

Open Geospatial Consortium Inc.

Date: 05-July-2005

Reference number of this OGC™ document: OGC 05-033r9

Version: 0.0.19

Category: OGC™ Implementation Specification Profile

Editor: Panagiotis (Peter) A. Vretanos

GML simple features profile

Copyright © Open Geospatial Consortium (2005)

Warning

This document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

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: | OGC™ Implementation Specification |
| Document subtype: | Profile |
| Document stage: | Proposed version 0.0 |
| Document language: | English |

| Contents | Page |
|---|-------------|
| i. Preface..... | v |
| ii. Submitting organizations | v |
| iii. Document contributor contact points..... | vi |
| iv. Revision history | vi |
| v. Changes to the OGC™ Abstract Specification..... | viii |
| vi. Changes to the OGC™ Implementation Specifications | viii |
| Geography Markup Language (GML) Implementation Specification | ix |
| vii. Future Work Items | x |
| Foreword..... | xi |
| Introduction..... | xii |
| 1 Scope..... | 1 |
| 2 Conformance..... | 1 |
| 3 Normative references | 2 |
| 4 Terms and definitions | 2 |
| 5 Conventions | 2 |
| 5.1 Abbreviated terms | 2 |
| 5.2 Document terms and definitions..... | 2 |
| 6 Usage Scenarios | 3 |
| 6.1 Simple desktop or browser based map viewer | 3 |
| 6.2 Get a collection of features using spatial and non-spatial constraints..... | 3 |
| 6.3 "Value-add" editor (edit geometry and other attribute values) | 3 |
| 6.4 Create feature instances of specified type or id..... | 3 |
| 6.5 Update features with modified geometry and/or attribute values | 3 |
| 7 Overview..... | 3 |
| 7.1 Simple feature structure..... | 3 |
| 7.2 Supported geometries | 4 |
| 7.3 CRS support | 5 |
| 7.3.1 Introduction..... | 5 |
| 7.3.2 Homogenous CRS within geometry instances..... | 5 |
| 7.3.3 Scoping rules for srsName..... | 5 |
| 7.4 Metadata handling | 6 |
| 7.5 Conformance Levels..... | 6 |
| 8 A rigid coding pattern for GML application schemas | 7 |
| 8.1 Introduction | 7 |

| | | |
|----------|--|----|
| 8.2 | Root element..... | 7 |
| 8.3 | Importing the GML schemas..... | 8 |
| 8.3.1 | Importing and including other schemas..... | 8 |
| 8.4 | Coding pattern for feature types..... | 9 |
| 8.4.1 | Introduction..... | 9 |
| 8.4.2 | Defining feature types..... | 9 |
| 8.4.3 | Basic data types..... | 10 |
| 8.4.3.1 | Null values..... | 10 |
| 8.4.3.2 | Common facets..... | 11 |
| 8.4.3.3 | Defining elements with integer content..... | 12 |
| 8.4.3.4 | Defining elements with real content..... | 12 |
| 8.4.3.5 | Defining elements with character content..... | 13 |
| 8.4.3.6 | Defining elements with date content..... | 14 |
| 8.4.3.7 | Defining elements with Boolean content..... | 14 |
| 8.4.3.8 | Defining elements with binary content..... | 14 |
| 8.4.3.9 | Defining elements with geometric content..... | 15 |
| 8.4.3.10 | Defining elements that reference other resources..... | 16 |
| 8.4.3.11 | Elements with a code list..... | 16 |
| 8.4.3.12 | Elements with content that is a measurement..... | 17 |
| 8.5 | Comments and annotations..... | 17 |
| Annex A | (normative) Conformance testing..... | 19 |
| A.1 | Introduction..... | 19 |
| A.2 | Elements to ignore..... | 19 |
| A.3 | General conformance rules..... | 19 |
| A.4 | Root element..... | 20 |
| A.5 | Importing the profile schemas..... | 21 |
| A.6 | Importing and including other schemas..... | 21 |
| A.7 | Feature types..... | 22 |
| A.7.1 | Global element..... | 22 |
| A.7.2 | Complex type..... | 22 |
| A.8 | Properties..... | 23 |
| A.8.1 | Integer valued properties..... | 23 |
| A.8.2 | Real valued properties..... | 24 |
| A.8.4 | Character valued properties..... | 24 |
| A.8.5 | Character valued properties with a dynamic list of values..... | 25 |
| A.8.6 | Date valued properties..... | 27 |
| A.8.7 | Boolean valued properties..... | 28 |
| A.8.8 | Binary valued properties..... | 28 |
| A.8.9 | Geometry valued properties..... | 29 |
| A.8.10 | Properties with references to other resources..... | 30 |
| A.8.12 | Measurements..... | 31 |
| Annex B | (Informative) Mapping to SQL/MM Geometric Types..... | 33 |
| B.1 | SQL/MM geometric property type mapping..... | 33 |
| Annex C | (Informative) Examples..... | 34 |
| C.1 | News item example..... | 34 |

| | | |
|---------------------|--|----|
| C.2 | Roads_bts example from the CIPI1.2 testbed | 36 |
| Annex D (Normative) | Reference schema for GML simple features profile | 39 |
| Bibliography | | 49 |

i. Preface

The Open Geospatial Consortium (OGC) is an international industry consortium of more than 270 companies, government agencies, and universities participating in a consensus process to develop publicly available geo-processing specifications.

