S100TSM xx-xx

Paper for Consideration by S100WG TSM

Geometry Levels, Validation and S-100 Part 7

Submitted by:	IIC Technologies
Executive Summary:	Proposals for handling of realtime data in S-100.
Related Documents:	S-100WG5 papers, OGC Sensor API
Related Projects:	GML Revision to Part 10b

Introduction / Background

The goal of the S-58 revision is to provide a set of validation tests against generic S-100 product specification datasets rather than just validation tests against one particular product specification (S-101). Geometry tests, therefore are designed to validate the geometry of a dataset against a level defined by its product specification under test (e.g. S-101, Level 3a) and the definition of each geometry level is defined in S-100 Part 7-4.3.

Analysis/Discussion

To do this, each level must be well defined in S-100, and the encoding the product specification uses must define how it implements each (or all) of the geometry levels defined in Part 7.4.3. The validation tests then test the dataset's implementation of the geometry level, as encapsulated by the particular encoding used (e.g. ISO8211 or GML). S-100 Part 7 bases its geometry model on ISO19107.

S-100, unlike ISO19107, defines a progression of geometry "levels" of complexity (and dimensionality). In order to properly define these levels, then, a profiling of the complete 19107 model should be made for each level. Against these, the encoding and validation tests can then be defined. S-57 defined similar levels of progressive complexity but differed in that the link with ISO19107 was not explicit and other geometry models remained largely untested. S-100, with its aim of providing a rich environment for multiple product specifications therefore requires a tighter reference to ISO19107 and a solid description of each geometry level expressed in at least one of the standard encodings.

In creating a draft of tests which can apply to multiple S-100 product specifications a number of observations on the existing Part 7 of S-100 have been made. These are summarized in Annex 1 which presents a revision to S-100 Part 7-4.3. The sections in the rest of the document define the individual levels in more detail, with tightyer references to ISO19107 and draft validation tests that follow from the definitions.

The revision to 7-4.3 require clarification in order to map the 19107/S-100 levels to the encodings consistently and these are noted in the Annex. Although most of the levels are incremental there are exceptions and there is no guarantee that a geometry at level N is well-formed at level N+1 (e.g. at Level 1 curve start/end points are not defined by reference, but Level 2a they are, therefore under the current S-100 regime Level 1 geometry is not valid under level 2a).

This paper is a draft for information only. Edits will be proposed to S-100 Part 7 in line with the observations made and will be used as a basis for the revision to S-58.

Conclusions

As most S-101 datasets have been created using conversion tools the full range of geometry allowed within S-100 has not been thoroughly tested, nor the individual levels described in S-100 Part 7. The ISO8211 and GML encodings provide contrasting expressions of geometry and only the ISO8211 encoding can currently support all levels described in Part 7. The inclusion of the two extra levels of abstraction in S-100 geometry (composite curves and surfaces) requires some minor clarifications as does the language in Part 7-4.3 (described here). The questions still to answer are:

1. Should features of primitive "surface" explicitly map to only surface geometry.
2. Can surfaces be composed (and have more than one exterior?) There's nothing to prohibit this in ISO19107 or S-100 currently.

3. The “self-intersection except at a single point” rule is S-100 specific and does not exist in ISO. It would be possible to move this to the S-101 product specification thus prohibiting its use in any other products. It should probably be reviewed to establish use cases and whether it should be preserved at whether at an S-100 level or just for S-101. The argument against it would be that it is not part of the ISO19107 geometry model and relies on manufacturers and implementers creating special cases to deal with such intersections. Given the fundamental relationship between S-100 and the ISO framework and the importance of spatial inclusion to data consistency this should require strong justification.
4. “Surfaces must provide exhaustive cover” can not be enforced at an S-100 level by validation tests as product specifications are not obligated to include coverage features. This could be enforced by validation tests within individual product specifications.

Recommendations

1. Alter Section 7-4.3 in line with suggested edits
2. Include UML and explanatory text mapping each level to the subset of the ISO19107 model it implements
3. Ensure sufficient test data for each level is available which tests each defining aspect of each level.
4. Define the levels which each encoding is capable of describing (ISO8211 = all levels, Part 10b only level 1 and 2a)
5. Include descriptions for both ISO8211 and GML in validation tests.

