- December

S-100 - Part 10b

GML Data Format

- December

Page intentionally left blank

- December

Contents

	-	
10b-1	Scope	1
10b-2	Conformance	1
10b-3	References	1
10b-3.1	Non-normative references	2
10b-4	Introduction	2
10b-5	General concepts	2
10b-6	Notation and diagram conventions	2
10b-7	Components and relationships to standards	2
10b-7.1	Use of profile	<u>4</u> 3
10b-7.2	Interpretations	<u>5</u> 3
10b-8	Profile for feature data	<u>5</u> 4
10b-8.1	Feature and information types	<u>5</u> 4
10b-8.2	Feature collections	54
10b-8.3	Associations	<u>5</u> 4
10b-8.3.1	Association classes	64
10b-8.4	Data types	75
10b-8.4.1	Primitive types	75
10b-8.4.2	Value types	86
10b-8.4.3	Other data types	86
10b-8.5	Spatial types	86
10b-8.5.1	Geometric primitives	
10b-8.5.2	Curve Interpolation	86
10b-8.5.3	Geometric complex, geometric composites, and geometric aggregates	
10b-8.5.4	Inline and by-reference encoding	97
10b-8.5.5	Envelope	1 <u>0</u> 8
10b-8.5.6	Masking truncation and scale ranges	108
10b-8.6	Unsupported GML functionality	1210
10b-8 7	Compliance levels	1310
10b-9	S-100 base schema for feature data	1311
105-5 106-0 1	Introduction	1311
10b-0.1	Features	1411
100-3.2 10h-0 3	Information types	1512
10b-0.0	Spatial types	1512
100-9.4 10b-0.4 1	Inline and referenced geometry	1613
106-9.4.1	Spatial types defined in base schema	1613
100-9.4.2	According	1612
100-9.5	Conorio togo for acceptiona	1015
100-9.5.1	Belo name co property element	<u>10</u> +9
100-9.5.2	Detect general information	<u>19</u> +0
100-9.0	Dataset general information	<u>20</u> +7
100-9.0.1	Dataset identification	<u>20</u> +7
100-9.0.2	Dataset structure information	<u>22</u> +9
100-9.7	Coordinate Deference System	<u>24</u> 20
100-9.8	Detected structure definition	<u>25</u> 21
100-9.9	Dataset structure definition	<u>25</u> 24
100-9.9.1	Dataset metadata	<u>25</u> 21
100-10	Constraints and validation	<u>25</u> 21
100-11	Dataset level metadata and integrity checks	<u>20</u> 22
10D-12	Schema locations and namespaces	<u>2622</u>
10D-13	Divergences from common GNL practices	<u>2622</u>
100-14	Conventions for 5-100 GiviL data formatives	<u>20</u> 22
10D-15	Processing of GiviL datasets (informative)	<u>21</u> 23
Appendix	Tub-A Application Schema (Informative)	<u>30</u> 25
IUD-A-1.	Example of a GML application schema for an S-100-based data product	<u>30</u> 25
Appendix	IUD-B Use of Profile in GIVIL Application Dataset (Informative)	<u>34</u> 29
100-B-1.	Introduction	<u>34</u> 29
100-B-Z.	Dataset sumplies in VML (CML	<u>34</u> 29
100-D-3.	Dalasel examples III AIVIL/GIVIL	<u></u>

- December

- December

Page intentionally left blank

December 2018

10b-1 Scope [ALL]

This Part specifies a profile of GML meant to be used as a basis for the development of GML application schemas for S-100 data products. The GML application schema for each <u>S-100</u> data product defines a file format for the machine-to-machine exchange of information structured in conformance with the application schema for the data product <u>and matching the content of the Feature Catalogue</u>, as defined in the appropriate product specification.

The scope of this Part includes:

- Feature and information type data conforming to the S-100 General Feature Model defined in Part 3, encoded using GML (ISO 19136), and structured as datasets (identifiable collections of data).
- 4)2) Construction of GML Schemas from Feature Catalogues
- 2)3) Guidance for use of the schemas in application schemas for data products.

The following are outside scope:

- 1) A format for updates to datasets.
- 2) Interchange by means other than datasets encapsulated as files, such as Web Feature Service (WFS), other Web services, email, etc.
- Information not encapsulated using GML, such as feature catalogues, exchange set metadata, portrayal catalogues, and support files in other XML formats.
- 4) Tools for developing GML application schemas for data products.
- 5) Design and programming of software for processing GML data.
- 6) Gridded and coverage data.

10b-2 Conformance [ALL]

The profile described in this Part conforms to the requirements for GML profiles described in ISO 19136.

10b-3 References [ALL]

ISO 19106:2003, Geographic information - Profiles ISO 19107:2003, Geographic information - Spatial schema ISO 19111:2007, Spatial referencing by coordinates (coordinate reference systems) ISO 19118:2005, Geographic information - Encoding ISO 19123:2005, Schema for coverage geometry and functions ISO 19136:2007, Geographic information – Geography Markup Language ISO 19136-2:2015, Geographic information – Geography Markup Language ISO/TS 19139, Geographic information - Metadata - XML schema implementation ISO/IEC 19757-3, Information technology - Document Schema Definition Languages (DSDL) - Part 3: Rule-based validation - Schematron IETF RFC 2396, Uniform Resource Identifiers (URI): Generic Syntax IETF RFC 3986, Uniform Resource Identifiers (URI): Generic Syntax W3C XLink, XML Linking Language (XLink) Version 1.0, W3C Recommendation W3C XML Namespaces, Namespaces in XML, W3C Recommendation W3C XML, Extensible Markup Language (XML) 1.0, W3C Recommendation W3C XML Schema Part 1, Structures, W3C Recommendation W3C XML Schema Part 2, Datatypes, W3C Recommendation

1

S-100 Edition 4.0.0	December 2018
LEIRI, <i>Legacy Extended IRIs for XML Resour<u>c</u>e Identifica</i> URL: <u>http://www.w3.org/TR/leiri</u>	tion, W3C Working Group Note 3.
10b-3.1 Non-normative references	
The following references are listed only for informative p document. Drafts are subject to change and are not internated and are not inte	urposes or to clarify parts of this tional standards.
ISO/DIS 19107, Geographic information – Spatial schema (Draft – June 2018) Formatted: English (United States)
10b-4 Introduction [ALL - Comments]	

The S-100 GML profile defines the core GML components that shall be used in GML encodings for S-100 data products. This profile defines a restricted subset of XML and GML types that excludes GML features not required by S-100 GML datasets. This_The_profile of GML is contained in multiple Schema files a single file and reduces the complexity of the full GML encoding to a more manageable level Part of the schema A separate XML schema dd efines common elements and types needed for all S-100 datasets encoding feature-based information.

10b-5 General concepts

A GML application schema is an XML schema that conforms to the rules for application schemas given in the GML specification (ISO 19136).

A GML document is an XML document with a root element conforming to the rules for GML data specified in the GML specification (ISO 19136). Specifically, in the context of S-100 this means the root element must be a GML AbstractFeature or Dictionary element, or in a substitution group of any of these elements.

