Open Geospatial Consortium

Date: 2010-10-27

Reference number of this Document: **OGC 09-146r1**

Version: 1.0.0

Category: OpenGIS® Interface Standard

Editor: Peter Baumann

OGC® GML Application Schema - Coverages

Warning

This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	**OpenGIS®** Interface Standard
Document subtype:	Extension
Document stage:	Approved
Document language:	English

Contents

Page

Tables

Page

i. Preface

This document specifies a GML coverage structure extending the definition of GML 3.2.1 [07-036] in a compatible way.

The main change extending GML is the addition of one mandatory component, `rangeType`, to the `Coverage` definition of GML 3.2.1 to provide a concise description of the coverage range value definition.

This enhanced coverage type is used, for example, by the Web Coverage Service (WCS) Standard [1] version 2.0 and higher, but is independent from WCS service. This augmented coverage structure can serve a wide range of coverage application domains and service types, thereby contributing to harmonization and interoperability.

ii. Terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 06-121r9], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

iii. Submitting organizations

The following organizations have submitted this Implementation Specification to the Open Geospatial Consortium, Inc.:

- Jacobs University Bremen
- National Center for Atmospheric Research (NCAR)
- Oracle USA
- PCI Geomatics Inc.
- ERDAS, Inc.
- EOX IT Services GmbH
- Spot Image
- BAE Systems - C3I Systems
- Natural Environment Research Council (NERC)
- George Mason University

iv. Document Contributor Contact Points

Name	**Organization**
Peter Baumann	Jacobs University Bremen, rasdaman GmbH

v. Revision history

Date	Release	Author	Paragraph modified	Description
2009-11-08	0.0.1	Peter Baumann, Andrei Aiordachioaie, Jinsongdi Yu	All	Created

vi. Changes to the OpenGIS® Abstract Specification

The OpenGIS® Abstract Specification does not require any changes to accommodate the technical contents of this (part of this) document.

vii. Future Work

In collaboration with the GML Standard Working Group it is foreseen to rationalize between Application Schema and the coverage schema in GML 4.0.

Foreword

Some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium Inc. shall not be held responsible for identifying any such patent rights.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation

Introduction

Coverages represent digital geospatial information representing space/time-varying phenomena. OGC Abstract Topic 6 [OGC 07-011] – which is identical to ISO 19123 – defines an abstract model of coverages. Coverage instances may be encoded using the Geography Markup Language (GML) 3.2 [07-036], an XML grammar written in XML Schema for the description of application schemas as well as the transport and storage of geographic information.

However, the definition contained in GML 3.2.1 has turned out to not contain sufficient information to describe coverage instances in a flexible, interoperable, and harmonized manner. To remedy this, this document defines a GML Application Schema for coverages by applying the following enhancements to the GML 3.2.1 `Coverage` data type:

- A mandatory element `rangeType` has been added to carry information about the range value data structure of a `Coverage`.

- The property `coverageFunction`, which in GML 3.2.1 [OGC 07-036] is associated with every subtype of `Coverage`, is moved up into `AbstractCoverage` in the coverage type hierarchy of the standard on hand. This semantic-preserving modification does not impact instance documents.

- A metadata hook has been added which allows definition of application specific supplementary information to be transported with a coverage.

- The grid coverage types are subtypes of `AbstractCoverage` rather than being subtypes of `DiscreteCoverage` as in GML 3.2.1 [OGC 07-036].

This is a strict extension: no existing part of the GML 3.2.1 [OGC 07-036] `Coverage` is changed in its syntax, nor in its semantics.

OGC® GML Application Schema - Coverages

1 Scope

This document specifies the GML coverage structure to be used by OGC standards.

2 Conformance

Standardisation target of this document are concrete **coverage instance documents**, as generated by some service and/or consumed by some client.

This document establishes a single requirements class, *gml-coverage*, of URI http://www.opengis.net/spec/GMLCOV/1.0/req/gml-coverage with a single pertaining conformance class, *gml-coverage*, of URI http://www.opengis.net/spec/GMLCOV/1.0/conf/gml-coverage. Requirements and conformance test URIs defined in this document are relative to http://www.opengis.net/spec/GMLCOV/1.0/.

Annex A lists the conformance tests which shall be exercised on any software artefact claiming to implement this Application Schema.

3 Normative references

This *OGC GML Application Schema for Coverages* standard consists of the requirements defined in this document and an XML Schema including Schematron constraints. The complete specification is identified by OGC URI http://www.opengis.net/spec/GMLCOV/1.0, the document has OGC URI http://www.opengis.net/doc/AppSchema/GMLCOV/1.0.

The complete standard is available for download from http://www.opengeospatial.net/standards/gmlcov. Additionally, the XML Schema is posted online at http://schemas.opengis.net/gmlcov/1.0 as part of the OGC schema repository. In the event of a discrepancy between bundled and schema repository versions of the XML Schema files, the schema repository shall be considered the normative reference.

The following normative documents contain provisions (conformance classes) that, through reference in this text, constitute provisions of this specification. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

OGC 07-011 *Abstract Specification Topic 6: The Coverage Type and its Subtypes*, version 7.0 (identical to ISO 19123:2005)

Conformance classes used:

- Simple coverage interface
- Discrete coverage interface
- Thiessen polygon coverage interface
- Quadrilateral grid coverage interface

- Hexagonal grid coverage interface
- TIN coverage interface
- Segmented curve coverage interface
- Discrete coverage interchange
- Thiessen polygon coverage interchange
- Quadrilateral grid coverage interchange
- Hexagonal grid coverage interchange
- TIN coverage interchange
- Segmented curve coverage interchange

OGC 07-036, *Geography Markup Language (GML) Encoding Standard*, version 3.2.1
Conformance classes used:

- GML application schemas defining coverages
- GML documents

OGC 08-094, *OGC® SWE Common Data Model Encoding Standard*, version 2.0
Conformance classes used:

- Core
- UML models
- XML Schema

4 Terms and definitions

For the purposes of this document, the terms and definitions given in the above references apply. In addition, the following terms and definitions apply.