This profile is a product of OGC's Interoperability Program: a global, collaborative, hands-on engineering and testing program designed to deliver prototype technologies and proven candidate specifications into the OGC's Specification Development Program. In OGC Interoperability Initiatives, international teams of technology providers work together to solve specific geo-processing interoperability problems posed by Initiative.

ii. Submitting organizations

This profile is being submitted to the OGC by the following organizations:

Cadcorp Ltd.

Sterling Court, Norton Road,
Hertfordshire, SG1 2JY
U.K.

CubeWerx Inc.

15 Rue Gamelin, Suite 506
Gatineau, QC, J8Y 6N5
CANADA

ESRI

380 New York Street
Redlands, CA 92373-8100
U.S.A.

Galdos Systems Inc.

#1300 - 409 Granville St.
Vancouver, BC, V6C 1T2
CANADA

interactive instruments GmbH

Trierer Straße 70-72
53115 Bonn
Germany

PCI Geomatics

480 St. Joseph Blvd., Suite 400
 Gatineau, PQ, J8Y 3Y7
 CANADA

iii. Document contributor contact points

All questions regarding this submission should be directed to the editor or the submitters:

Panagiotis (Peter) A. Vretanos (Editor)
 CubeWerx Inc.

Aleksander Milanovic (co-Author)
 formerly of Galdos Inc.

Contributors:

| Contact | Company |
|-------------------|-----------------------------|
| Bruce, Craig | CubeWerx Inc. |
| Cox, Simon | CSIRO |
| Keighan, Edric | CubeWerx Inc. |
| Paul Daisey | Census |
| Arctur, David | ESRI |
| Burggraf, David | Galdos Inc. |
| Martin Kyle | Galdos Inc. |
| Ron Lake | Galdos Inc. |
| Daly, Martin | Cadcorp Ltd. |
| Danko, David | ESRI |
| Lemieux, Sylvain | PCI Ltd. |
| Portele, Clemens | Interactive Instrument GmbH |
| Sonnet, Jerome | IONIC |
| Whiteside, Arliss | BAE SYSTEMS NSS |

iv. Revision history

| Date | Release | Description |
|-------------|---------|-----------------------------------|
| 07-JUL-2005 | 0.0.19 | Editorial comment from Carl Reed. |

| Date | Release | Description |
|-------------|---------|---|
| 16-JUN-2005 | 0.0.18 | Review comments from SF-GML ad-hoc teleconference held on 16JUN2005. |
| 13-JUN-2005 | 0.0.17 | Review comments from St Johns TC. Rename to GML Simple Features Profile.gmlSimpleFeaturesProfile.xsd |
| 23-APR-2005 | 0.0.16 | Review comments from David Burggraf. Added profile schema gmlSimpleFeaturesProfile.xsd. |
| 16-APR-2005 | 0.0.15 | Additional comments from David Arctur and Keith Ryden; first step to attempt to harmonize this profile with SQL/MM (since there is a desire to harmonize SF-SQL with SQL/MM). |
| 08-APR-2005 | 0.0.15 | Incorporate final comments from Frescatti TC. Add conformance rules for including schemas. |
| 06-APR-2005 | 0.0.14 | Incorporate comment from second meeting of WFS WG at Frescatti TC. |
| 05-APR-2005 | 0.0.13 | Incorporate comments from WFS WG at Frescatti TC. |
| 17-MAR-2005 | 0.0.12 | Incorporate review comments from Arliss Whiteside Renumber document from 03-003r11 to 05-033. |
| 13-MAR-2005 | 0.0.11 | Incorporate review comments from Clemens Portele and David Arctur Change name from “Level 0 Profile of GML for WFS” to “Profile of GML for Simple Features”. |
| 23-DEC-2003 | 0.0.11 | Incorporate Simon Cox comment about using lowerCamelCase for property names and UpperCamelCase for feature names. |
| 30-OCT-2003 | 0.0.10 | Integrate John Davidson final comments (very minor changes) |
| 26-SEP-2003 | 0.0.9 | Integrate final Galdos comments. Remove previous review comment ANNEX’s. |
| 25-SEP-2003 | 0.0.8 | Final scrub Indicate that clients should be prepared to deal with properties in any order. Include ROAD_BTS example in the examples clause Address P.Daisy comments. Indicate that only GML geometry elements are valid. That is that a compliant schema cannot use deprecated GML2 elements. Describe the CubeWerx schema validator in ANNEX A – the conformance annex. Add a future work annex. |
| 28-JUL-2003 | 0.0.7 | Add conformance testing annex Make minOccurs and maxOccurs attributes optional Moved clause v. (changes to implementation specifications) to clause 6.1.2. Editorial and content changes to section 6.1.2. |
| 10-JUN-2003 | 0.0.6 | Integrate round 2 comments from Galdos Integrate comments from John Davidson |
| 22-May-2003 | 0.0.5 | Migrate document to support GML3 Incorporate comments from Galdos |
| 29-Mar-2003 | 0.0.4 | add requirements section based on JohnD email put Galdos comments into Appendix comment on Galdos comments make changes based on Galdos comments |