The terms "GML application schema" and "application schema" as used in this Part mean respectively an *XML schema* and a *conceptual schema*. The former may be an XSD file conforming to the XML schema rules, the latter a UML diagram, <u>conforming to the S-100</u> application schema rules.

These terms and definitions conform to ISO 19101 and ISO 19136. Complete definitions are given in ISO 19101 and ISO 19136, and are reproduced in Annex A.

10b-6 Notation and diagram conventions

Diagram element	Meaning					
e or terment	XML Schema <sequence></sequence>					
e or €	XML Schema <choice></choice>					
0∞	XML schema multiplicity constraints (here, "0" and "unbounded")					

10b-7 Components and relationships to standards [JP]

The GML data <u>format encoding</u> for S-100 consists of the following components, realized as separate XML schemas:

 An XML schema that defines a GML profile ("Profile"). This is a restricted subset of types and elements ("XML constructs") defined in the GML 3.2.1 schemas. XML constructs not needed for S-100 data products are excluded.

December 2018

2) An XML schema defining additional XML constructs ("S100base"). This schema uses the GML profile schema. The constructs defined in this schema are expected to be needed in order for a product specification conforming to the S-100 standard to define a format for datasets.

2)

The figure below illustrates the dependency relationships. The GML encoding standard (ISO 19136:2007) provides an implementation schema (XML Schema) for the ISO 19100 conceptual schemas. The S-100 GML profile is a subset of the constructs defined by the GML implementation schema. S-100 common elements are defined in a common elements XML schema conforming to the profile. GML formats for specific data products use the constructs in the common elements schema in defining XML types and elements corresponding to feature and information types defined by the relevant Product Specification. A dataset is an XML file conforming to the GML data format (GML application schema).



Figure 10b-1 – Derivation of profile and its use by a data product

10b-7.1 Use of profile

The typical use of the profile is to define a format for GML files encapsulating datasets packaged as_computer files.

Formats for modes of exchange other than datasets are not required to use these schemas.

The S-100 GML Profile has been defined to support validation and an application schema may import the GML 3.2.1 schemas or replace the <import> statement in the common elements with an <import> of the GML 3.2.1 schemas. To enable validation engines to understand that the S-100-based Application Schema adheres to a profile, the S-100 GML Profile must be declared

December 2018

within the schema. This allows the validation engine to select this schema instead of the GML 3.2.1 schema to validate the data against.

A GML Application schema mirrors the feature/attribute names, multiplicities, types and relationships which are defined in the feature catalogue. An implementing system should be capable of unambiguously parsing a GML dataset without recourse to the schema for details of its elements and types.

10b-7.2 Interpretations

The schema components of the GML profile which are part of the "gml" namespace are to be interpreted in conformance with GML as defined by ISO 19136, with the exceptions below. 1) "Linear" curve interpolation type in datasets using a geographic CRS shall be interpreted as loxodromic interpolation.

10b-8 Profile for feature data [HA]

10b-8.1 Feature and information types

The profile supports the ability to encode classes defined as identifiable objects as derived from either the abstract GML type or abstract feature type:

- AbstractGML (can shall be used to derive S-100 Information Classes).
- AbstractFeature (canshall be used to derive S-100 Feature Classes).

The S-100 GML Profile prohibits the use of the gml:StandardObjectProperties group.

10b-8.2 Feature collections

A feature collection is a collection of feature instances. Within GML 3.2.1, the generic gml:FeatureCollection element has been deprecated. A feature collection is any feature class with a property element in its content model (for example member) which is derived by extension from *gml:AbstractFeatureMemberType*.

In addition, the complex type describing the content model of the GML feature collection may also include a reference to the attribute group *gml:AggregationAttributeGroup* to provide additional information about the semantics of the object collection.

The S-100 GML Profile supports the GML 3.2.1 approach to model a Feature Collection class within an S-100 GML Application Schema.

For exchange of file-based GML data using S-100 the GML Profile includes:

- A FeatureCollection, named "Dataset"
- Generic association elements

Generic dataset metadata types with mandatory dataset metadata elements

Other exchange mechanisms, for example via API mechanisms may wrap GML data encoded in conformance with this part with different feature collection mechanisms.

10b-8.3 Associations

The profile allows associations to be encoded inline<u>or</u>, by reference<u>or</u> either inline or by reference. The dataset metadata field associationEncoding shall be defined as either "reference" or "inline" to define which method is used throughout conforming datasets.

December 2018

For bi-directional associations, the profile supports the encoding of the name of reverse property in the *appInfo* annotation element in the Application Schema XSD. [mandatory?]

10b-8.3.1 Association classes

The profile allows the GML 3.3 convention for encoding of association classes using the GML 3.3 association class conversion rule, which converts association classes to an equivalent intermediate class. The figures below illustrate the conversion rule.

Where associations contain attributes in a product specification feature catalogue, this structure shall be used to realise those attributes via an intermediate Information Type



Figure 10b-2 – Model with association classes (from OGC 10-129r1 / ISO 19136-2:2015)

Part 10b - GML Data Format

Formatted: Font: Bold

December 2018





10b-8.4 Data types

10b-8.4.1 Primitive types

The S-100 GML Profile supports the primitive types defined in Part 1, clause 1-4.5.2 with the following equivalents. Types are supported using XML Schema ("xs:") built-in data types where possible except as noted:.

S-100 Primitive Type	GML Profile Equivalent
Boolean	<u>xs:boolean</u>
Integer	<u>xs:integer</u>
Real	xs:decimal
CharacterString	xs:string
Date	S100_TruncatedDate
Time	<u>xs:time</u>
DateTime	xs:dateTime
S100_TruncatedDate	S100_TruncatedDate
URI, URN, URL	xs:anyURI

Boolean

ż

Table 14: S-100 Primitive Types

- <u>All S-100 types referred</u>Integer
- Real
- CharacterString
- Date
- Time
- DateTime
- S100_TruncatedDate

URI, URN, URL to in the above table are defined within the S-100 GML Profile. The profile also provides common restrictions of types for non-negative Decimals and Decimals in the range 0.0-360.0 (for ° representations).

•____

Types may be supported using XML Schema built-in data types.

7

Part 10b - GML Data Format

	Formatted: No bullets or numbering
	Formatted: Font: Bold
Ν	Formatted: Centered
Ľ	Formatted: Font: Bold
$\left(\right)$	Formatted: Font: Bold
	Formatted: No bullets or numbering

Formatted: Caption, Keep with next

10b-8.4.2 Value types

The S-100 GML Profile supports the value types defined in Part 1, clause 1-4.5.3.5:

- Measure
- Length
- Angle

NOTE: S100_UnitsOfMeasure type shall be realised by the uom property, the value of which should reference to a value defined in a codelist register which provides the name, definition and symbol.

10b-8.4.3 Other data types

Application schemas describing data formats for data products may make use of built-in data types defined in "W3C XML Schema: Part 2" and additional data types to be defined within the GML Application Schema.

<u>10b-8.4.3</u> Example: The GML application schema encapsulating a data format may support the S-100 data type URI using the XML schema built-in type anyURI and S100_TruncatedDate using one or more of the XML Schema built-in types gYear, gMonthDay, gDay, gYearMonth, gMonthDay, as governed by the product specification.<u>Other data types</u>

S-100 Simple Attributes shall take one of the types defined in table 1. S-100 Complex Attributes have no named type in the GFM. All complex attributes shall have a defined type name "<Name>Type" where <Name> is the S-100 GF Attribute Name defined in the associated Feature Catalogue.