4.1 coverage

feature that acts as a function to return values from its range for any direct position within its spatiotemporal domain [OGC 07-011]

4.2 GML coverage

feature which is a subclass (specialization) of a `Coverage` as defined in the GML Application Schema for Coverages [OGC 09-146r1]

5 Conventions

5.1 Use of term "coverage"

The definition of "coverage" in Subclause 4.1 is the generic one provided by Abstract Topic 6 [OGC 07-011]. The term "GML coverage" is coined to denote the concrete data structure defined in the document on hand, relying on GML 3.2.1 [OGC 07-036] and SWE Common 2.0 [OGC 08-094].

For the remainder of this document, "coverage" shall be understood as a shorthand for "GML coverage" unless explicitly stated otherwise.

5.2 UML notation

All the diagrams that appear in this specification are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of OGC Web Service Common [OGC 06-121r9]. Further, the following conventions hold:

- UML elements having a package name of `GML` are those defined in the UML model of GML 3.2.1 [OGC 07-036].
- UML elements having a package name of “`SWE Common`” are those defined in the UML model of SWE Common 2.0 [OGC 08-094].
- UML elements not qualified with a package name are those defined in this Application Schema.

5.3 Namespace prefix conventions

The namespace prefixes used in this document are **not** normative and are merely chosen for convenience; they may appear in examples without being formally declared, and have no semantic significance. The namespaces to which the prefixes correspond are normative, however.

Table 1 Namespace mapping conventions

Prefix	Namespace URI	Description
gml	http://www.opengis.net/gml/3.2	GML 3.2.1
swe	http://www.opengis.net/swe/2.0	SWE Common 2.0
gmlcov	http://www.opengis.net/gmlcov/1.0	GML Application Schema for Coverages 1.0

6 Coverage Model

This Clause specifies the changes over the GML 3.2.1 coverage model and the components adopted from the SWE Common data model.

6.1 Overview

In GML 3.2.1, all coverage types are derived from the abstract `Coverage` data type. This structure contains a `domainSet` describing the coverage’s domain and a `rangeSet` component containing the range values (“pixels”, “voxels”) of the coverage. This Application Schema extends GML 3.2.1 [OGC 07-036] class `Coverage` with two components, `rangeType` and `metadata`.

- The `rangeType` element describes the coverage's range set data structure. A range value often consists of one or more fields (in remote sensing also referred to as *bands* or *channels*), however, much more general definitions are possible. Range value structure description is based on the SWE Common [OGC 08-094] `DataRecord`.

- The abstract coverage definition is augmented with an extensible slot for metadata. The intended use is to define concrete metadata structures and their semantics in extensions or application profiles.

The following changes apply over the GML 3.2.1 [OGC 07-036] specification:

- The property `coverageFunction`, which in GML 3.2.1 [OGC 07-036] is associated with every subtype of `Coverage`, is moved up into `Coverage` in the coverage type hierarchy of the standard on hand.

  NOTE This way, the coverage function is available in any subtype of `Coverage`. This serves to prepare for continuous coverages, like in the case described next.

- The grid coverage types are subtypes of `Coverage` rather than being subtypes of `DiscreteCoverage` as in GML 3.2.1 [OGC 07-036].

  NOTE This allows representing not only discrete grid coverages, but also continuous coverages by using grids for the reference points in conjunction with a coverage function defining interpolation.

No further changes over GML 3.2.1 [OGC 07-036] are made in this document. In particular, no pre-existing component changes its semantics.

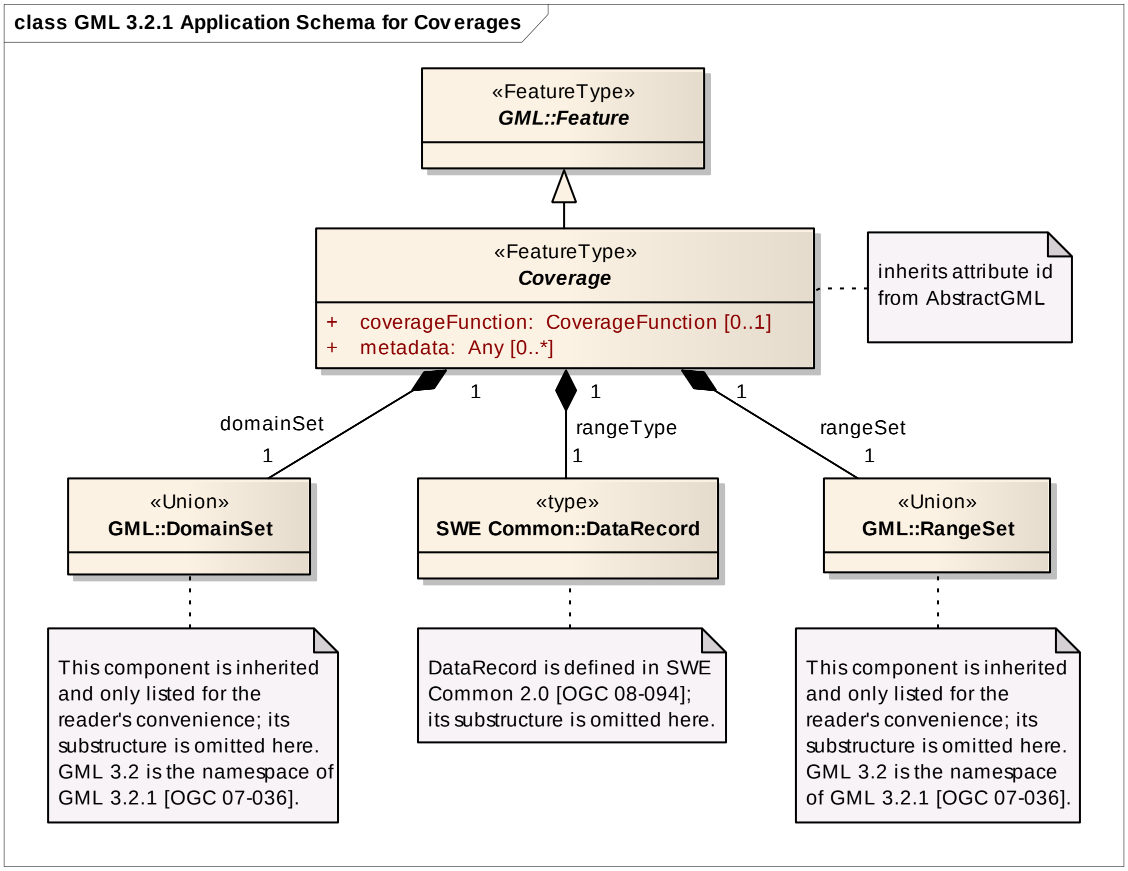
class GML 3.2.1 Application Schema for Coverages
«FeatureType»
GML::Feature
«FeatureType»
Coverage
+ coverageFunction: CoverageFunction [0..1]
+ metadata: Any [0..*]
inherits attribute id from AbstractGML
domainSet
1
1
rangeType
1
1
rangeSet
1
1
«Union»
GML::DomainSet
«type»
SWE Common::DataRecord
«Union»
GML::RangeSet
This component is inherited and only listed for the reader's convenience; its substructure is omitted here. GML 3.2 is the namespace of GML 3.2.1 [OGC 07-036].
DataRecord is defined in SWE Common 2.0 [OGC 08-094]; its substructure is omitted here.
This component is inherited and only listed for the reader's convenience; its substructure is omitted here. GML 3.2 is the namespace of GML 3.2.1 [OGC 07-036].