Justification and Impacts

Action Required of S-100TSM

The S-100TSM is invited to:

a. Note the proposed revisions to S-100 Part 7
b. Comment on the content of each of the levels and their structure
c. Contribute to a revision of the section in line with the aims outlined in this paper.

Annex 1. Revision to S-100 7-4.3

Current content of S-100 Part 7-4.3

7-4.3 Geometry configurations

Figure 7-3 depicts a one size fits all geometry model which can be further constrained in both dimensionality and complexity. This is broken down into 5 basic levels.

7-4.3.1 Level 1 – 0-, 1-Dimension (no constraints)

A set of isolated point and curve primitives. Curves do not reference points (no boundary), points and curves may be coincident. Areas are represented by a closed loop of curves.

7-4.3.2 Level 2a – 0-, 1-Dimension

A set of point and curve primitives with the following constraints:

1) Each curve must reference a start and end point (they may be the same).
2) Curves must not self-intersect as shown in Figure 7-5.
3) Areas are represented by a closed loop of curves beginning and ending at a common point.
4) In the case of areas with holes, all internal boundaries must be completely contained within the external boundary and the internal boundaries must not intersect each other or the external boundary. Internal boundaries may touch other internal boundaries or the external boundary tangentially (that is at one point) as shown in Figure 7-6.
5) The outer boundary of a surface must be in a clockwise direction (surface to the right of the curve) and the curve orientation positive. The inner boundary of a surface must be in a counter-clockwise direction (surface to the right of the curve) and the curve orientation negative as shown in Figure 7-7.

7-4.3.3 Level 2b – 0-, 1-Dimension

A set of point and curve primitives. The constraints for Level 2a apply plus the following:

1) Each set of primitives must form a geometric complex;
2) Curves must not intersect without referencing a point at the intersection;
3) Duplication of coincident geometry is prohibited.

7-4.3.4 Level 3a – 0-, 1- and 2-Dimension

A set of point, curve and surface primitives. The constraints for Level 2a applies.

7-4.3.5 Level 3b – 0-, 1- and 2-Dimension

A set of point, curve and surface primitives. The constraints for Levels 2a and 2b apply plus the following:

1) Surfaces must be mutually exclusive and provide exhaustive cover.

Proposed revised content for Part 7 4.3

7-4.3 Geometry configurations

Figure 7-3 depicts a one size fits all geometry model which can be constrained in both dimensionality and complexity. This is broken down into 5 levels in S-100 with product specifications defining geometry against one chosen level. Product specifications may further constrain geometry.

7-4.3.1 Level 1: 0,1-Dimension (no constraints)

Features are associated with a set of isolated, uniquely identified, Point and Curve primitives only. Curves do not reference points (no boundary). Points and Curves may be coincident. Polygons are represented by a (closed) Curve with coincident start and end Points.

7-4.3.2 Level 2a: 0,1-Dimension

Features are associated with a set of uniquely identified Point and Curve primitives only, with the following constraints.

1. Each Curve must reference a start and end point which may be the same.
2. Curves must not self-intersect as shown in Figure 7.5
3. Polygons with zero or more holes are represented by a sequence of closed loops of (one or more) Curves, each loop beginning and ending at a common Point.

4. In the case of Polygons with holes, all internal boundaries must be completely contained within the external boundary and the internal boundaries must not intersect each other or the external boundary except tangentially (that is, at one Point) as shown in Figure 7.8

7-4.3.3 Level 2b: 0,1-Dimension

Features are associated with a set of uniquely identified Point and Curve primitives only. The constraints for Level 2a apply plus the following:

1. The set of primitives must form a set of geometric complexes.
2. Curves must not intersect without referencing a Point at the intersection
3. Duplication/Coincident geometry is prohibited

7-4.3.4 Level 3a: 0,1,2-Dimensions

Features are associated with a set of uniquely identified Point, Curve and Surface primitives only. Polygons are expressed only as Surfaces. The constraints for Level 2a apply plus the following:

1. The exterior boundary of a Surface must be in a clockwise direction (Surface to the right of the Curve). Interior boundaries must be in a counter-clockwise direction (Surface to the right of the Curve) as shown in Figure 7.7

7-4.3.5 Level 3b: 0,1,2-Dimensions

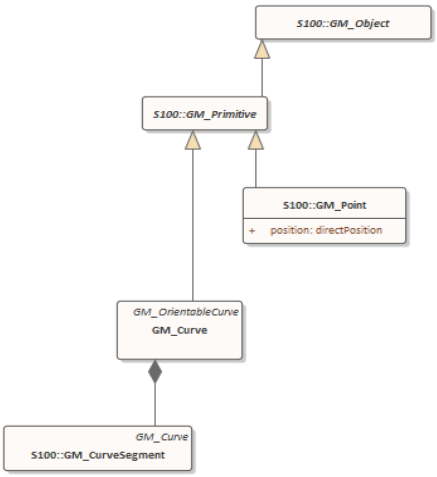
Features are associated with a set of uniquely identified Point, Curve and Surface primitives only. Polygons are expressed only as Surfaces. The constraints for Levels 2a,2b and 3a apply plus the following:

1. Surfaces must be mutually exclusive

Annex 2 – Individual Geometry Levels and Validation Test proposals.

7-4.3.1 Level 1: 0,1-Dimension (no constraints)

Features are associated with a set of isolated Point and Curve primitives only. Curves do not reference points (no boundary). Points and Curves may be coincident. Polygons are represented by a (closed) Curve with coincident start and end Points.



The diagram shows the ISO19107 elements defined by Level 1. In the ISO8211 encoding these are just Curve and Point fields (and their subfields). This defines a conformant Curve, Curve Segment and Point with no multiplicities, aggregations or composites.

Validation tests are, therefore,
[1.1] For each encoded ISO8211 field not of geometry type Curve or Point. The allowable field/subfield combinations are:
From 10a-5.5.2 PRID/[C2IT,C3IT,C2FT,C3FT]
From 10a-5.7.2 CRID/[SEGH]
[For each record of type Multi Point (10a-5.6.2-MRID), Surface (10a-5.9.2-SRID)or composite curve (10a-5.8.2-CCID)]

Index	Level	Short	Text	Ref
1	1	Duplicate identifiers	For each encoded ISO8211 PRID or CRID where RCNM/RCID is not unique	
2	1	Oriented Features	For each feature with geometric primitive point, curve or surface where the spatial association field (SPAS) has ORNT != 255 (For each record of type Multi Point (10a-5.6.2-MRID), Surface (10a-5.9.2-SRID)or composite curve (10a-5.8.2-CCID)	
3	1	Referenced curve start/end points	For each Curve record with a non-null PTAS record	
4	1	Non-closed loop curve for a surface feature	For each feature with geometric primitive surface where the end point of the referenced curve(s) is/are not coincident with the start point	
5	*1*	*Individual segments repeat st/en points*	*For each feature with geometric primitive surface with >1 segments where the individual line segments do not have coincident start/end points (e.g. end segment N != start segment N+1)*	
6	1	Self-intersecting curves	For each Curve which self-intersects	
		Curve Segment type	Each curve segment must be of a single type only and contain correct parameters (is this an ISO8211 test?)	

[1.1a] For each encoded ISO8211 PRID or CRID where RCNM/RCID is not unique.
[1.2] For each feature with geometric primitive point, curve or surface where the spatial association field (SPAS) has ORNT != 255.
[1.2.1] For each Curve record with a non-null PTAS record
[1.3] For each feature with geometric primitive surface where the end point of the defined curve is not coincident with the start point
[1.4] For each feature with geometric primitive surface with >1 segments where the individual line segments do not have coincident start/end points (e.g. end segment N != start segment N+1)

It seems ISO19107 uses the repeated idea in segments.