10b-8.5 Spatial types

10b-8.5.1 Geometric primitives

The S-100 GML Profile supports the GML 3.2.1 encodings of basic geometries (Part 7 clause 7-5.1):

- Point
- AbstractCurve
- Curve
- OrientableCurve
- LineStringSegment
- Surface
- LineString
- Polygon
- S100_ArcByCenterPoint
- S100_CircleByCenterPoint

The S-100 GML Profile constrains the GM_CurveInterpolation type values and constrains the curve encoding to a subset of GML curve geometries.

Note: S100_ArcByCenterPoint and S100_CircleByCenterPoint are not the same as the GML primitives ArcByCenterPoint and CircleByCenterPoint.

10b-8.5.2 Curve Interpolation

The list of allowable values consists of a subset of the values allowed by ISO 19136 plus extensions for spline and interpolated curve segments (ISO/DIS 19107 draft – June 2018, clarifies that the list of interpolations in the standard is not exhaustive):

- Linear (linear) in a non-geographic CRS the interpolation is defined by a series of DirectPositions on a straight line between each consecutive pair of controlPoints.
- 2) Linear interpolation (linear) in a geographic CRS (interpreted as Loxodromic) the interpolation method shall return DirectPositions on a loxodromic curve between each consecutive pair of controlPoints. A loxodrome is a line crossing all meridians at the same angle, that is, a path of constant bearing.

Part 10b - GML Data Format

December 2018

Formatted: Strikethrough

Formatted: Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

- 3) Geodesic (geodesic) the interpolation mechanism shall return DirectPositions on a geodesic curve between each consecutive pair of controlPoints. A geodesic curve is a curve of shortest length. The geodesic shall be determined in the coordinate reference system of the GM_Curve in which the GM_CurveSegment is used.
- 4) Circular arc by 3 points (circularArc3Points) the interpolation is defined by a series of 3 DirectPositions on a circular arc passing from the start point through the middle point to the end point for each set of three consecutive control points. The middle point is located halfway between the start and end point.
- 5) Elliptical arc (elliptical) for each set of four consecutive controlPoints, the interpolation mechanism shall return DirectPositions on an elliptical arc passing from the first controlPoint through the middle controlPoints in order to the fourth controlPoint. Note: if the four controlPoints are co-linear, the arc becomes a straight line. If the four controlPoints are on the same circle, the arc becomes a circular one.
- Conic arc (conic) the same as elliptical arc but using five consecutive points to determina a conic section.
- 7) Circular arc with centre and radius (circularArcCenterPointWithRadius) the interpolation is defined by an arc of a circle of the specified radius centred at the position given by the single control point. The arc starts, at the start angle parameter and extends for the angle given by the angular distance parameter. This interpolation type shall be used only with S100_ArcByCenterPoint and S100_CircleByCenterPoint geometry. The precise semantics of the parameters are defined in Part 7 clause 7-5.2.20 (S100_ArcByCenterPoint).
- 8) Polynomial (polynomialSpline) the control points are ordered as in a line-string, but they are spanned by a polynomial function. Normally, the degree of continuity is determined by the degree of the polynomials chosen.
- 9) Bézier Spline (bezierSpline) the data are ordered as in a line string, but they are spanned by a polynomial or spline function defined using the Bézier basis. Normally, the degree of continuity is determined by the degree of the polynomials chosen.
- 10) B-spline (bSpline) the control points are ordered as in a line string, but they are spanned by a polynomial or rational (quotient of polynomials) spline function defined using the B-spline basis functions (which are piecewise polynomials). The use of a rational function is determined by the Boolean flag "isRational". If isRational is TRUE then all the DirectPositions associated with the control points are in homogeneous form. Normally, the degree of continuity is determined by the degree of the polynomials chosen.
- 11) Blended parabolic (blendedParabolic) the control points are ordered as in a linestring, but are spanned by a function that blends segments of parabolic curves defined by triplet sequences of successive data points. Each triplet includes the final two points of its predecessor. Further details of the semantics are provided in Part 7 clause 7-4.2.2.2.

10b-8.5.3 Geometric complex, geometric composites, and geometric aggregates

10b-8.5.3.1 Geometric complex and geometric composites

The S-100 GML Profile supports the following composite geometries (Part 7 clause 7-5.1):

CompositeCurve.

10b-8.5.3.2 Geometric aggregates

The S-100 GML Profile supports the aggregate geometry types (Part 7 clause 7-5.1): • MultiPoint

10b-8.5.4 Inline and by-reference encoding

The S-100 GML Profile supports the ability to encode a geometry either inline or by reference where two features share the same instance of a GM_Object (see Part 3 clause 3-6.5.4.5). Where both are specified the GML convention is followed and the inline reference takes priority.

December 2018

10b-8.5.5 Envelope

The S-100 GML Profile supports the ability to encode an appropriate geometry via bounding box or envelope. The Profile does not constrain the use of the GML implementation of GM_Envelope.

10b-8.5.6 Masking, truncation and scale ranges

Beginning with Edition 5.0.0, the S-100 GML format defines a generic complex type S100 SpatialAttributeType for spatial attributes with *scaleMinimum* and *scaleMaximum* attributes and a *maskReference* tag. These correspond to the attributes of the S100 SpatialAttribute metaclass in the S-100 General Feature Model (Part 3, Figures 3-1 and 3-2 and clause 3-5.3.5). *scaleMinimum* and *scaleMaximum* are implemented as integer attributes. The *maskReference* attribute is implemented using the GML Reference Type with the following constraints:

- The value of the xlink:href attribute must be the gml:id of the masked/truncated object.
 The value of the xlink:role attribute must be either 'truncated' or 'suppressed'.
 - The value of the Xink. Tole attributes is undefined in \$ 100 and as they are not used.
- The meaning of other attributes is undefined in S-100 and so they are not used within
 the GML profile. Product specifications may specify their use depending on the needs
 of the data product.

The structure of the S100 SpatialAttribute Complex type is depicted in Figures 10b-4 and 10b-5 below.



Figure 10b-4 - Structure of generic spatial attribute type in the S-100 GML format

December 2018



Figure 10b-5 – Structure of mask reference type

NOTE: This Part specifies only the *href* and *role* attributes. The other AssociationAttributeGroup members, OwnershipAttributeGroup members, and nilReason are not used.

An example of the use of masking is depicted in Figure 10b-6 below. The surface boundary is defined by reference to two curves (sequential curves comprising the exterior ring), whose gml:id's are JS.C.123 and JS.C.567. These curves are defined elsewhere in the file. The *maskReference* tag in the example indicates that the curve JS.C.567 is suppressed.