Figure 1: The `Coverage` structure

For the standard on hand, Coverage shall always refer to the definition of this Application Schema and *not* to the GML definition of the same name, unless explicitly stated otherwise.

Figure 1 shows the UML diagram pertaining to this Application Schema.

Requirement 1 /req/gml-coverage/structural-adherence:
Any XML document instantiating a concrete subtype of Coverage **shall** conform with the UML diagram in Figure 1, with Table 2, and with the XML schema defined as part of this standard.

Table 2 The Coverage data structure

Name	Definition	Data type	Multiplicity
coverage-Function	GML 3.2.1 coverage function to describe how range values at coverage locations can be obtained	GML:: Coverage-Function	Zero or one (optional)
metadata	Application specific metadata	Any	Zero or more (optional)
domainSet	GML 3.2.1 Definition of coverage domain	GML:: DomainSet	One (mandatory)
rangeType	Structure definition of the coverage range values	SWE:: DataRecord	One (mandatory)
rangeSet	GML 3.2.1 Coverage range values	GML:: RangeSet	One (mandatory)

NOTE 1 The optional element gml:Envelope serves to establish a bounding box of the coverage on hand. For a purely spatial coverage, gml:Envelope is appropriate. In case the coverage also has a spatial axis, gml:Envelope can be substituted by a gml:EnvelopeWithTimePeriod; in case of a purely temporal coverage, spatial dimension in gml:EnvelopeWithTimePeriod will be zero.

NOTE 2 UML data type Any is used here with the same meaning as XML's xsd:any, which does not have a direct equivalent in UML.

NOTE 3 Following the GML pattern described in [OGC 07-036], on GML level SWE:: DataRecord is linked to rangeType via an association SWE::DataRecordPropertyType.

6.2 CoverageFunction

The coverageFunction component is identical in its syntax and meaning to the coverageFunction element defined in GML [OGC 07-036] Subclause 19.3.11.

6.3 Metadata

The metaData component is a carrier for any kind of application dependent metadata. Hence, no requirements are imposed here.

6.4 RangeType

The `rangeType` component adds a structure description and technical metadata required for an appropriate (however, application independent) understanding of a coverage. For this structure description, the SWE Common `DataRecord` is used.

Requirement 2 /req/gml-coverage/dataRecord:
The range type component of a coverage **shall** conform with the `DataRecord` of SWE Common [OGC 08-094].
Dependency: [OGC 08-094] Clause 7 (http://www.opengis.net/doc/SWE/2.0/clause/7), [OGC 08-094] Clause 8 (http://www.opengis.net/doc/SWE/2.0/clause/8),

NOTE Following GML patterns the `swe:DataRecord` is linked into `gmlwcs:AbstractCoverageType` via `swe:DataRecordPropertyType`.

Atomic data types available for range values are those given by the SWE Common data type `AbstractSimpleComponent`. As a range structure contains only structure definitions, but not the values themselves (these sit in the coverage range set component), the optional `AbstractSimpleComponent` component `value` is suppressed in coverages.

Requirement 3 /req/gml-coverage/no-value-in-rangeType:
For all SWE Common `AbstractSimpleComponent` subtypes in a range type structure, instance multiplicity of the `value` component **shall** be zero.

NOTE Following [OGC 08-094], omission of the `value` component implies that in a `DataArray` there is no `encoding` component either.

Range values can be structured as records or arrays. Both structuring principles can be nested (and mixed) to any depth for a concrete coverage range structure definition.

Requirement 4 /req/gml-coverage/record-or-dataArray:
Wherever the SWE Common XML schema allows an `AbstractDataComponent` in a coverage range structure the concrete instance **shall** be one of the `AbstractDataComponent` subtypes `DataRecord` and `DataArray`.

NOTE In particular, these `AbstractDataComponent` subtypes are not allowed in range structures: `DataChoice`, `Vector`, `Matrix`.