Construction can begin with the creation of the points. There is a minor issue here since GM_Point, being a type, cannot be instantiated. To be a compliant application schema, a instantiable class that is a subtype of GM_Point must be included, and this class would have to be substituted in the creation cascade below for each use of GM_Point. First, the 7 GM_Points, indicated by dots and identified as {P1, . . . P7} are created:

```
P1 = GM_Point < position = < 1.00, 5.00 > >
P2 = GM_Point < position = < 3.00, 5.00 > >
P3 = GM_Point < position = < 3.00, 2.00 > >
P4 = GM_Point < position = < 1.75, 2.75 > >
P5 = GM_Point < position = < 1.50, 4.50 > >
P6 = GM_Point < position = < 2.00, 3.25 > >
P7 = GM_Point < position = < 5.00, 4.00 > >
Insert P1, P2, P3, P4, P5, P6, P7 into Datastore
```

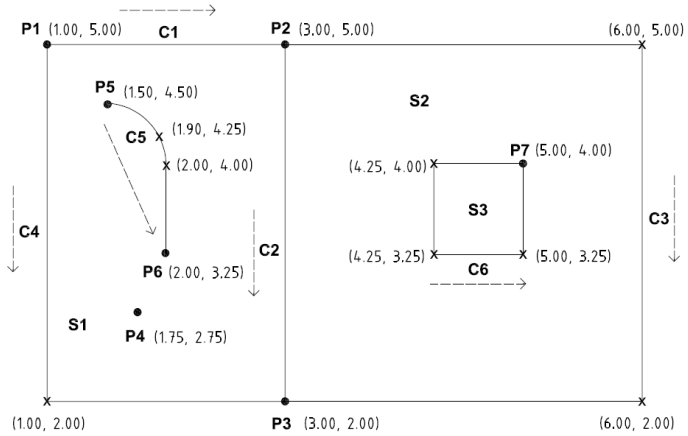


Figure C.1 — A data set composed of the GM_Primitives

With the existence of the points, the cascade can continue with the construction of the 7 GM_CurveSegments, identified {CS1, CS2, CS3, CS4, CS5, CS6, CS7} which can be used to construct the curves to follow. Recall that subtypes of GM_CurveSegment are data types and cannot hold persistent identification. Thus, the variables used to define the curve segments below are "heap" or local variables, defined within the context of the construction, but not persistently stored until they are included as members of an object type (in this case, the curves defined later). All of the curve segments defined here are either line strings or arcs.

```
CS1 = GM_CurveSegment <controlPoint = <P1,P2>, interpolation = "linear" >
CS2 = GM_CurveSegment <controlPoint = <P2,P3 >, interpolation = "linear" >
CS3 = GM_CurveSegment <controlPoint = <P2,(6,5),(6,2),P3>,
      interpolation = "linear" >
CS4 = GM_CurveSegment <controlPoint = <P1,(1,2), P3> ,
      interpolation = "linear" >
CS5 = GM_CurveSegment <controlPoint = <P5,(1.9,4.25), (2,4)>
      interpolation = "arc">
```