S-100	Edition	4.0.0	
-------	---------	-------	--

December 2018

NOTE 1: The S-100 GML format does not require that the object geometry (the Surface object in Figure X.X) be encoded inline as depicted in Figure X.X. It can be encoded elsewhere in the dataset as an separate spatial data object, like the curves. NOTE 2: The '#' character preceding the identifier is an XML convention indicating that the part which follows is the identifier of an XML element inside an XML file (since no filename is specified, the convention is that the referenced element is in the same file). Note that the reference mechanism also allows references to objects in external files by prefixing the object identifier with the file name or URL of the external file. <S127:MilitaryPracticeArea gml:id="JS.MIPARE.676"> <featureName><name>Jussland Naval Firing Area</name></featureName> <textContent><information><text>Example of masked spatial type</text></inform <theServiceHours xlink:href="#JS.SRVHRS.685"/> <geometry> <\$100:surfaceProperty scaleMinimum="1000000" scaleMaximum="10000"> <S100:maskReference xlink:href="#JS.C.567" xlink:role="suppressed"/> <5100:Surface gml:id="JS.MIPARE.676.S.1" srsDimension="2" srsName="u</pre> <gml:patches><gml:PolygonPatch><gml:exterior> <gml:Ring> <gml:curveMember_xlink:href="#JS.C.123"/> <gml:curveMember xlink:href="#JS.C.567"/> </gml:Ring> </gml:exterior></gml:PolygonPatch></gml:patches> </S100:Surface> </S100:surfacePropertv> </geometry> <geometry>

Figure 10b-6 - Example of masking in the GML format

10b-8.6 Unsupported GML functionality

•

Support for GML 3.2.1 and GML 3.3 geometries not defined in ISO 19107 is not included. Specifically, this means CircleByCenterPoint and ArcByCenterPoint (as defined in GML 3.2.1) are not supported, nor are the compact geometry encodings defined in GML 3.3. <u>S-100 Edition</u> 2.0.0 provides different definitions for arc and circles by center point and radius.

The temporal model and temporal primitives defined in ISO 19108, including temporal positions, instants, time periods, are not supported. S-100 data should code dates and times as thematic attributes.

- Dynamic features are not supported by the S-100 GML profile.
- Topology is not supported by the S-100 GML profile.
- Linear Referencing is not supported by the S-100 GML profile.
- Coverages are not supported by the S-100 GML profile.
- The ability to define coordinate reference systems is not supported. The products should be defined using a well-known, pre-defined coordinate reference system such as WGS84.
- Observations are out of scope for the S-100 GML Profile. (The observations schema within GML has been superseded by the OGC (10-025r1) XML encoding for ISO 19156: Observations and Measurements.)

Part 10b - GML Data Format

Formatted: Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

Formatted: Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

December 2018

10b-8.7 Compliance levels

In order for a client to be able to properly interpret a schema, it needs a capability to identify the compliance level of the application schema. An XML Schema annotation shall be used for this purpose. The following schema fragment shows how this annotation shall be declared in an application schema¹:

<annotation>

<appinfo>

<gmlProfileSchema xmlns="http://www.opengis.net/gml/3.2">

http://www.iho.int/S-100/profiles/s100_GMLProfile.xsd

</gmlProfileSchema>

<s100:ComplianceLevel>1</s100:ComplianceLevel>

</appinfo>

</annotation>

Table 10b-1 – Compliance declaration XML code

Compliance level	Description
1	$\ensuremath{S}\xspace{-100}$ feature types, information types, feature and information associations. Point, curve, and surface primitives
2	All features of Level 1, plus circle and arc by center point geometry, splines, and blended interpolations

To manually add the compliance declaration to the schemas after they have been generated involves 3 steps:

1. Add the S-100 GML Profile XML Namespace declaration:

xmlns:s100_profile="http://www.iho.int/S-100/profile/s100_gmlProfile"

- 2. Add the S-100 GML Profile compliance declaration within the schema annotation. The compliance declaration is the XML code in Table 10b-1 above.
- Add an Import statement for the S-100 GML Profile Levels schema. Add the following import statement for the S-100 GML Profile Levels schema into the list of imported schemas to the list of imported schemas:

10b-9 S-100 base schema for feature data

10b-9.1 Introduction

A second XML schema is provided which defines a small set of derived types and elements in an "S100" namespace. The schema defining these common elements and types is technically a "GML application schema" in the sense defined by ISO 19136. It defines GML constructs which are expected to be used by different product specifications to define detailed GML application schemas encoding formats for GML datasets. This schema provides a common core structural paradigm for GML datasets across a variety of application domains. The intention is to reduce the complexity of application development, facilitate sharing of software modules, and information integration and mapping across different application domains, by minimising the proliferation of structural variations. The XML Schemas are designed to ensure a match between the feature catalogue structure and the produced GML Schema.

Elements and types are defined using <u>only</u> the restricted subset of GML defined in the S-100 GML Profile.

¹ Line breaks and spaces have been added for clarity.

S-100 Edition 4.0.0	December 2018
10b-1.1.1 Construction of the xsd header	Formatted: Heading 3, No bullets or numbering
Top level schema element, example based on Product Spec S-	123 Formatted: No bullets or numbering
<pre><rul> <xs:schema< pre=""></xs:schema<></rul></pre>	T20.
xmlns:xs="http://www.w3.org/2001/XMLSchema"	Formatted: Font: Consolas
xmlns:S100="http://www.iho.int/s100gml/1.0"	Formatted: Indent: Left: 0.25", No bullets or numbering
<pre>xmlns:gml="http://www.opengis.net/gml/3.2"</pre>	
<pre>xmlns="http://www.iho.int/S123/gml/1.0"</pre>	
<pre>xmlns:S100EXT="http://www.iho.int/s100gml/1.0+EXT"</pre>	
<pre>targetNamespace="http://www.iho.int/S123/gml/1.0"</pre>	Formatted: Font: Consolas, Bold
elementFormDetault="qualitied"	Formatted: Font: Consolas
<u>version= 1.0.0-201/0831 ></u>	
S-100 part 10b should define the sS tructure or patterns of the namespaces use schema:-	ed in the gml
	Formatted: No bullets or numbering
xmlns:gml=http://www.opengis.net/gml/3.2	
This seems consistent and could be set in part 10b as such-	Formatted: Bulleted + Level: 1 + Aligned at: 0.25" +
<u>xmins:S100= nttp://www.inc.int/S100gmi/1.0</u> <u>Seems to be used consistently but perhaps should be inline with the ver</u>	sion of S-100
and thus the gml profile.	
mins-"http://www.ibo.int/S123/aml/1.0"	Formatted: No bullets or numbering
	Formatted: No bullets or numbering
targetNamespace=http://www.iho.int/S123/gml/1.0	
elementFormDefault="qualified"	Formatted: No bullets or numbering
 The schema default namespace and the target namespace are the sam 	e and should
be formed using a consistent pattern defined in S-100 10b. Perhaps I	based on the
product spec and the product spec version. Using elementFormDefaul	<u>t = 'qualified'</u>
means every element in the dataset must belong to a namespace but it de	bes not mean
that a namespace prefix is needed, a default namespace can be use	ed to set the
namespace for any elements without a specific prefix.	of sult is used
- Note: If the dataset is meant to contain more than one product then no d	erault is used
and every element requires a namespace of namespace prenx.	Formatted: No bullets or numbering
The S-100 Feature catalogue XML model does not include fields to indicat	e the product
catalogue and a catalogue version. Since there is little desire to change the >	(ML FC spec
perhaps we can standardize on making the 'name' of the feature catalogue the	same as the
product specification	<u>identifier.</u>
Currently the S-123 FC name is "Feature Catalogue for S-123" but since it is a FC	C XML file the
Feature Catalogue for" is redundant and it should be good to just make the cat S-123".	alogue name
 version="1.0.0-20170831"	
There are no set rules in xsd for how the user defined version attribute i	s to be used.
Jsing a semantic version would be a recognized pattern but it could also be use he version of the product spec for which the schema was defined. If that inform	d to match to nation can be
pleaned from the targetnamespace pattern then the version attribute may be redu	Indant.
	Formatted: Space After: 0 pt
10b-9.2 Features	
An XML complex type AbstractFeatureType is defined as the base types for <u>a</u> features. The AbstractFeatureType extends the gml:AbstractFeatureType as def 100 GML Profile by adding <u>an optional</u> feature object identifier (clause <u>10b-9.8</u>	II <u>geographic</u> ined in the S- 10b 9.7) and