| Date | Release | Description |
|-------------|---------|---|
| 22-Mar-2003 | 0.0.4 | Remove references to Option 1 and Option 3. |
| 11-Feb-2003 | 0.0.3 | Add normative references |
| 02-Feb-2003 | 0.0.2 | E.K. review comments |
| 17-Jan-2003 | 0.0.1 | Initial version |

v. Changes to the OGC™ Abstract Specification

None.

vi. Changes to the OGC™ Implementation Specifications

Table 1 – Additional values for inputFormat and outputFormat parameters

| Request | AttributeName | MIME Type | Description |
|----------------------|---------------|--------------------------------------|--|
| DescribeFeatureType | outputFormat | text/xml; subType=gml/3.1.1/sfgml | The WFS should respond by generating a GML application schema that conforms to this profile specification |
| GetFeature | outputFormat | text/xml; subType=gml/3.1.1/sfgml | The WFS should respond by generating an instance document that validates against a schema document that conforms to this profile |
| Transaction (Insert) | inputFormat | text/xml; subType=gml/3.1.1/sfgml | The feature instance in the insert action validates against a schema that conforms to this profile. |

Examples

The following examples show fragments from the capabilities document of a WFS that can generate schemas that comply with this specification profile and sample DescribeFeatureType requests indicating that a GML-SF compliant application schemas be generated.

Capabilities XML fragment:

The following XML fragment shows how a WFS may advertise in its capabilities document that it can generate schemas that comply with this profile:

```

1.  <ows:Operation name="DescribeFeatureType">
2.    <ows:DCP>
3.      <ows:HTTP>
4.        <ows:Post xlink:href="http://www.BlueOx.org/wfs"/>
5.      </ows:HTTP>
6.    </ows:DCP>
7.    <ows:Parameter name="outputFormat">
8.      <ows:Value>text/xml; subType=gml/3.1.1</ows:Value>
9.      <ows:Value>text/xml; subType=gml/3.1.1/sfgml</ows:Value>
10.   </ows:Parameter>
11. </ows:Operation>

```

Line 9 indicates that this WFS can generate a GML application schema that conforms to this profile.

DescribeFeatureType XML-encoded request:

The following example XML fragment shows a DescribeFeatureType request indicating that the response should conform to this profile:

```
<DescribeFeatureType outputFormat="text/xml; subType=gml/3.1.1/sfgml"/>
```

DescribeFeatureType KVP-encode request:

The following example URL shows a DescribeFeatureType request indicating that the response should conform to this profile:

<http://www.BlueOx.org/wfs/wfs.cgi?service=WFS&version=1.1.0&request=DescribeFeatureType&outputFormat=text/xml;+subType=gml/3.1.1/sfgml>

Geography Markup Language (GML) Implementation Specification

The following is taken from the latest XML Schema Part 1: Structures Second Edition (end of section 4):

Schema Representation Constraint: Schema Document Location Strategy

Given a namespace name (or none) and (optionally) a URI reference from `xsi:schemaLocation` or `xsi:noNamespaceSchemaLocation`, schema-aware processors may implement any combination of the following strategies, in any order:

- 1) *Do nothing, for instance because a schema containing components for the given namespace name is already known to be available, or because it is known in advance that no efforts to locate schema documents will be successful (for example in embedded systems);*
- 2) *Based on the location URI, identify an existing schema document, either as a resource which is an XML document or a `<schema>` element information item, in some local schema repository;*

- 3) *Based on the namespace name, identify an existing schema document, either as a resource which is an XML document or a <schema> element information item, in some local schema repository;*
- 4) *Attempt to resolve the location URI, to locate a resource on the web which is or contains or references a <schema> element;*
- 5) *Attempt to resolve the namespace name to locate such a resource.*
- 6) *Whenever possible configuration and/or invocation options for selecting and/or ordering the implemented strategies should be provided.*

The implication of points 3 and 5 in this narrative is that it possible that a GML instance document that claims to validate against a particular profile of GML, but if fact does not, may be incorrectly validated because of previously cached copies of the complete GML schema.

To prevent this problem, this profile defines the namespace for the profile as <http://www.opengis.net/gml/profile/sfgml/1.0>.