Within a `DataRecord` contained in a concrete range structure, each of its record components is locally uniquely identified by the record component's `field` attribute, in accordance with the “soft-typing” property introduced by SWE Common.

Example The following XML fragment represents a valid range structure; it models the red, green, and blue channel of a Landsat scene. Pixels are defined as unsigned 8-bit quantities where 0 and 255 denote null values:

```
<rangeType>
  <swe:field name="red">
    <swe:Quantity definition="http://opengis.net/def/property/OGC/0/Radiance">
      <gml:description>Red Channel</gml:description>
      <gml:name>Red</gml:name>
      <swe:nilValues>
        <swe:NilValues gml:id="NIL_VALUES">
          <swe:nilValue reason="http://www.opengis.net/def/nil/OGC/0/BelowDetectionRange">
```

```
                  0
               </swe:nilValue>
               <swe:nilValue reason="http://www.opengis.net/def/nil/OGC/0/AboveDetectionRange">
                  255
               </swe:nilValue>
            </swe:NilValues>
         </swe:nilValues>
         <swe:uom code="W/cm2"/>
         <swe:constraint>
            <swe:AllowedValues gml:id="VALUE_SPACE">
               <swe:interval>0 255</swe:interval>
               <swe:significantFigures>3</swe:significantFigures>
            </swe:AllowedValues>
         </swe:constraint>
      </swe:Quantity>
   </swe:field>
   <swe:field name="green">
      <swe:Quantity definition="http://opengis.net/def/property/OGC/0/Radiance">
         <gml:description>Green Channel</gml:description>
         <gml:name>Green</gml:name>
         <swe:nilValues xlink:href="#NIL_VALUES"/>
         <swe:uom code="W/cm2"/>
         <swe:constraint xlink:href="#VALUE_SPACE"/>
      </swe:Quantity>
   </swe:field>
   <swe:field name="blue">
      <swe:Quantity definition="http://opengis.net/def/property/OGC/0/Radiance">
         <gml:description>Blue Channel</gml:description>
         <gml:name>Blue</gml:name>
         <swe:nilValues xlink:href="#NIL_VALUES"/>
         <swe:uom code="W/cm2"/>
         <swe:constraint xlink:href="#VALUE_SPACE"/>
      </swe:Quantity>
   </swe:field>
</rangeType>
```

6.5 RangeSet coherence

Both `domainSet` and `rangeType` describe the coverage values given in the `rangeSet`. Hence, consistency must be enforced between them. The pertaining requirements are listed in this Subclause.

Requirement 5 /req/gml-coverage/one-range-value-per-position:
For each coordinate position contained in the domain set description of a coverage there **shall** exist exactly one range value in the coverage's range set.

NOTE Both duplicates and values omitted are not allowed. For range values not known for some reason nil values can be used.

Requirement 6 /req/gml-coverage/range-structure-consistency:
All range values contained in the range set of a coverage **shall** be consistent with the structure description provided in its range type.

6.6 Specific coverage types

This standard supports all coverage types which GML 3.2.1 [OGC 07-036] supports, which implement the discrete coverage types defined in ISO 19123 (listed in parenthesis). The supported types are substitutable from `Coverage` and include:

- `MultiPointCoverage` (ISO 19123: `CV_DiscretePointCoverage`)
- `MultiCurveCoverage` (ISO 19123: `CV_DiscreteCurveCoverage`)
- `MultiSurfaceCoverage` (ISO 19123: `CV_DiscreteSurfaceCoverage`)
- `MultiSolidCoverage` (ISO 19123: `CV_DiscreteSolidCoverage`)
- `GridCoverage` (ISO 19123: `CV_DiscreteGridPointCoverage`)
- `RectifiedGridCoverage` (ISO 19123: `CV_DiscreteGridPointCoverage`)
- `ReferenceableGridCoverage` (added to GML via Change Request [OGC 07-112r3])

The above coverage types may be used as is, or new coverage types may be constructed by using or deriving from one of the subtypes of `Coverage` or one of its subtypes.

Requirement 7 /req/gml-coverage/coverage-derivation:
The type of the root element of a coverage document instance **shall** be a concrete direct or indirect subtype of `Coverage`.

Figure 2 shows the UML diagram of the coverage hierarchy.

NOTE As in GML, continuous coverages are not currently supported. Consequently, `ContinuousCoverage` does not have any concrete subtype.

6.6.1 DiscreteCoverage

The domain set of a discrete coverage consists of either spatial or temporal geometry objects, finite in number. The range set is comprised of a finite number of attribute values each of which is associated to every direct position within any single spatiotemporal object in the domain. In other words, the range values are constant on each spatiotemporal object in the domain. This coverage function maps each element from the coverage domain to an element in its range.

This class serves as the head of a specialization hierarchy which contains `MultiPointCoverage`, `MultiCurveCoverage`, `MultiSurfaceCoverage`, and `MultiSolidCoverage`.

NOTE In GML 3.2.1 [OGC 07-036] grid coverages are contained in this class hierarchy as well based on a distinction between discrete and continuous coverages which is not considered state of the art any more. This Application Schema changes the hierarchy in that coverages are put separately, allowing to model continuous grid coverages.