Part 10b - GML Data Format

|

associations to feature and information objects (clause 10b-9.5). An inverse feature association is also included to allow reverse pointers for feature associations.



Figure 10b-4-7 – Base type definition for S-100 feature elements

10b-9.3 Information types

The common type definition for information types is similar to that for the common feature type, but omits the feature object identifier and feature associations.



Figure 10b-5-8 - Base type definition for S-100 information type elements

10b-9.4 Spatial types

Spatial types are defined as extensions of the corresponding GML spatial types with an information association added, since in S-100 spatial objects can have information associations. The figure below shows the design of the Point type in the S-100 schema. It includes a single gml:Point type and 0 or more associations to S-100 information types.

December 2018



Figure 10b-6-9 - S-100 generic S-100 spatial type for point spatial object

The other spatial types have a similar structure.

10b-9.4.1 Inline and referenced geometry

The base schema also allows geometry to be defined either inline or by reference, conforming with the same ability in GML. <u>Where both are included in a dataset, the inline geometry is normative.</u>

10b-9.4.2 Spatial types defined in base schema

The base schema defines the point, curve, and surface spatial objects, as well as multipoint and composite curve objects. Curves may be simple, composite, or orientable curves. This is the same set defined in the S-100 GML profile (clause 10b-8.5). It also supports *gml:Polygon* (ISO 19136) which is a special surface that is defined by a single surface patch.



Figure 10b-7-10 - Geometry types defined in the base schema

10b-9.5 Associations

Feature and information association properties are defined as extensions of GML feature properties. The pointers to the object at the other end of the association are encoded in the Xlink attributes.

XLink components are the standard method to support hypertext referencing in XML.

December 2018

GML provides an XML Schema attribute group, *gml:AssociationAttributeGroup*, to support the use of Xlinks as the method for indicating the value of a property by reference in a uniform manner in GML. This structure shall be used to encode all associations in the associated feature catalogue.

-ISO 19136 specifies that the value of a GML property that carries an *xlink:href* attribute is the resource returned by traversing the link.

The data types of the attributes are listed in the table below.

Table 10b-2 – Requirements for XLink attributes in associations

Xlink attribute	Data type	Remarks
href	URI	Reference to the object at the other end of the association, for example, the gml:id of an object in the current data set. May be a URI fragment.as described in the XLink specification
role	URI	Optional description of the nature of the target resource, given as a URI
arcrole	legacy extended IRI	Description of the role or purpose of the target resource in relation to the present resource, given as a URI (ISO 19136). May be constructed from the role name from the application schema. The XLink 1.1 specification requires: 1) The value must be a Legacy extended IRI 2) The identifier must not be relative
title	character string	Optional string describing the relationship. Product specifications may constrain its format and define its semantics
show		not used
actuate		not used
type		not used

Product specifications may use one of the two methods described in the following sub-clauses to encode associations. Consistent use of a single method for each data product is strongly recommended.

December 2018

10b-9.5.1 Generic tags for associations

The profile defines two generic tags for feature and information associations, depicted in the figures below.



Figure 10b-8-11 - S-100 type for feature association

Information associations have a structure similar to feature associations except that they are information properties.

S-100 Edition 4.0.0 December 2018 InformationPropertyType O Base Type AbstractAttributeType Content complex AbstractAttributeType (extension base) Base Type gml:AbstractGMLType Content complex Abstract true Abstract type for attributes ⊙ 🧿 Attributes \ominus 🚦 gml:AssociationAttributeGroup (informationAssociation) ⊖ 🔒 xlink:simpleAttrs xlink:type) ① xlink:href xlink:role xlink:arcrole @ xlink:title)⊕ xlink:show 🕀 xlink:actuate nilReason)⊙ XLink components are the standard method to support hypertext referencing in XML. An XML Schema attribute group,. Abstract type for an S-100 information associations

Figure 10b-9-12 – S-100 type for information association

Example (non-normative): Given the XML fragments below, the *MarineProtectedArea* feature has an information association to the *Regulations* information object.

<MarineProtectedArea gml:id="US123450"

<informationAssociation xlink:href="#US50004"

xlink:arcrole="http://www.iho.int/S-122/roles/theRegulations"/>

</MarineProtectedArea>

and elsewhere in the same file:

<Regulations gml:id="US50004">

</Regulations>

10b-9.5.2 Role name as property element

Alternatively, the roles defined in the application schema may be used as a property element of the feature or information type, with XLink attributes providing the reference to the instance. In this case the role at the far end of the association should be used for the XML tag defining

19

...

the property. The role name may be usable as-is for the property tag, or it may have to be mapped to a tag conforming to XML and GML conventions.

Example (non-normative): Given an application schema containing the relationship in the figure below, the *NavigationLine* feature can encode the association as a property element named *navTrack* as below. The format, construction rules, and semantics for the *arcrole* and *title* values would be defined in the product specification.

<NavigationLine gml:id="US123098">

<navTrack xlink:href="#US890321"

xlink:arcrole="urn:iho:s101:1.0:52.2" title="RangeSystem"/>

</NavigationLine>

and elsewhere in the same file:

<RecommendedTrack gml:id="US890321"> <navLine xlink:href="#US123098"

xlink:arcrole="urn:iho:s101:1.0:52.1" title="RangeSystem"/>

</RecommendedTrack>

«FeatureType» NavigationLine	+navLine		+navTrack	«FeatureType» RecommendedTrack
	1•	RangeSystem	0*	

Figure 10b-10-13 - Association in application schema

10b-9.6 Updating

Datasets may have two purposes.

- 1. Base datasets containing all features, information types and associations within a specific coverage area, for a given dataset issue date.
- 2. Numbered update datasets, containing only updates to individual features within an earlier base dataset.

Updates are numbered, with a sequential update number, starting at 1, listed in the dataset metadata. A dataset's purpose is listed in the dataset metadata element and in the catalogue metadata for the exchange set in which it is contained.

Updates are only defined at a feature or information type level only. They can only add new features or replace earlier definitions of individual dataset features. Updated features shall retain the GML identifier issued in the base dataset. It is not possible to update associations except by replacing all features affected by a change to references. It is not possible to delete features through update.

10b-9.610b-9.7 Dataset general information

10b-9.6.110b-9.7.1 Dataset identification

Dataset identification information is defined by the complex type *DatasetIdentificationType*. The fields are shown in the table and figure below.