The implication of this change is that the GML profiling rules will need to be amended to allow profiles to define their own namespace. In addition this means that the meaning of the word “replace” uses in the narrative for the functional test in sub-clause 7.16.4 should be clarified to mean not only reference the complete gml schema but also use the proper gml namespace rather than the namespace of the profile.

Changes to Web Feature Service Implementation Specification V1.1 (04-094)

Table 1 summarized the additional MIME types that a WFS implementation must support as values for the inputFormat and outputFormat parameters. Using these parameter values, a client application can request schemas compliant with this profile and receive responses that validate against compliant schemas.

vii. Future Work Items

- Harmonize simple feature model with SF RWG work on SF-SQL.
- Harmonize null handled with GML 3.2 when it becomes an adopted specification.

Foreword

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

Introduction

The generation and parsing of Geographic Markup Language (GML) [1] and XML Schema [4, 5] code are required in the implementation of many components that have to deal with GML data. This profile defines a restricted but useful subset of XML-Schema and GML to lower the “*implementation bar*” of time and resources required for an organization to commit for developing software that supports GML. It is hoped that by lowering the effort required to manipulate XML encoded feature data, organizations will be encouraged to invest more time and effort to take greater advantage of GML’s rich functionality.

In both GML and previous Simple Features (SF) specifications for OGC, such as Simple Features for SQL (SF-SQL) [10], features are considered to be objects which can have geometry and other properties. The SF specifications are more restrictive than GML, however, in that geometry is limited to points, lines, and polygons (and collections of these), with linear interpolation between vertices of lines, and planar (flat) surfaces within polygons. This GML Simple Features (GML-SF) profile has a similar understanding of the structure and geometry of features as SF-SQL. However, GML-SF goes beyond SF-SQL in some important ways: GML-SF supports three-dimensional coordinates (location and elevation) on feature geometry, where SF-SQL just supports two dimensions (location). GML-SF also goes beyond SF-SQL by supporting metadata (section 7.4), a means of referencing local or remote resources which could be used for primary/foreign key references (section 8.4.3.10), and dynamic codelists (section 8.4.3.11). These extensions have been incorporated into GML-SF based on their usefulness across a large base of applications.

Development of this profile does not reduce the need for distinct communities of users to develop common application schemas (data models) for information sharing. However, to the extent that users’ application schemas fit within the scope of GML-SF capabilities, this profile facilitates the ability to use WFS for interoperable feature data exchange with much less software development investment.

GML simple features profile

1 Scope

This OGC™ document defines a set of schema encoding rules that allow simple features to be described using GML application schemas.

This OGC™ document defines:

- a) Rigid coding patterns for the use of a subset of XML Schema constructs (XML Schema profile)
- b) Rigid coding patterns for the use of a subset of GML constructs (GML profile).

This document prescribes the encoding of GML application schemas in sufficient detail that client applications do not need to deal with the entire scope of XML-Schema and GML but only need to understand a restricted subset of both specifications in order to be able interpret schema documents generated or referenced by servers offering data encoded in GML. It is expected that making it easier to interpret GML application schemas will enhance interoperability between clients and servers and make the task of implementing client applications less onerous.

This document is applicable to any client wishing to interpret GML application schema documents in an ad hoc manner in order to interact with servers that offer data encoded in GML. That is to say, this document is not meant to address the case where a community of interest has defined a standard GML application schema or set of schemas for exchanging geographic data but rather the case where a client interacts with a previously unknown server offering data encoded in GML using a previously unknown GML application schema.

This profile does not address the semantic issues associated with interpreting a previously unknown GML application schema. Instead this document tries to simplify the processing required to determine which feature types a service offers, and what the structure (in terms of property names and data types) of those feature types is.

2 Conformance

Conformance with this profile shall be checked using all the relevant tests specified in Annex A (normative).

3 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. 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.

- [1] OGC Document 03-105r1, OpenGIS® Geography Markup Language (GML) Implementation Specification, version 3.1, 2003
- [2] W3C, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, 6 October 2000, <http://www.w3.org/TR/REC-xml>
- [3] OGC Document 05-008, OGC Web Service Common Specification, version 0.4.0, 2005
- [4] W3C, XML Schema Part 1: Structures, <http://www.w3.org/TR/xmlschema-1>
- [5] W3C, XML Schema Part 2: Datatypes, <http://www.w3.org/TR/xmlschema-2>
- [6] OGC Document 04-094, OpenGIS Web Feature Service Implementation Specification version 1.1, 2004.
- [7] W3C, Namespaces in XML, <http://www.w3.org/TR/1999/REC-xml-names-19990114>

4 Terms and definitions

For the purposes of this profile, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 05-008] shall apply. In addition, the following terms and definitions apply.

5 Conventions

5.1 Abbreviated terms

Most of the abbreviated terms listed in sub-clause 5.1 of the OGC Web Service Common Specification [3] apply to this document, plus the following abbreviated terms.