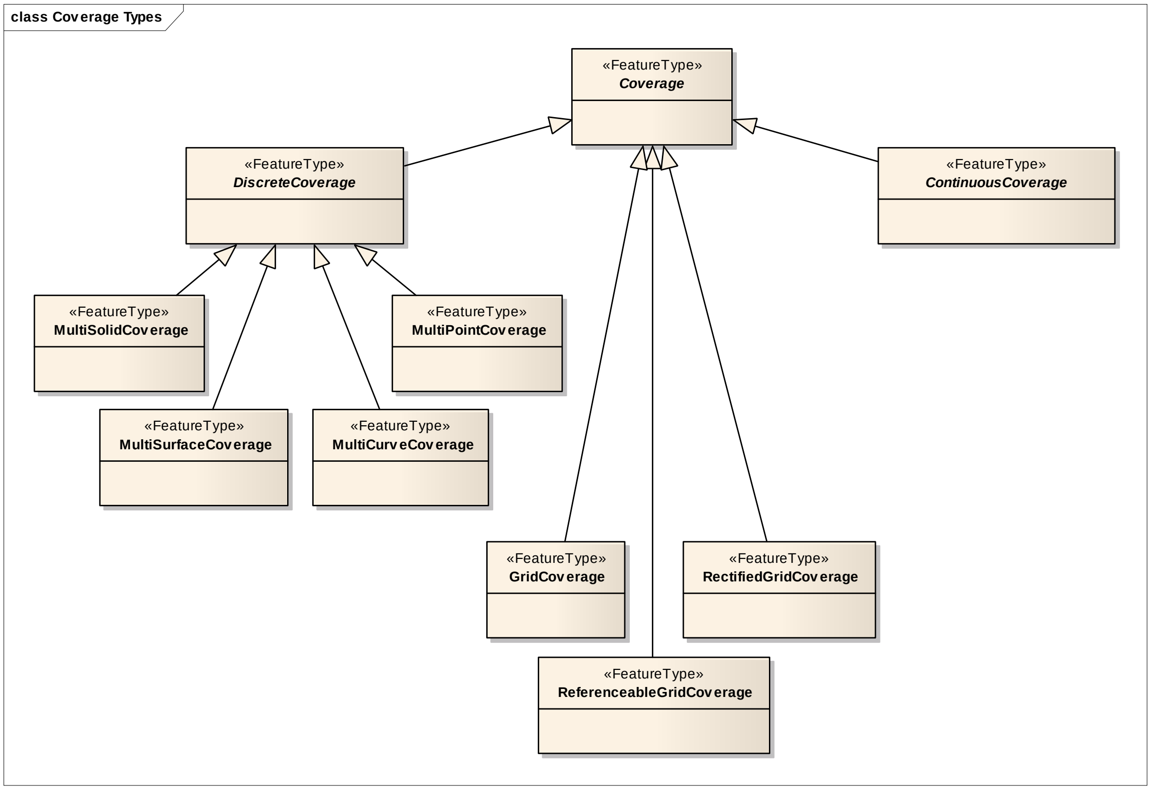

Figure 2: The `Coverage` type hierarchy

6.6.2 ContinuousCoverage

A continuous coverage as defined in ISO 19123 is a coverage that can return different values for the same feature attribute at different direct positions within a single spatiotemporal object in its spatiotemporal domain. The base type for continuous coverages is `ContinuousCoverage`.

Abstract class `ContinuousCoverage` serves as the head of a substitution group which may contain any continuous coverage whose type is derived from `ContinuousCoverage`. It parallels `GML::ContinuousCoverage`, except that the `coverageFunction` element has been moved "up" into `Coverage`.

NOTE This GML handling of continuous coverages is under reconsideration, therefore use of `ContinuousCoverage` is not encouraged. The various grid coverage types (see Subclause 6.6.7ff) allow already modelling certain types of continuous coverages.

6.6.3 MultiPointCoverage

In a `MultiPointCoverage` the domain set is a `GM_MultiPoint`, that is a collection of arbitrarily distributed geometric points.

Requirement 8 /req/gml-coverage/multiPointCoverage:
A coverage of type `MultiPointCoverage` **shall** have a content model identical with `DiscreteCoverage`, except that the `domainSet` **shall** have `GML::MultiPoint` values.

In the GML representation of a `MultiPointCoverage` the mapping from the domain to the range is straightforward:

- For `gml:DataBlock` encodings the points of the `gmlcov:MultiPoint` are mapped in document order to the tuples of the data block.
- For `gml:CompositeValue` encodings the points of the `gmlcov:MultiPoint` are mapped to the members of the composite value in document order.
- For `gml:File` encodings the points of the `gmlcov:MultiPoint` are mapped to the records of the file in sequential order.

6.6.4 MultiCurveCoverage

In a `MultiCurveCoverage` the domain is partitioned into a collection of curves comprising a `GM_MultiCurve`. The coverage function then maps each curve in the collection to a value in the range set.

Requirement 9 /req/gml-coverage/multiCurveCoverage:
A coverage of type `MultiCurveCoverage` **shall** have a content model identical with `DiscreteCoverage`, except that the `domainSet` **shall** have `“GML 3.2”::MultiCurve` values.

In the GML representation of a `MultiCurveCoverage` the mapping from the domain to the range is straightforward:

- For `gml:DataBlock` encodings the curves of the `gmlcov:MultiCurve` are mapped in document order to the tuples of the data block.
- For `gml:CompositeValue` encodings the curves of the `gmlcov:MultiCurve` are mapped to the members of the composite value in document order.
- For `gml:File` encodings the curves of the `gmlcov:MultiCurve` are mapped to the records of the file in sequential order.

6.6.5 MultiSurfaceCoverage

In a `MultiSurfaceCoverage` the domain is partitioned into a collection of surfaces comprising a `GM_MultiSurface`. The coverage function maps each surface in the collection to a value in the range set.

Requirement 10 /req/gml-coverage/multiSurfaceCoverage:
A coverage of type `MultiSurfaceCoverage` **shall** have a content model identical with `DiscreteCoverage`, except that the `domainSet` **shall** have `GML::MultiSurface` values.

In the GML representation of a `MultiSurfaceCoverage` the mapping from the domain to the range is straightforward:

- For `gml:DataBlock` encodings the surfaces of the `gmlcov:MultiSurface` are mapped in document order to the tuples of the data block.