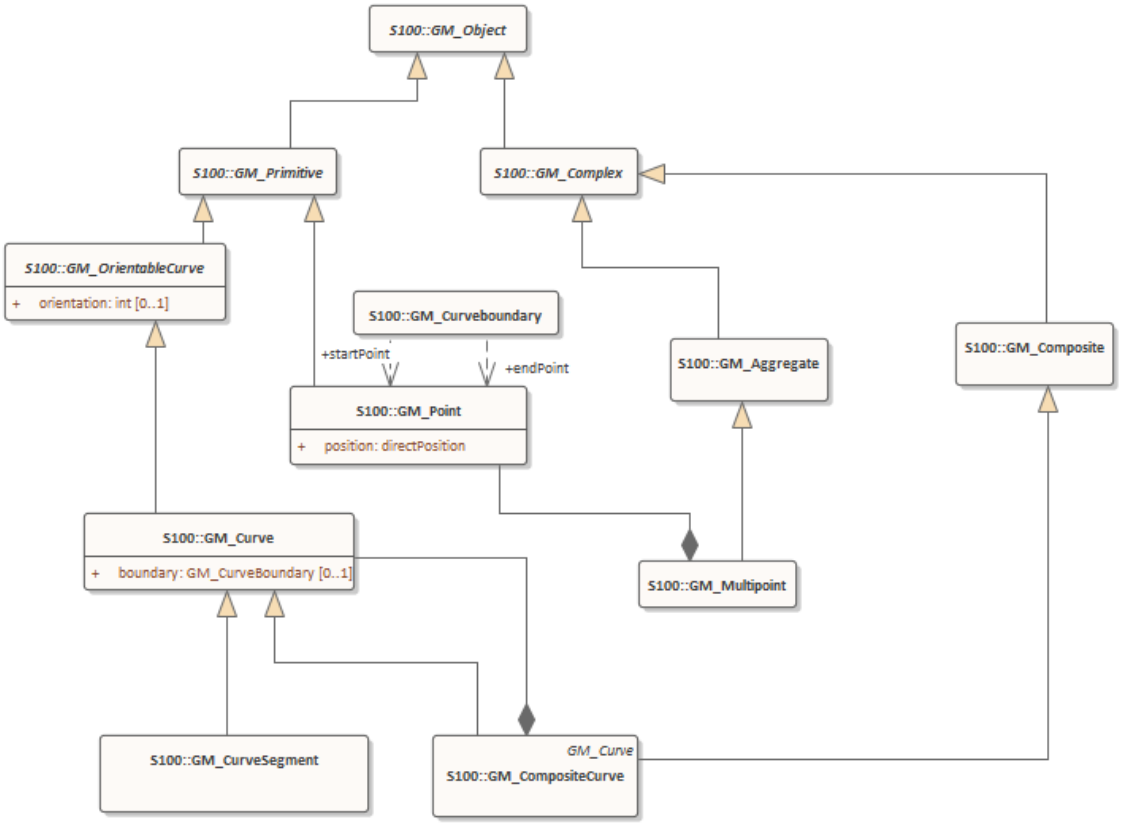
[1.5] For each Curve which self-intersects? Or maybe this is introduced at level 2?

1. Check that only Curve and Point Records are used within the geometry records.
2. Curve and Point records must have unique identifiers.
3. Curve Records should have no boundary (no start end points defined)
4. Feature Associations should have no ORNT defined.
5. Check polygons are simple (no holes allowed at this level)
6. Check polygons are closed
    a. start point of curve = end point of curve
    b. segment start/end points agree

7-4.3.2 Level 2a: 0,1-Dimension

Features are associated with a set of Point and Curve primitives only, with the following constraints.

1. **Each Curve must reference a start and end point which may be the same.**
2. **Curves must not self-intersect as shown in Figure 7.5**
3. **Polygons with zero or more holes are represented by a sequence of closed loops of (one or more) Curves, each loop beginning and ending at a common Point.**
4. **In the case of Polygons with holes, all internal boundaries must be completely contained within the external boundary and the internal boundaries must not intersect each other or the external boundary except tangentially (that is, at one Point) as shown in Figure 7.8**



The diagram shows the Level 2a geometry elements of the ISO19107 model. At this level, boundaries and aggregations are defined, as is the composite curve model. All derived curve types in S-100 Part 7 are also enabled.

Validation tests are:

[2a.1] For each record of type surface (10a-5.9.2-SRID) [or other prohibited records]
[2a.2i] For each curve record without a defined PTAS field or with >2 PTAS fields
[2a.2ii] For each curve record with a single defined PTAS field with subfield TOPI != 3
[2a.2iii] For each curve record with two PTAS fields with no TOPI subfield = 1 or no TOPI subfield = 2

Index	Level	Short	Text	Ref
1	2a	No Surfaces	For each record of type surface (10a-5.9.2-SRID) [or other prohibited records?]	
2	2a	All curves must have one or two start/end points	For each curve record without a defined PTAS field or with >2 PTAS fields	
3	2a	All curves must have a start and an end	For each curve record with two PTAS fields with either (no TOPI subfield = 1) or (no TOPI subfield = 2)	
4	2a	Composite curves must be well defined.	Each composite curve must be continuous. The end point of each component (taking into account ORNT	