Table 10b-3 – Dataset identification header elements

Field	XML Tag	Value	Mult	Туре	Description	•	Formatted Table
			1				

Part 10b - GML Data Format

20

Formatted: Normal, Space After: 0 pt, No bullets or numbering

December 2018

Encoding specification	encodingSpecification	'S-100 Part 10b'	1	CharacterStrin g	Encoding specification that defines the encoding
Encoding specification edition	encodingSpecificationE dition	"1.0"	1	CharacterStrin g	Edition of the encoding specification
Product identifier	productIdentifier		1	CharacterStrin g	Unique identifier for the data product
Product edition	productEdition		1	CharacterStrin g	Edition of the product specification
Application profile	applicationProfile		1	CharacterStrin g	"1" – base datasets"2" – update datasets
Dataset file identifier	datasetFileIdentifier		1	CharacterStrin g	The file name including the extension but excluding any path information
Dataset title	datasetTitle		1	CharacterStrin g	The title of the dataset
Dataset reference date	datasetReferenceDate		1	date	The issue date of the dataset. Format: YYYY-MM- DD
Dataset language	datasetLanguage	"EN <u>G</u> "	1	ISO 639-1	The (primary) language used in this dataset
Dataset abstract	datasetAbstract		01	CharacterStrin g	The abstract of the dataset
Dataset topic category	datasetTopicCategory	{14}{18}	1*	MD_TopicCate goryCode (ISO 19115-1)	A set of topic category codes from the MD_TopicCategoryCo de list in ISO 19115-1 (except "extraTerrestrial")
Dataset Purpose	<u>datasetPurpose</u>	<u>{"Base",</u> <u>"Update</u> <u>"}</u>	1	<u>CharacterStrin</u> 9	Whether dataset consists of updated features or all features.
Update Number	updateNumber		1	Integer	The sequential update number of this dataset



Figure 10b-11-14 – Dataset identification

10b-9.6.2 Dataset structure information

Dataset structure information is defined by the complex type DatasetStructureInformationType. The fields are shown in the table and figure below.

Table 10b-4 – Dataset structure information elements

Subfield name	XML Tag	Default Value	Mult.	Type	Description
Dataset Coordinate Origin X	datasetCoordOriginX	0.0	01	Real	Shift used to adjust x- coordinate before encoding. Set to 0.0 if no shift is used.
Dataset Coordinate Origin Y	datasetCoordOriginY	0.0	01	Real	Shift used to adjust y- coordinate before encoding. Set to 0.0 if no shift is used.

December 2018

Dataset Coordinate Origin Z	datasetCoordOriginZ	0.0	01	Real	Shift used to adjust z- coordinate before encoding.
Coordinate multiplication factor for x-coordinate	coordMultFactorX	1	01	Positive Integer	Floating point to integer multiplication factor for the x- coordinate or longitude.
Coordinate multiplication factor for y-coordinate	coordMultFactorY	4	01	Positive Integer	Floating point to integer multiplication factor for the x- coordinate or longitude.
Coordinate multiplication factor for z-coordinate	coordMultFactorZ	4	01	Positive Integer	Eloating point to integer multiplication factor for the z- coordinate or depths or height.
Number of Information Type records	nInfoRec		01	Integer ≥ 0	Number of information records in the dataset.
Number of Point records	nPointRec		01	Integer ≥ 0	Number of point records in the dataset.
Number of Multi Point records	nMultiPointRec		01	Integer ≥ 0	Number of multi point records in the dataset.
Number of Curve records	nCurveRec		01	Integer ≥0	Number of curve records in the dataset.
Number of Composite Curve records	nCompositeCurveRec		01	Integer ≥ 0	Number of composite curve records in the dataset.
Number of Surface records	nSurfaceRec		01	Integer ≥ 0	Number of surface records in the dataset.
Number of Feature Type records	nFeatureRec		01	Integer ≥ 0	Number of feature records in the dataset

December 2018

(datasetCoordOriginX)⊙			
	Shift used to adjust X coordinate before encoding			
-	datasetCoordOriginY) ⊙			
	Shift used to adjust Y coordinate before encoding			
-	datasetCoordOriginZ)⊙			
	Shift used to adjust Z coordinate before encoding			
	coordMultFactorX)⊙			
	Floating point to integer multiplication factor for X coordinate or longitude			
-				
	Floating point to integer multiplication factor for Y coordina or latitude			
-	- coordMultFactorZ)⊕			
	Floating point to integer multiplication factor for Z coordina or depths or height			
Vata Set Structure information	ninfoRec			
	Number of information records in the data set			
-	nPointRec 🕑			
	Number of point records in the data set			
-	nMultiPointRec ⊙			
	Number of multipoint records in the data set			
-	nCurveRec) 🕀			
	Number of curve records in the data set			
-	nCompositeCurveRec)			
	Number of composite curve records in the data set			
-	nSurfaceRec)			
	Number of surface records in the data set			
	nFeatureRec)			
	Number of feature records in the data set			
Figure 10b-12 <u>15</u> - Data:	set structure information			

10b-9.710b-9.8 Feature object identifier

The S-100 base schema provides a definition of feature object identifier structured similarly to the feature object identifier of S-57 Edition 3.1. <u>Encoding of these identifiers is optional for dataset instances.</u>



Figure 10b-13-16 - Structure of feature object identifier element

10b-9.810b-9.9 Coordinate Reference System

GML allows the coordinate reference system ("spatial reference system") used for geometry to be determined in different ways – by explicit specification, or by "inheriting" the SRS of outer elements. For S-100 datasets, this means the SRS can be specified in one of two ways:

- Using the srsName attribute of the gml:Envelope element in a feature collection implies that the same SRS is uses for all geometries contained in that collection.
- Using the srsName and srsDimension attributes for individual geometry elements.

Application data formats may use either method, but shall ensure that the SRS of every instance of geometry in a dataset can be determined by application software, using one method or another.

"Standard" geodetic coordinate reference systems shall be identified using the URI convention for SRS specified by OGC.

Example: http://www.opengis.net/def/crs/EPSG/0/4326

10b-9.910b-9.10 Dataset structure definition

Application schemas for data products shall <u>define_define_an XML</u> type and element to serve as the root element of a GML dataset, consisting of a collection of XML elements for feature, information type, and spatial data objects defined elsewhere in the application schema.

10b-9.9.1 Dataset metadata

The dataset class may contain one or metadata properties to encode dataset level metadata (for example ISO 19115/19139) either inline or by reference.

10b-10 Constraints and validation

Some validation of data can be done using validating XML processors if the data product's GML application schema created well defined types wherever possible, for example, enumerated types for the enumeration attributes, and maximum and minimum allowed values for real attributes. However, complete validation, especially of conditional attributes, is likely to require an additional means of data validation.

Constraints allow complex business rules to be defined that restrict the allowable values based on well-defined limits or relationships between properties (for example the end date must be equal to or greater than the start time).

Constraints can be defined in many different ways - human readable text only, object constraint language (OCL), Semantics of Business Vocabulary and Business Rules (SVBR) and these can be documented as part of the UML model or external to the model.