- For gml:CompositeValue encodings the surfaces of the gmlcov:MultiSurface are mapped to the members of the composite value in document order.

- For gml:File encodings the surfaces of the gmlcov:MultiSurface are mapped to the records of the file in sequential order.

6.6.6 MultiSolidCoverage

In a MultiSolidCoverage the domain is partitioned into a collection of solids comprising a GM_MultiSolid. The coverage function then maps each solid in the collection to a value in the range set.

Requirement 11 /req/gml-coverage/multiSolidCoverage:
A coverage of type MultiSolidCoverage **shall** have a content model identical with DiscreteCoverage, except that the domainSet **shall** have GML::MultiSolid values.

In the GML representation of a MultiSolidCoverage the mapping from the domain to the range is straightforward:

- For gml:DataBlock encodings the solids of the gmlcov:MultiSolid are mapped in document order to the tuples of the data block.

- For gml:CompositeValue encodings the solids of the gmlcov:MultiSolid are mapped to the members of the composite value in document order.

- For gml:File encodings the solids of the gmlcov:MultiSolid are mapped to the records of the file in sequential order.

6.6.7 GridCoverage

A GridCoverage is a discrete point coverage in which the domain is a geometric grid of points encoded using gml:Grid (not its subtypes gml:RectifiedGrid or a subtype of AbstractReferenceableGrid). Note that this is similar to the MultiPointCoverage except that a gml:Grid shall be used to describe the domain.

Requirement 12 /req/gml-coverage/gridCoverage:
A coverage of type GridCoverage shall have a domain that is a GML::Grid.

NOTE Such geometric positioning is introduced in the RectifiedGridCoverage.

In order to address ambiguities in the gml:Grid definition, this GML Application Schema for Coverages imposes additional constraints on the use of a gml:Grid within a gmlcov:GridCoverage. (Specifically, there is no provision in the definition of gml:Grid definition to express the relationship between the grid positions and this geometry's coordinate reference system, which will always exist in some contexts, such as a Web Coverage Service. This coordinate reference system will be explicitly referenced in the srsName attribute of the gml: SRSReferenceGroup of gml:Grid, or be inherited from an enclosing container element, such as the gml:Envelope of this gmlcov:GridCoverage.) Since provision for expressing a relationship does not exist, whenever used in gmlcov:GridCoverage, the relationship shall be simple. In this simple relationship, the dimension attribute of the gml:Grid shall be identical to the dimension of the geometry's coordinate system, the axes of the gml:Grid shall be identical to the axes of the geometry's coordinate system (which requires that the axisLabels be identical to those in the coordinate system definition), and the limits shall be treated as being expressed as coordinates in the geometry's coordinate reference system.

Clearly these additional constraints are quite limiting, in that gridded datasets whose Reference points happen to exist exactly at integral coordinates of a spatial coordinate system at a spacing of exactly one in all coordinate dimensions are exceedingly rare, unless that coordinate system is part of a gml:ImageCRS. Nevertheless, the gmlcov:GridCoverage is available for such purposes.

It is recommended that the more sensible provisions of the gmlcov:RectifiedGridCoverage or gmlcov:ReferenceableGridCoverage be utilized for all gridded datasets, since their domains can accommodate the simple provisions of the gmlcov:GridCoverage as well as more complex referencing situations.

Since this GridCoverage uses Coverage, it can be used for both discrete and continuous coverages.

6.6.8 RectifiedGridCoverage

A RectifiedGridCoverage is a discrete point coverage based on a rectified grid. It is similar to the grid coverage except that the points of the grid are geometrically referenced.

Requirement 13 /req/gml-coverage/rectifiedGridCoverage:
A coverage of type RectifiedGridCoverage **shall** have a domain that is a GML::RectifiedGrid geometry.

Since this RectifiedGridCoverage uses Coverage, it can be used for both discrete and continuous coverages.

6.6.9 ReferenceableGridCoverage

A ReferenceableGridCoverage is an implementation of ISO 19123 DiscreteGridPointCoverage for a ReferenceableGrid domain.

Requirement 14 /req/gml-coverage/referenceableGridCoverage:
A coverage of type ReferenceableGridCoverage **shall** have a domain geometry that is a subtype of GML::ReferenceableGrid.

Since this ReferenceableGridCoverage uses the gmlcov:AbstractCoverageType, it can be used for both discrete and continuous coverages.

NOTE The equivalent of this element has been added to GML 3.2.1 by approved Change Request 07-112r3 and, therefore, has been added to this standard as well.