			orientation) must be a duplicate of the start point of the next curve component (again, taking into account ORNT).	
5	2a	Exterior (first closed loop) must contain all other closed loops (interior(s))	For each Polygon (referenced by features with geometric primitive surface) which have >1 closed loops, the first closed loop must contain all other closed loops.	
6	2a	Interiors must not touch in a polygon (with holes)	For each Polygon (referenced by features with geometric primitive surface) which have >1 closed loops, the interiors must not touch except tangentially.	
7	2a	Exterior must not touch interior in a polygon (with holes)	For each Polygon (including Polygons referenced by features with geometric primitive surface in level 3a/b) which have >1 closed loops, the exterior must not touch any of the interiors ***except tangentially***.	
8	2a	No self-intersection (simple intersection)	Curves and Composite curves (including Polygons referenced by Surface features in level 3a/b) must not self-intersect [defined with reference to diagram]	1.6
9	2a	No self-intersection (repeated vertices)	Curve Segments must not start or end with repeated vertices. Segments must not repeat any vertex (this constitutes self-intersection)	
10	2a	No self-intersection (number of vertices)	A closed curve must have two or more intermediate vertices (or it self-intersects)	
	2a	Everything Curve is oriented (unlike L1)	For each feature referencing a Curve or Composite Curve with Orient = 255 (null)	
	2a	Well formed segments		1.5

1. Only the following ISO8211 records are allowed for datasets defined at this level: {Point, Multipoint, Curve, Composite Curve}
2. Each record must be uniquely identified as per encoding.
3. Each curve segment must be of a single type
4. Each curve must have a start and end point
5. Polygons must be well-defined ("**sequence of closed loops of (one or more) Curves, each loop beginning and ending at a common Point")**
    a. Curves composed of a sequence of composite curve components, defined by CUCO must have start/end points equal (by reference)
    b. Individual curves with >1 curve segment must be well defined (start/end points equal for each segment)? Composite curves….?
6. Polygon boundaries must be geometrically consistent
    a. For polygons defined with >1 closed loops the first in the sequence (defined as the exterior) must contain all the other closed loops.
    b. Interiors must not touch
    c. Interiors may only touch exterior tangentially

7-4.3.3 Level 2b: 0,1-Dimension

Features are associated with a set of Point and Curve primitives only. The constraints for Level 2a apply plus the following:

1. **The set of primitives must form a set of geometric complexes.**
2. **Curves must not intersect without referencing a Point at the intersection**
3. **Duplication/Coincident geometry is prohibited**

There is no diagram depicting Level 2b, the primitives used are the same as Level 2a. The only difficult requirement to capture is (1) above. S-100's definition of geometric complex is shown below.

> **3-6.5.4.3 Geometric complexes**
>
> Geometric complexes are used to represent the spatial characteristics of a feature as a set of connected geometric primitives. In addition, instances of GM_Complex allow geometric primitives to be shared by the spatial attributes of different features. There are no explicit links between the GM_Primitives in a GM_Complex; the connectivity between the GM_Primitives can be derived from the coordinate data.
>
> 1) A GM_Complex shall be used as the value for a spatial attribute that represents a feature as a collection of connected GM_Objects, which are disjoint except at their boundaries. Subclasses of GM_Complex may be specified to constrain the structure of the GM_Complex used to represent a particular spatial configuration.
>
> 2) Features that share elements of their geometry shall be represented as GM_Complexes that are subcomplexes within a larger GM_Complex.

And ISO19107's definition.

> **4.45**
> **geometric complex**
> **set** of disjoint **geometric primitives** where the **boundary** of each **geometric primitive** can be represented as the union of other **geometric primitives** of smaller dimension within the same **set**
>
> NOTE The **geometric primitives** in the set are disjoint in the sense that no **direct position** is **interior** to more than one **geometric primitive**. The set is closed under **boundary operations**, meaning that for each element in the **geometric complex**, there is a collection (also a **geometric complex**) of **geometric primitives** that represents the boundary of that element. Recall that the **boundary** of a point (the only 0D primitive object type in geometry) is empty. Thus, if the largest dimension geometric primitive is a solid (3D), the composition of the boundary operator in this definition terminates after at most three steps. It is also the case that the boundary of any object is a **cycle**.