The S-100 GML Profile does not provide explicit support for expressing constraints or for rulebased validation. Current industry best practice, advocates the uses of Schematron to validate XML files based against business rules defined using OCL, SVBR or human readable text. Schematron (ISO/IEC 19757-3) is a rule-based validation language for making assertions about the presence or absence of patterns in XML. Constraints encoded using Schematron may be directly encoded within the resulting Application Schema or may be defined in an associated Schematron document. <u>These may not be processed by implementing systems however</u>.

Part 10b - GML Data Format

25

December 2018

10b-11 Dataset level metadata and integrity checks

The S-100 GML Profile does not explicitly contain any elements relating to dataset level metadata or integrity checks. These should be defined within an S-100 Application Schema as properties of the Feature Collection class.



Figure 10b-14 17 - Encoding dataset metadata and integrity check properties

10b-1210b-11 Schema locations and namespaces

The GML profile and GML application schema for common elements are located at the IHO GI registry web site. Namespaces and versions are also defined on that site.

10b-1310b-12 Divergences from common GML practices

The GML profile (10b-8) and base schema (10b-9) diverge from common GML practice in the following items:

- Interpretation of missing curve interpolation value (clauses <u>10b-1.140b-7.2</u>, 10b-8.5.2).
 Allowing feature and information associations to use one of two standard properties <featureAssociation> and <informationAssociation> instead of the role name as the property name. The values XLink attributes in associations are specified as described in this Part which overrides the interpretations in the XLink specification.
- 3.2. Geometry properties are defined individually instead of using substitution groups. There is no single property which functions as a spatial attribute in all features.

10b-1410b-13 Conventions for S-100 GML data formats

Data formats must use the camel case codes of features, information types, and attributes and <u>subattributes</u> as specified in feature catalogues as the 'local name' in element tags for GML features, and objects, or attributes.

EXAMPLE: Given a Feature Catalogue that defines a feature named "Marine Protected Area" with code "MarineProtectedArea" the corresponding feature in the dataset must use "MarineProtectedArea" as the local name – for example, <S122:MarineProtectedArea ... or <MarineProtectedArea

For S-100 Enumeration or S-100 Codelist attributes, datasets must use the code, label, or alias field of the listed value as encoded in the Feature Catalogue. Dataset header information must specify which is used in the *attributeEncoding* field of the header metadata. <u>If unspecified then</u> the label shall be used.

December 2018

All enumerations shall be defined as types named <enumeration code>Type where the code is identical to the code defined in the feature catalogue. Similarly all feature and information types shall be of type "<featurename>Type" or "<informationtype>"Type"

Ordering of elements in GML Schemas corresponding to simple and complex attributes shall be identical with ordering in the feature catalogue. Attributes and sub-attributes inherited from abstract types in the feature catalogue shall appear before those belonging to the specialised type.

Spatial objects that are encoded independently of features (that is, not embedded in a feature) mustshall be encoded with tags whose local name components are the spatial object elements in the S-100 GML profile (for example, S100:Point). <u>The S-100 profile defines common types</u> for features which may use more than one spatial primitive, e.g. PointOrCurve and <u>CurveOrSurface</u>.

The GML Profile defines abstract types for feature and information types to be used for definition of concrete types contained in the feature catalogue.

Feature and information associations must encode at least one of the *role* or *arcrole* attributes of the reference.

The following tags are reserved and may not be used in GML data formats as local names of elements:

- member
- imember
- geometry
- location

GML data formats for S-100 datasets must follow the GML rules as described in the GML specification (ISO 19136/OGC 07-036), as modified by the S-100 GML profile and this Part.

10b-1510b-14 Processing of GML datasets (informative)

Implementations, including applications and production tools, may use any suitable method for processing GML datasets. While GML datasets must conform to the GML application schemas defined in product specifications, processors are not required to use the GML application schemas for processing datasets. However, the combination of the GML specification, this Part, and the S-100 GML profile result in the following commonalities:

- Each dataset has a single root element (<u>"S1XX:DatasetROOTELEMENT</u>"). GML datasets are XML documents and this is an XML requirement.
- 2)1)The tags "member" and "imember" are reserved for use as wrapper tags for feature and information types. Use of these wrapper tags is optional in S-100 GML application schemas.
- 3) Given the path /ROOTELEMENT/member/X1/X2 then X1 is a feature and X2 is an attribute or association role. Similarly given /ROOTELEMENT/imember/X1/X2 X1 is an information type and X2 one of its attributes or associations.
- 4)2) If X2 has XML attributes xlink:href and xlink:role and/or xlink:arcrole it is an association role.
- 5)3) If X2 has element content it is a complex or spatial attribute.
- 6)4)A spatial attribute or object will have one of the allowed spatial properties as its content.
- 7)5)If X2 is empty and nilled, or has text or numeric content, it is a simple attribute.
- 8)6)Applications must allow for the presence or absence of namespaces, for example X1 might be of the form S122:FeatureA, etc. Namespaces in XML precede a ':' so it is possible for applications to distinguish the namespace part of the tag from the 'local name' part.

l

December 2018

S-100 Edition 4.0.0	December 2018

Page intentionally left blank

December 2018

Appendix 10b-A Application Schema (informative)

10b-A-1.10b-A-2. Example of a GML application schema for an S-100based data product

The example schema defines abstract base types for features and information types in the data product. Attributes common to all features/information types can be defined here. The abstract elements act as substitution group heads for feature and information type instances. This is convenient especially if the number of features or information type definitions is relatively high. GML also requires that features in GML application schemas be in the substitution group of *gml:AbstractFeature* and this can be achieved using the substitution groups.

<!-- base types for all features in this data product --> <xs:complexType name="AbstractFeatureType"> <xs:complexContent> <xs:extension base="s100:AbstractFeatureType"> <xs:sequence> <xs:element name="scaleMinimum" type="xs:positiveInteger"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> <xs:element name="AbstractFeature" type="AbstractFeatureType" abstract="true" substitutionGroup="gml:AbstractFeature"> <xs:annotation> <xs:documentation>Substitution group head for features</xs:documentation> </xs:annotation> </xs:element> <!-- base types for all information types in this data product --> <xs:complexType name="AbstractInformationTypeType"> <xs:complexContent> <xs:extension base="s100:AbstractInformationType"> <xs:sequence/> </r> </xs:complexContent> </xs:complexType> <xs:element name="AbstractInformationType" type="AbstractInformationTypeType" abstract="true" substitutionGroup="gml:AbstractGML"> <xs:annotation> <xs:documentation>Substitution group head for information objects</xs:documentation> </xs:annotation> </xs:element>

Figure 10b-A-1 – Base abstract types and elements in GML application schema

The figure below is a graphical representation of the model of *AbstractFeatureType*, defined above, showing the inherited feature identifier element, inherited association elements for feature and information associations, as well as the locally defined attribute *scaleMinimum*.

December 2018





Information types are derived from the base type. The example has a single information type named ChartNote. Attributes specific to the information type are defined here.