API Application Program Interface

5.2 Document terms and definitions

This document uses the specification terms defined in sub-clause 5.3 of the OGC Web Service Common Specification [3].

6 Usage Scenarios

6.1 Simple desktop or browser based map viewer

Allow users to select a collection of features for access and subsequent visualization by a client application.

6.2 Get a collection of features using spatial and non-spatial constraints

A client application is able to process a schema description of the features that a WFS serves sufficiently to be able to generate a valid WFS query, that uses spatial and non-spatial constraints, to obtain one or more feature instances.

6.3 "Value-add" editor (edit geometry and other attribute values)

Users are able to augment geo-spatial information supplied by a data producer with data of their own, creating new feature instances or modifying existing feature instances.

6.4 Create feature instances of specified type or id

A client application is able to process a schema description of the features that a WFS serves sufficiently to be able to generate a valid, new feature instances that may then be inserted into the feature store using a WFS Transaction request.

6.5 Update features with modified geometry and/or attribute values

A client application is able to process a schema description of the features that a WFS serves sufficiently to understand the type of each property of a feature and be able to generate a valid Transaction request that updates the values of one or more properties of a feature. The properties may be spatial or non-spatial.

7 Overview

7.1 Simple feature structure

Features must be defined using a "well-known" and simplified feature model that can be represented using only the GML and XML-Schema components prescribed in this profile.

This profile is concerned with the 'basic' schemas for use with relatively simple systems, such as those that use features that are represented (at least conceptually) by a single database table. That is, they have a flat list of feature properties and the properties may be of simple scalar types like Number, String, and Boolean and may include one or more spatial properties.

7.2 Supported geometries

The following GML geometry property types may be used in a GML application schema that conforms to this profile:

Table 1 – Supported GML Geometric Property Types

| GML Geometric Property Type | Defined in GML Schema File | Restrictions |
|--|-------------------------------------|---|
| <code>gml:PointPropertyType</code> | <code>geometryBasic0d1d.xsd</code> | None |
| <code>gml:CurvePropertyType</code> | <code>geometryBasic0d1d.xsd</code> | The supported property values are <code>gml:LineString</code> or <code>gml:Curve</code> with <code>gml:LineStringSegment</code> segments. |
| <code>gml:SurfacePropertyType</code> | <code>geometryBasic2d.xsd</code> | The supported property value is <code>gml:Polygon</code> or <code>gml:Surface</code> with <code>gml:PolygonPatch</code> patches. |
| <code>gml:GeometryPropertyType</code> | <code>geometryBasic0d1d.xsd</code> | The supported property values are <code>gml:Point</code> , <code>gml:LineString</code> , <code>gml:Curve</code> , <code>gml:Polygon</code> , <code>gml:Surface</code> . |
| <code>gml:MultiPointPropertyType</code> | <code>geometryAggregates.xsd</code> | None |
| <code>gml:MultiCurvePropertyType</code> | <code>geometryAggregates.xsd</code> | Supported members are the same as <code>CurvePropertyType</code> |
| <code>gml:MultiSurfacePropertyType</code> | <code>geometryAggregates.xsd</code> | Supported members are the same as <code>SurfacePropertyType</code> |
| <code>gml:MultiGeometryPropertyType</code> | <code>geometryAggregates.xsd</code> | Supported members are the same as <code>GeometryPropertyType</code> |

Additional requirements for Geometry include:

- a) in all cases, geometry coordinates may only be specified using the **gml:pos** or **gml:posList** elements
- b) coordinate reference systems might have 1, 2 or 3 dimensions
- c) features may have any number of geometric properties

the response feature collection must include the **gml:boundedBy** property with the appropriate **srsName**. All geometries in the feature collection may be assumed to be in the same **srsName** unless otherwise indicated in a particular geometry instance

7.3 CRS support

7.3.1 Introduction

This sub-clause discusses topics related to CRS handling within GML instance documents that validate against application schemas that conform to this profile. Specifically, the following topics are discussed:

1. Homogenous CRS within geometry instances.
2. Scoping rules for srsName.

7.3.2 Homogenous CRS within geometry instances

This profile allows geometry instances within a compliant instance document to be in different coordinate reference systems. However, in order to simplify CRS handling, all the coordinates of all nested components of any individual geometry instance must be in the same CRS and thus the **srsName** attribute may only be specified on the global element of the geometry.

7.3.3 Scoping rules for srsName

A client application must be able to unambiguously determine the CRS of each geometry instance in a GML instance document that validates against a GML application schema that complies with this profile. GML defines the **srsName** attribute for this purpose.

For the purposes of the profile, the **srsName** attribute may be specified at three different levels in an instance document:

1. On the bounding box of the WFS response container, **<wfs:FeatureCollection>**. The **<wfs:FeatureCollection>** element is defined in the WFS specification [6].
2. On the bounding box of each feature instance. The “object” level element of each feature type is defined using the pattern described in sub-clause 8.2.
3. On the root element of each GML geometry instance.