> A GM_Complex (Figure 25) is a collection of geometrically disjoint, simple GM_Primitives. If a GM_Primitive (other than a GM_Point) is in a particular GM_Complex, then there exists a set of primitives of lower dimension in the same complex that form the boundary of this primitive.

> A geometric complex (GM_Complex) is a set of primitive geometric objects (in a common coordinate system) whose interiors are disjoint. Further, if a primitive is in a geometric complex, then there exists a set of primitives in that complex whose point-wise union is the boundary of this first primitive.
>
> A subcomplex of a complex is a subset of the primitives of that complex that is, in its own right, a geometric complex. A supercomplex of a complex is a superset of primitives that is also a complex. These definitions are essentially subset and superset with the added restriction that they must be a complex. A complex is maximal if it is a subcomplex of no larger complex.
>
> The boundary of a geometric object in a geometric complex is a subcomplex of that complex. The simplest complex is a single point. The simplest 1-dimensional complex is a curve with its two end points. The simplest 2-dimensional complex is a surface with its boundary curve, and the curve's start and end points.

S-100's encoding do not contain explicit encodings of geometric complexes, they are all derived from S100_GM_Complex as either S100_GM_Multipoint or S100_GM_CompositeCurve.

As the ISO8211 encoding is explicitly by reference (i.e. there is no "inline" geometry) the geometric complex requirement is equivalent to one where the interiors are spatially disjoint, i.e where the interiors of each primitive making up the complex are within a common set.

So, in S-100 Level 2b the validation requirements are

1. Tests from Level 2a
2. As there are no surfaces in Level 2b this is equivalent to mutually exclusive polygons (represented as a series of closed loops)
3. Curve primitives must not intersect without referencing a point at the intersection
4. All geometry primitives must be unique.

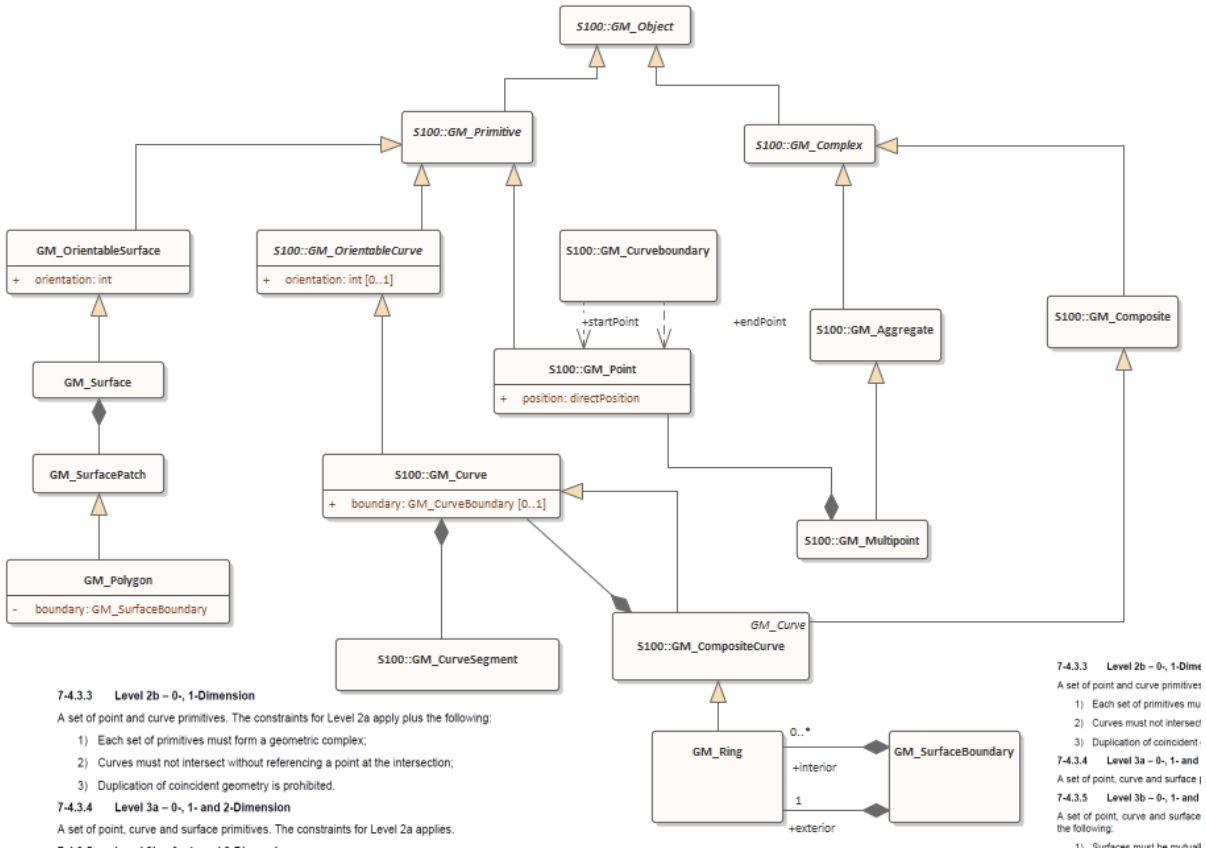
The tests are:

[2b.2] For each curve segment which intersect (without a referenced point at the intersection)
[2b.3] For each coincident geometric primitive either point or curve – coincident is identical position(s) with duplicate identifiers.
[2b.4] For each spatially overlapping polygon

7-4.3.4 Level 3a: 0,1,2-Dimensions

Features are associated with a set of Point, Curve and Surface primitives only. The constraints for Level 2a apply plus the following:

1. **The exterior boundary of a Surface must be in a clockwise direction (Surface to the right of the Curve). Interior boundaries must be in a counter-clockwise direction (surface to the right of the Curve) as shown in Figure 7.7**



The diagram illustrates the full set of primitives required for Level 3a. Essentially this is the full diagram from S-100. It adds surfaces to the levels of abstraction and makes a formal definition of polygons (with >=0 holes) at the geometry level (rather than them being "expressed" through sequences of curves)

Validation tests at this level are, therefore:

[3a.1] For each Geometry field with RCNM != (Level 2a + Surfaces)
[3a.2] For each Feature of geometric primitive Surface which is associated with a non-Surface geometry field. (RCNM != 130). [Question – Surface Features MUST associate with Surfaces]
[3a.3bi] For each surface where the exterior boundary shares more than one node with an interior boundary
[3a.3bii] [segment start/end points as per previous section]
[4i] For each feature of primitive surface, where the exterior boundary is not oriented clockwise
[4ii] For each feature of primitive surface, where an interior boundary is not oriented anti-clockwise

1. Point, Curve, Composite Curve, Surface fields and subfields. These are:
    a. 10a-5.4 Curve
    b. Composite Curve
    c. The derived curves
2. Surface features must be associated with surface geometry components only.
3. Validation tests (3) and (4) from Level 2a are re-expressed as Surface validation tests.
    a. Surfaces must be well-defined

i. A surface is composed of a sequence of well-defined curve and composite curve components, defined by CUCO and must have start/end points (of the surface) equal (by reference)
ii. Individual curves composed of n>1 curve segments must be well defined (start point for segment n+1 = end point of segment n)

b. Surface boundaries must be geometrically consistent
i. The Surface exterior(s) must contain all interiors.
ii. Interiors must not touch [Or, are interiors allowed to touch tangentially too – DIAG S-101 answers this. Probably belongs at S-100 level for consistency]
iii. Interiors may only touch exterior tangentially

4. Orientation constraints.
a. A feature associated to a surface must define orientation correctly using the ORNT subfield of the SPAS field. This must place the surface to the right of its exterior boundary and all interior boundaries.

7-4.3.4 Level 3b: 0,1,2-Dimensions

Level 3b geometry is equivalent to Level 2b with the addition of surface primitives. The disjoint interiors requirement of Level 2b applies to surfaces by requiring that all surfaces do not intersect (except at boundaries). The extra constraint is actually a consequence of adding surface primitives to the list of allowed primitives. Exhaustive cover can not be enforced at an S-100 level as no explicit coverage features are defined in S-100 (they are defined at product specification level)

Extra Validation tests.

[3b.1] For each spatially intersecting surface