<xs:complexType name="ChartNoteType"> <xs:complexContent> <xs:extension base="AbstractInformationTypeType"> <xs:sequence> </xs:extension> </xs:extension> </xs:complexContent> </xs:complexContent>

<xs:element name="chartNote" type="ChartNoteType" substitutionGroup="AbstractInformationType"> <xs:annotation>

<xs:documentation>A chart note conveys information in plain text.</xs:documentation> </xs:annotation>

</xs:element>

Figure 10b-A-3 – Information type definition

The definition of feature classes is similar, except that they derive from AbstractFeatureType and include spatial attributes.

Since this data product requires alternative types of spatial properties for some feature classes (hat is, the *LandArea* feature can have either point or area geometry), geometry is defined as a *choice* element offering either point or surface property).

Note that the GML specification states that geometry properties shall have application-specific names which express the semantics:

Application-specific names shall be chosen for geometry properties in GML application schemas. The names of the properties should be chosen to express the semantics of the value. Using application specific names is the preferred method for names of properties including geometry properties.

There are no inherent restrictions in the type of geometry property a feature type may have as long as the property value is a geometry object substitutable for gml:AbstractGeometry.

This profile therefore allows spatial attributes be implemented using geometry substitution groups or element names with application-specific semantics. The schema includes predefined

December 2018

property types which may be used as types of geometry property elements. However GML application schema developers should keep in mind the relevant requirements for stated in ISO 19136, for example, in clause 10.1.3.1 about derivation from gml:AbstractGeometryType and substition groups for geometry.

<!-- feature definitions --> <xs:complexType name="DepthAreaType"> <xs:complexContent> <xs:extension base="AbstractFeatureType"> <xs:sequence> <xs:element name="depthValue1" type="xs:double"/> <xs:element name="depthValue2" type="xs:double"/> <xs:element ref="s100:surfaceProperty"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType> <xs:element name="depthArea" type="DepthAreaType" substitutionGroup="AbstractFeature"/> <xs:complexType name="LandAreaType"> <xs:annotation> <xs:documentation>One of the features in this data product is

LandArea</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="AbstractFeatureType"> </s:extension base="AbstractFeatureType"> </s:ext

<xs:element name="landArea" type="LandAreaType" substitutionGroup="AbstractFeature"/>

Figure 10b-A-4 – Feature class definition

The example also defines an XML type for datasets and convenience groups collecting features and information types for use in the dataset definition. Developers of GML application schemas should keep in mind the rules for GML application schemas in clause 20 of ISO 19136.

The figure below shows a graphical representation of the *DepthArea* feature defined above, showing the components inherited from the abstract feature hierarchy and locally defined thematic attributes and allowed spatial attribute. The abstract type defined above is extended by adding the thematic attributes bound to the feature class and also the appropriate spatial attribute(s). A feature class may have different types of spatial objects associated with it, this can be represented using appropriate XML <choice> constructs.

Example: A feature may have either Point or Surface geometry but not curve geometry. This is represented by an XML <choice> particle containing point or surface property types but none of the curve types.



Figure 10b-A-5 – Type hierarchy of feature class in GML application schema

December 2018

Appendix 10b-B Use of Profile in GML Application Dataset (informative)

10b-B-1. Introduction

This clause illustrates the use of the GML profile (10b-8) and base schema (10b-9) and a GML application schema (App. 10b-A) for an S-100-based data product and a GML dataset.

10b-B-2. Dataset structure in GML application schema

An example of the format of a GML dataset is shown in the figure below. This dataset defines data objects as information objects, spatial objects. or feature objects. It specifies the sequence of objects in the file as information objects first, followed by spatial objects, then features.



Figure 10b-B-1 – Example of dataset definition in GML application schema

10b-B-3. Dataset examples in XML/GML

The figure below shows a partial example dataset. Content in ${\scriptstyle \textit{italics}}$ has been omitted for brevity.

<Dataset (... namespaces and schemaLocation ...) gml:id="ds">

<gml:boundedBy>

<gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
<gml:lowerCorner>0.0 0.0</gml:lowerCorner>

S-100 Edition 4.0.0 December 2018 <gml:upperCorner>3.0 3.0</gml:upperCorner> </gml:Envelope> </gml:boundedBy> <!-- information objects -->
<chartNote gml:id="cn1"> <text>The reporting area does not include the Areas to be Avoided within the Monument.</text> </chartNote> <chartNote gml:id="cn2"> <text>Vessels shall notify the authority when leaving the reporting area to enter an Area to be Avoided.</text> </chartNote> <!-- spatial objects --> <s100:Curve gml:id="curve1">... geometry of curve1 ...</s100:Curve>
<s100:Curve gml:id="curve2">... geometry of curve2 ...</s100:Curve> <!-- curve3 references curve2 --> <s100:OrientableCurve gml:id="curve3" orientation="-"> <gml:baseCurve xlink:href="#curve2"/> </s100:OrientableCurve> <!-- this surface uses curve1 as boundary --> <s100:Surface gml:id="su1"> <gml:patches> <gml:PolygonPatch> <gml:exterior> <gml:Ring> <gml:curveMember xlink:href="#curve1"/> </gml:Ring> </gml:exterior> </gml:PolygonPatch> </gml:patches> </s100:Surface> <s100:Surface gml:id="su2">... geometry of surface su2 ...</s100:Surface> <!-- features --> <depthArea gml:id="da1"> <s100:featureObjectIdentifier> <s100:agency>JS</s100:agency> <s100:featureIdentificationNumber>123</s100:featureIdentificationNumber> <s100:featureIdentificationSubdivision>345</s100:featureIdentificationSubdivision> </s100:featureObjectIdentifier> <scaleMinimum>10000</scaleMinimum> <depthValue1>1.0</depthValue1><!-- thematic attribute --> <depthValue2>6.0</depthValue2><!-- thematic attribute --> <!-- the geometry here is given by reference to surface su1 above --> <s100:surfaceProperty xlink:href="#su1"/><!-- example of geometry reference --> </depthArea> <depthArea gml:id="da2"> <s100:featureObjectIdentifier> <s100:agency>JS</s100:agency> <s100:featureIdentificationNumber>123</s100:featureIdentificationNumber> <s100:featureIdentificationSubdivision>345</s100:featureIdentificationSubdivision> </s100:featureObjectIdentifier>

December 2018

<!-- association to the first information object above, identified by its gml:id in the xlink:href attribute --> <s100:informationAssociation gml:id="ia2" xlink:href="#cn1" xlink:arcrole="http://example.iho.int/roles/hasNote"/>

<!-- feature association to a depth area feature, identified by xlink:href --> <s100:featureAssociation gml:id="fa1" xlink:href="#da1" xlink:arcrole="http://example.iho.int/roles/rolea"/>

<scaleMinimum>10000</scaleMinimum> <depthValue1>1.0</depthValue1> <depthValue2>6.0</depthValue2>

<!-- geometry is given by reference --> <s100:surfaceProperty xlink:href="#su2"/> </depthArea>

<landArea gml:id="la1"> <scaleMinimum>10000</scaleMinimum> <objectName>Micklefirth City</objectName> <s100:pointProperty> <l-- inline geometry - directposition --> <s100:Point gml:id="pnt1"> <gml:pos>1.5 1.5</gml:pos> </s100:Point> </s100:pointProperty> </landArea>

</Dataset>

Figure 10b-B-2 – Example GML dataset

December 2018

Page intentionally left blank