6.7 Complete coverage example

Example The following is a complete RectifiedGridCoverage instance:

```
<?xml version="1.0" encoding="UTF-8" ?>
<gmlcov:RectifiedGridCoverage
```

```
xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gmlcov=”http://www.opengis.net/gmlcov/1.0”
xmlns:gml=”http://www.opengis.net/gml/3.2”
xsi:schemaLocation=
   ”http://www.opengis.net/gmlcov/1.0 http://schemas.opengis.net/gmlcov/1.0/gmlcovAll.xsd”
gml:id="C001">
<gml:boundedBy>
   <gml:Envelope srsName="http://www.opengis.net/def/crs/EPSG/0/4326" axisLabels="Lat Long"
      uomLabels="deg deg" srsDimension="2">
      <gml:lowerCorner>1 1</gml:lowerCorner>
      <gml:upperCorner>3 3</gml:upperCorner>
   </gml:Envelope>
</gml:boundedBy>
<gml:domainSet>
   <gml:RectifiedGrid gml:id="RG001_C001"
      srsName="http://www.opengis.net/def/crs/EPSG/0/4326" axisLabels="Lat Long"
     uomLabels="deg deg" dimension="2">
      <gml:limits>
         <gml:GridEnvelope>
            <gml:low>0 0</gml:low>
            <gml:high>9999 9999</gml:high>
         </gml:GridEnvelope>
      </gml:limits>
      <gml:axisLabels>Lat Long</gml:axisLabels>
      <gml:origin>
         <gml:Point gml:id="P001_C001" srsName="http://www.opengis.net/def/crs/EPSG/0/4326">
            <gml:pos>99. 99.9</gml:pos>
         </gml:Point>
      </gml:origin>
      <gml:offsetVector>1 0</gml:offsetVector>
      <gml:offsetVector>0 1</gml:offsetVector>
   </gml:RectifiedGrid>
</gml:domainSet>
<rangeType>
   <swe:field name="white">
     <swe:Quantity definition="http://opengis.net/def/property/OGC/0/Radiance">
        <gml:description>Panchromatic</gml:description>
        <gml:name>White</gml:name>
        <swe:nilValues>
           <swe:nilValue reason="http://www.opengis.net/def/nil/OGC/0/BelowDetectionRange">
              0
           </swe:nilValue>
           <swe:nilValue reason="http://www.opengis.net/def/nil/OGC/0/AboveDetectionRange">
              255
           </swe:nilValue>
        </swe:nilValues>
        <swe:uom code="W/cm2"/>
        <swe:constraint>
           <swe:AllowedValues>
              <swe:interval>0 255</swe:interval>
              <swe:significantFigures>3</swe:significantFigures>
           </swe:AllowedValues>
        </swe:constraint>
     </swe:Quantity>
   </swe:field>
</rangeType>
```

```
   <gml:coverageFunction>
      <gml:GridFunction>
         <gml:sequenceRule axisOrder="+1 +2">Linear</gml:sequenceRule>
         <gml:startPoint>0 0</gml:startPoint>
      </gml:GridFunction>
   </gml:coverageFunction>
   <gml:rangeSet>
      <DataBlock>
         <rangeParameters/>
         <tupleList>
            1  2  3  4  5
            6  7  8  9  10
            11 12 13 14 15
         </tupleList>
      </DataBlock>
   </gml:rangeSet>
</gmlcov:RectifiedGridCoverage>
```

Bibliography

[1] OGC 09-110r3, *Web Coverage Service (WCS) Core Interface Standard*, version 2.0, 2010

[2] W3C Recommendation, *XML Path Language (XPath)*, version 2.0, 2007 (www.w3.org/xpath20)

[3] ISO/IEC 19757-3:2006 Information technology – Document Schema Definition Languages (DSDL) – Part 3: Rule-based validation – Schematron

Annex A
(normative)

Abstract test suite

This Annex specifies an Abstract Test Suite which shall be passed in completeness by any implementation claiming conformance with this Application Schema.

A.1 Conformance Test Class: gml-coverage

The OGC URI identifier of this conformance class is:
http://www.opengis.net/spec/GMLCOV/1.0/conf/gml-coverage.
Tests identifiers below are relative to http://www.opengis.net/spec/GMLCOV/1.0/.

A.1.1 Document validates

Test id: **/conf/gml-coverage/structural-adherence**

Test Purpose: **Requirement /req/gml-coverage/structural-adherence:**
Any XML document instantiating a concrete subtype of `Coverage` `shall` **conform** with the UML diagram in Figure 1, with Table 2, and with the XML schema defined as part of this standard.

Test method: Load document into an XML validator. Test passes if coverage instance document is a valid concrete subtype of `gmlcov:AbstractCoverage-Type.`

A.1.2 DataRecord range structure

Test id: **/conf/gml-coverage/dataRecord**

Test Purpose: **Requirement /req/gml-coverage/dataRecord:**
The range type component of a coverage **shall** conform with the `DataRecord` of SWE Common [OGC 08-094].

Test method: Validate XML structure of given coverage instance against SWE Common by evaluating its conformance test suite. Test passes if all applicable SWE Common tests pass.

A.1.3 No value component in rangeType

Test id: **/conf/gml-coverage/no-value-in-rangeType**

Test Purpose: **Requirement /req/gml-coverage/no-value-in-rangeType:**
For all SWE Common `AbstractSimpleComponent` subtypes in a range type structure, instance multiplicity of the `value` component **shall** be zero.

Test method: Inspect `swe:DataRecord` element in the given instance document and check that no value component is present.

One way of doing so is to evaluate this Schematron rule:

```
<sch:rule context="//swe:Quantity |
//swe:QuantityRange |//swe:Count|//swe:CountRange |
//swe:Time | //swe:TimeRange | //swe:Boolean |
//swe:Category | //swe:CategoryRange | //swe:Text">
<sch:assert test="count(//swe:value)=0"/>
</sch:rule>
```

Test passes if constraint holds.

A.1.4 Admissible `DataRecord` subtypes

Test id: **/conf/gml-coverage/record-or-dataArray**

Test Purpose: **Requirement /req/gml-coverage/record-or-dataArray:**
Wherever the SWE Common XML schema allows an `AbstractDataComponent` in a coverage range structure the concrete instance **shall** be one of the `AbstractDataComponent` subtypes `DataRecord` and `DataArray`.

Test method: Inspect the given instance document and check the above constraint. One way of doing so is to evaluate this Schematron rule:

```
<sch:rule>
<sch:assert test="descendant-or-self::*
    [name()='swe:DataRecord' or
     name()='swe:DataArray']"
/>
</sch:rule>
```

Test passes if constraint holds.

A.1.5 Exactly one range value

Test id: **/conf/gml-coverage/one-range-value-per-position**

Test Purpose: **Requirement** Both `domainSet` and `rangeType` describe the coverage values given in the `rangeSet.` Hence, consistency must be enforced between them. The pertaining requirements are listed in this Subclause.

/req/gml-coverage/one-range-value-per-position:
`For` each `coordinate` position contained in the domain set description of a coverage there **shall** exist exactly one range value in the coverage’s range set.