The following example XML fragment shows the **srsName** attribute used at the feature collection, feature instance and geometry instance levels.

```

1. <wfs:FeatureCollection>
2.   <gml:boundedBy>
3.     <gml:Envelope srsName="..."> ... </gml:Envelope>
4.   </gml:boundedBy>
5.   ... feature instances go here ...
6.   <gml:featureMember>
7.     <myns:MyFeature>
8.       <gml:boundedBy>
9.         <gml:Envelope srsName="..."> ... </gml:Envelope>
10.      </gml:boundedBy>
11.      ... zero or more property values go here ...
12.      <myns:geomProperty>
13.        <gml:Polygon srsName="..."> ... </gml:Polygon>

```

```

14.         </myns:geomProperty>
15.         ... zero or more property values go here ...
15.     </myns:MyFeature>
16. </gml:featureMember>
17. ... more feature instances go here ...
18. </wfs:FeatureCollection>

```

Line 3 shows the **srsName** attribute specified at the feature collection level. Line 9 shows the **srsName** attribute specified at the feature instance level. Line 13 show the **srsName** attribute specified at the geometry instance level.

In order to reduce redundancy, instance documents should specify the **srsName** attribute at the highest possible level. The CRS of a geometry instance may then implicitly be assumed to take on the value of the CRS of the bounding box of the containing object (i.e. feature instance or feature collection). This implies that if all geometry instances in a document are in the same CRS, then the best practice for a compliant instance document is to not repeatedly specify that same value for the **srsName** attribute for each geometry instance. Instead, the compliant document should specify a **<gml:boundedBy>** element at the feature collection level and specify a value for the **srsName** attribute on the **<gml:Envelope>** element. Thus, the **srsName** attribute is specified once in the entire document.

In the case where the bounding box at the feature collection level and the feature type level has no value, then the **srsName** attribute must be specified for each geometry instance of the corresponding feature type even if the CRS is the same for all geometry instances.

7.4 Metadata handling

GML application schemas that comply with this profile must use a restricted form of the predefined GML element **gml:metaDataProperty** to reference metadata about a specific feature instance. For this profile, no inline content is permitted for **gml:metaDataProperty**, so metadata must be referenced using a simple xlink. Metadata encoded in-line with the feature instance is not supported.

7.5 Conformance Levels

Conformance to this profile must be viewed from two perspectives; from the perspective of schema generators and the perspective of schema consumers.

A schema generator is any system that generates GML application schemas that comply with this profile. These schemas may be static documents or generated on the fly. In general, schema generators only need to support that subset of this profile that allows them to describe their system and may ignore the rest. For example, a web feature service that only serves point data will never need to generate a schema (in response to a DescribeFeatureType request) that includes curves, surfaces or any of the other supported geometric types. Similarly, a system that does not support value cardinality greater than zero will never generate a schema where maxOccurs is greater than one.

A schema consumer, on the other hand, is any system that reads and interprets GML application schemas that comply with this profile. Schema consumers must be prepared to deal with all aspects of this profile even if it is to report that the system cannot process a particular aspect of this profile. For example, a web feature service client may not be able to render heterogeneous geometry collections but it must be able to recognize this structure in the schema and possibly raise an exception.

8 A rigid coding pattern for GML application schemas

8.1 Introduction

Clause 8 describes a rigid coding pattern for GML application schemas. The main motivation behind this pattern is to limit the set of XML-Schema and GML features that may be used to code a GML application schema. This in turn should simplify the task of building clients that can ingest schema documents that conform to this coding pattern and understand the structure of the feature types defined within.

In the following clauses, schema fragments defined may be combined to create a complete GML application schema.

The schema fragments must be *structurally* encoded exactly as presented in the document. This means that all mandatory elements and attributes presented in the fragment must be included as shown even if they are optional in XML-Schema. Furthermore, no other optional elements, attributes or facets that might be defined in XML-Schema or GML may be used unless specified in this document.

Please note, that these requirements have absolutely nothing to do with the formatting of the XML fragments. They are structural and syntactic requirements, not formatting requirements. White spaces can be used freely to format the generated schema documents in any way.

8.2 Root element

The following XML fragment shows how to encode the GML application schema document's root element:

```

1 <xsd:schema
2   targetNamespace="target_name_space"
3   xmlns:prefix="target_name_space"
4   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
5   xmlns:gml="http://www.opengis.net/gml/profile/sfgml/1.0"
6   elementFormDefault="qualified"
7   version="99.99.99">
```

The attributes shown for the **schema** element can be specified in any order.

Line 2 declares the target namespace for the elements defined in the schema document. The value *target_name_space* is a placeholder for the actual namespace identifier.

Line 3 defines a prefix for the target namespace. The value *prefix* is a placeholder for an actual prefix value. A conformant schema may also declare the target namespace to be the default namespace in which case no prefix is defined at all.

