO 19107 GM_BSplineCurve (see D.2.3.4 and ISO 19107:2003, 6.4.30).
A B-Spline is a piecewise parametric polynomial or rational curve described in terms of control points and basis functions as specified in ISO 19107:2003, 6.4.30. Therefore, `interpolation` may be either "polynomialSpline" or "rationalSpline" depending on the interpolation type; default is "polynomialSpline".
`degree` shall be the degree of the polynomial used for interpolation in this spline.
`gml:knot` shall be the sequence of distinct knots used to define the spline basis functions (see ISO 19107:2003, 6.4.26.2).

The attribute isPolynomial shall be set to “true” if this is a polynomial spline (see ISO 19107:2003, 6.4.30.5).

The attribute knotType shall provide the type of knot distribution used in defining this spline (see ISO 19107:2003, 6.4.30.4).

gml:startMeasure is the start measure of the curve.

gml:endMeasure is the end measure of the curve.

The content model follows the general pattern for the encoding of curve segments (see 10.4.7).

10.4.7.16 BezierType, Bezier

```
<complexType name="BezierType">
    <complexContent>
        <restriction base="gml:BSplineType">
            <sequence>
                <choice>
                    <choice minOccurs="0" maxOccurs="unbounded">
                        <element ref="gml:pos" />
                        <element ref="gml:pointProperty" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:posList" />
                    <element ref="gml:coordinates" />
                </choice>
                <element name="degree" type="nonNegativeInteger" />
                <element name="knot" type="gml:KnotPropertyType" minOccurs="2" maxOccurs="2" />
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="polynomialSpline" />
            <attribute name="isPolynomial" type="boolean" fixed="true" />
            <attribute name="knotType" type="gml:KnotTypesType" use="prohibited" />
            <attribute name="startMeasure" type="double"/>
            <attribute name="endMeasure" type="double"/>
        </restriction>
    </complexContent>
</complexType>

<element name="Bezier" type="gml:BezierType" substitutionGroup="gml:BSpline" />
```

gml:Bezier implements ISO 19107 GM_Bezier (see D.2.3.4 and ISO 19107:2003, 6.4.31).
Bezier curves are polynomial splines that use Bezier or Bernstein polynomials for interpolation purposes. It is a special case of the B-Spline curve with two knots.
gml:degree shall be the degree of the polynomial used for interpolation in this spline.
gml:knot shall be the sequence of distinct knots used to define the spline basis functions.
interpolation is fixed as "polynomialSpline".
isPolynomial is fixed as “true”.
knotType is not relevant for Bezier curve segments.
gml:startMeasure is the start measure of the curve.
gml:endMeasure is the end measure of the curve.

10.4.7.19 ClothoidType, Clothoid

```
<complexType name="ClothoidType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>
                <element name="refLocation">
                    <complexType>
                        <sequence>
                            <element ref="gml:AffinePlacement"/>
                        </sequence>
                    </complexType>
                </element>
                <element name="scaleFactor" type="decimal"/>
                <element name="startParameter" type="double"/>
                <element name="endParameter" type="double"/>
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="clothoid"/>
            <attribute name="startMeasure" type="double"/>
            <attribute name="endMeasure" type="double"/>
        </extension>
    </complexContent>
</complexType>

<element name="Clothoid" type="gml:ClothoidType" substitutionGroup="gml:AbstractCurveSegment"/>
```

A clothoid, or Cornu's spiral, is plane curve whose curvature is a fixed function of its length.

`gml:Clothoid` implements ISO 19107 GM_Clothoid (see D.2.3.4 and ISO 19107:2003, 6.4.22). `gml:refLocation`, `gml:startParameter`, `gml:endParameter` and `gml:scaleFactor` have the same meaning as specified in ISO 19107:2003, 6.4.22.

`interpolation` is fixed as "clothoid".

`gml:startMeasure` is the start measure of the curve.

`gml:endMeasure` is the end measure of the curve.

The content model follows the general pattern for the encoding of curve segments (see 10.4.7).

End of Paper.