Test method: Inspect the given instance document and check, for each possible cell location as defined in the coverage's domain set, that there is exactly one corresponding value in the range set.

A.1.6 Range values adhere to range structure definition

Test id: **/conf/gml-coverage/range-structure-consistency**

Test Purpose: **Requirement /req/gml-coverage/range-structure-consistency:** All range values contained in the range set of a coverage **shall** be consistent with the structure description provided in its range type.

Test method: Inspect the given instance document and check, for each range value tuple:

- Number of tuple components adheres to range structure definition.
- Data type of each atomic value conforms to the corresponding data type specification in the range structure definition.
- Value of attribute `uom` in `gml:QuantityList` in `gml:rangeSet` consistent with value of attributes `code` in `swe:uom` and `definition` in `swe:Quantity` in `gmlcov:rangeType`.

Test passes if constraints evaluate to true.

A.1.7 Coverage type correctly derived

Test id: **/conf/gml-coverage/coverage-derivation**

Test Purpose: **Requirement /req/gml-coverage/coverage-derivation:** The type of the root element of a coverage document instance **shall** be a concrete direct or indirect subtype of `Coverage.`

Test method: Check whether the XML type of the root element of the given instance document

- Is not abstract
- Is a direct or indirect subtype of `gmlcov:AbstractCoverage`.

Test passes if constraints evaluate to true.

A.1.8 Correct structure of multi-point coverage

Test id: **/conf/gml-coverage/multiPointCoverage**

Test Purpose: **Requirement /req/gml-coverage/multiPointCoverage:**
A coverage of type `MultiPointCoverage` **shall** have a content model identical with `DiscreteCoverage`, except that the `domainSet` **shall** have `GML::MultiPoint`

Test method: Check the XML type of the root element of the given instance document.

- If type is `MultiPointCoverage`: check whether the document's `domainSet` element contains values of type `gml:MultiPoint.`
- otherwise: pass test.

Test passes if constraints evaluate to true.

A.1.9 Correct structure of multi-curve coverage

Test id: **/conf/gml-coverage/multiCurveCoverage**

Test Purpose: **Requirement /req/gml-coverage/multiCurveCoverage:**
A coverage of type `MultiCurveCoverage shall` **have** a content model identical with DiscreteCoverage, `except that` the domainSet shall `have` **"GML** 3.2"::MultiCurve values.

Test method: Check the XML type of the root element of the given instance document.

- If type is `MultiCurveCoverage`: check whether the document's `domainSet` element contains values of type `gml:MultiCurve.`
- otherwise: pass test.

Test passes if constraints evaluate to true.

A.1.10 Correct structure of multi-surface coverage

Test id: **/conf/gml-coverage/multiSurfaceCoverage**

Test Purpose: **Requirement /req/gml-coverage/multiSurfaceCoverage:**
A coverage of type `MultiSurfaceCoverage` **shall** have a content model identical with `DiscreteCoverage`, except that the `domainSet` **shall** have `GML::MultiSurface` values.

Test method: Check the XML type of the root element of the given instance document.

- If type is `MultiSurfaceCoverage`: check whether the document's `domainSet` element contains values of type `gml:MultiSurface.`

- otherwise: pass test.

Test passes if constraints evaluate to true.

A.1.11 Correct structure of multi-solid coverage

Test id: **/conf/gml-coverage/multiSolidCoverage**

Test Purpose: **Requirement /req/gml-coverage/multiSolidCoverage:**
A coverage of type `MultiSolidCoverage` **shall** have a content model identical with `DiscreteCoverage`, except that the `domainSet` **shall** have `GML::MultiSolid` values.

Test method: Check the XML type of the root element of the given instance document.

- If type is `MultiSolidCoverage`: check whether the document's `domainSet` element contains values of type `gml:MultiSolid.`

- otherwise: pass test.

Test passes if constraints evaluate to true.

A.1.12 Correct structure of grid coverage

Test id: **/conf/gml-coverage/gridCoverage**

Test Purpose: **Requirement** A `GridCoverage` is a discrete point coverage in which the domain is a geometric grid of points encoded using `gml:Grid` (not its sub-types `gml:RectifiedGrid` or a subtype of `AbstractReferenceableGrid`). Note that this is similar to the `Multi-PointCoverage` except that a `gml:Grid` shall be used to describe the domain.

/req/gml-coverage/gridCoverage:
A coverage of type `GridCoverage` shall have a domain that is a `GML::Grid`.

Test method: Check the XML type of the root element of the given instance document.

- If type is `GridCoverage`: check whether the document's `domainSet` element is a `gml:Grid.`

- otherwise: pass test.

Test passes if constraints evaluate to true.

A.1.13 Correct structure of rectified grid coverage

Test id: **/conf/gml-coverage/rectifiedGridCoverage**

Test Purpose: **Requirement /req/gml-coverage/rectifiedGridCoverage:**
A coverage of type `RectifiedGridCoverage` **shall** have a domain that is a `GML::RectifiedGrid` geometry.

Test method: Check the XML type of the root element of the given instance document.

- If type is `RectifiedGridCoverage`: check whether the document's `domainSet` element is a `gml:RectifiedGrid.`
- otherwise: pass test.

Test passes if constraints evaluate to true.

A.1.14 Correct structure of referenceable grid coverage

Test id: **/conf/gml-coverage/referenceableGridCoverage**

Test Purpose: **Requirement /req/gml-coverage/referenceableGridCoverage:**
A coverage of type `ReferenceableGridCoverage` **shall** have a domain geometry that is a subtype of `GML::ReferenceableGrid`.

Test method: Check the XML type of the root element of the given instance document.

- If type is `ReferenceableGridCoverage`: check whether the document's `domainSet` element is in the substitution group of `gml:AbstractReferenceableGrid.`
- otherwise: pass test.

Test passes if constraints evaluate to true.

-- end of ATS --