Line 4 declares the prefix for the XML-Schema namespace, which contains all the elements used to define a schema. In this document, the prefix **xsd** is used to represent the namespace for the XML-Schema elements. However, a schema document that conforms to this profile may set the prefix to have any desired value as long as it is correctly bound to the XML-Schema namespace (<http://www.w3.org/2001/XMLSchema>). A conformant schema may also declare the XML-Schema namespace as the default namespace as long as the default namespace has not been previously bound.

Line 5 declares the prefix **gml** for the GML namespace. Although this profile recommends that this prefix be named **gml**, a compliant schema may use any name that conforms to XML-Schema for this prefix as long as it is correctly bound to the GML namespace for prefixes (<http://www.opengis.net/gml/profile/sfgml/1.0>).

Line 6 sets the default value for the **form** attribute for elements to **qualified**. This indicates that locally defined elements are added to the target namespace.

Finally, the **version** attribute should be set to reflect the version of the schema document being generated. The value ‘‘99.99.99’’ is simply a placeholder for any version value, in any format, that has meaning to the entity or organization creating the schema document.

8.3 Importing the GML schemas

The following XML fragment imports the GML feature schema:

```
1 <xsd:import namespace="http://www.opengis.net/gml/profile/sfgml/1.0"
2   schemaLocation="http://[...schema repository path...]/gmlSimpleFeaturesProfile.xsd"/>
```

The value of the **namespace** attribute must match the GML namespace declaration in the root element.

The value of the **schemaLocation** attribute must be a URL (absolute or relative) that resolves to the contents of the GML schema file *gmlSimpleFeaturesProfile.xsd*. The schema fragment above shows a URL that points to the physical file, *gmlSimpleFeaturesProfile.xsd*, but any URL may be used (e.g. an HTTP GET request that resolves to a schema). It should be noted that the **schemaLocation** attribute is a hint for schema aware parsers and may not necessarily be resolved or used. However, application schemas that comply with this profile must still specify the **schemaLocation** attribute.

8.3.1 Importing and including other schemas

A schema compliant with this profile may import or include other schemas as long as those schemas are also compliant with this profile. The coding pattern for importing is

the same as that used in clause 8.3 except that the **namespace** and **schemaLocation** attributes must be relevant to the schema being imported.

8.4 Coding pattern for feature types

8.4.1 Introduction

In GML, a feature is encoded as an XML element with the name of the feature-type. A property of a feature is also encoded as an XML element, whose name is the name of the property. This clause defines the XML-Schema and GML data types that may be used to declare elements that represent features and properties of features.

Following the GML lexical convention, this profile uses *UpperCamelCase* for the XML element names representing Objects or Feature Types, and *lowerCamelCase* for the XML element names representing properties. This provides a visual clue distinguishing the two "layers" within long instance documents with nested elements, and thus assists inspection of Object/property pattern by developers.

8.4.2 Defining feature types

The following XML-Schema fragment shows how to define a feature type in a GML application schema that conforms to this profile:

```

1  <xsd:element name="[FeatureTypeName]"
2      type="[prefix]:[FeatureTypeName]Type"
3      substitutionGroup="gml:_Feature"/>
4
5  <xsd:complexType name="[FeatureTypeName]Type">
6      <xsd:complexContent>
7          <xsd:extension base="gml:AbstractFeatureType">
8              <xsd:sequence>
9
10             <!-- ... zero or more element declarations as described in 8.4.3 ... -->
11
12             </xsd:sequence>
13             <xsd:attribute ref="gml:id" use="required"/>
14         </xsd:extension>
15     </xsd:complexContent>
16 </xsd:complexType>

```

The global element of the feature type is defined in line 1. The value *FeatureTypeName* is a placeholder for the actual name of the feature type. The global element must be defined to be of type *[FeatureTypeName]Type* and must be substitutable for the substitution group head *gml:_Feature*.

Line 5 begins the definition of the XML type that defines the complex type of the feature type. The complex type must be an extension of the GML abstract feature type (line 7) and must contain zero or more element declarations that represent the properties of the feature type. The use of the *sequence* model group in line 8 means that the order in which elements appear in an instance document must match the order in which they are declared in a compliant application schema document. Sub-clause 8.4.3 describes how to code elements of the various supported content types. In addition, the *gml:id* attribute is changed from optional in the GML specification [1] to mandatory in this profile.

8.4.3 Basic data types

XML-Schema defines a rich set of basic data types that can be used to define XML documents. However, since data can originate from any number of sources (all of them having a different set of supported basic types), this document limits the set of available basic types to a smaller subset. The reasoning is that a smaller common set of supported basic data types is likely to be more interoperable.

This profile does not permit the use of named XML Schema type declarations in GML application schemas except as explicitly stated in clause 8.4.2. This applies to both complex and simple types.

In addition to the GML geometric property types defined in Table 2, this profile supports the following list of basic data types:

- a) Integers (with optional specified value range and number of digits)
- b) Reals (with optional specified number of digits, decimal precision and value range)
- c) Character strings (with optional specified maximum length)
- d) Date
- e) Boolean
- f) Binary data
- g) URI references to local or remote resources.
- h) A measurement that includes a reference to units of measurement

The following clauses present XML patterns that act as templates for defining elements whose content type corresponds to one of the basic types supported by this profile. Unless otherwise specified, all elements and attributes presented in the templates, and only those elements and attributes, must be specified.

8.4.3.1 Null values

This sub-clause describes how the absence of a property value is to be interpreted in instances documents that validate against application schemas that conform to this profile. In this profile, the term *null* is used to indicate the absence of a property value.

Whether or not the value of a feature property is interpreted as being null or absent depends on the definition of the XML element into which the property is mapped in a GML application schema and the contents of that element in a GML instance document.

The salient facets of an element's declaration that affect whether a property's value is interpreted as being null or not are the **minOccurs** and **maxOccurs** facets. Table 3

summarizes how the various combinations of **minOccurs**, **maxOccurs** and an element's content are interpreted to set the value of a feature property.

It should be noted, that although XML-Schema [4] allows **minOccurs=maxOccurs=0**, this profile specifically restricts this combination.

Table 3 – Setting the value of a property to NULL

| Value of minOccurs facet on element declaration in GML application schema. | Value of maxOccurs facet on element declaration in GML application schema. | Content of element in a GML instance document. | Interpretation of feature property value. |
|--|--|--|--|
| 0 | 0 | EMPTY | Not allowed by this profile. |
| 0 | 0 | NOT EMPTY | Invalid combination |
| 0 | >=1 | EMPTY | Property value is interpreted as being null or absent. |
| 0 | >=1 | NOT EMPTY | Property value is the content of the element. |
| >=1 | 0 | EMPTY | Invalid according to XML Schema. |
| >=1 | 0 | NOT EMPTY | Invalid according to XML Schema. |
| >=1 | >=1 | EMPTY | Invalid according to XML Schema. |
| >=1 | >=1 | NOT EMPTY | Property value is the content of the element. |

8.4.3.2 Common facets

Table 4 summarizes the common facets that may be used when declaring elements that encode properties in application schemas that comply with this profile.

Table 4 – Allowed element facets

| XML Schema element facet Name | Applicable data types | Encoding Pattern |
|-------------------------------|--|--|
| minInclusive | integer, real, date, dateTime, measurement | <xsd:minInclusive value=" <i>min value</i> "/> |
| minExclusive | integer, real, date, dateTime, measurement | <xsd:minExclusive value=" <i>min value</i> "/> |
| maxInclusive | integer, real, date, dateTime, measurement | <xsd:maxInclusive value=" <i>max value</i> "/> |
| maxExclusive | integer, real, date, dateTime, measurement | <xsd:maxExclusive value=" <i>max value</i> "/> |
| Enumeration | integer, real, date, dateTime, measurement | <xsd:enumeration value=" <i>value</i> "/> |

| | | |
|--|--|--|
| | | |
|--|--|--|

8.4.3.3 Defining elements with integer content

The following XML-Schema fragments show how to declare an element with integer content:

```

1 <xsd:element name="propertyName" minOccurs="0|N" maxOccurs="0|N|unbounded">
2   <xsd:simpleType>
3     <xsd:restriction base="xsd:integer">
4       <xsd:totalDigits value="nDigits"/>
5       <!-- ... Optional facets described is sub-clause 8.4.3.2 ...-->
6     </xsd:restriction>
7   </xsd:simpleType>
8 </xsd:element>

```

SHORT FORM: `<xsd:element name="propertyName" type="xsd:integer" minOccurs="0|N" maxOccurs="0|N|unbounded">`

The value of the mandatory **name** attribute, *propertyName*, is a placeholder for the name of the element being declared which should match the name of feature property being encoded.

The attributes **minOccurs** and **maxOccurs** may be specified and indicate the minimum and maximum number of times that the element must appear in an instance document. If **minOccurs** and **maxOccurs** are omitted, then they are assumed to have the default values defined in XML-Schema for these facets – one (1). The combination `minOccurs=maxOccurs=0` is specifically forbidden by this profile.

An element with integer content (or an integer-valued property) must be derived from the base XML-Schema type **xsd:integer**.

The maximum number of digits in the integer may be specified using the **value** attribute on the **totalDigits** element. The value *nDigits* in the XML fragment is a placeholder for the value representing the maximum number digits in the integer.

The declaration of an element with integer content may also include one or more of the optional facets described in sub-clause 8.4.3.2. If no optional facets are specified in the element declaration, then the *short form* schema fragment may be used instead.

8.4.3.4 Defining elements with real content

The following XML-Schema fragments show how to define elements with real content in a GML application schema that conforms to this profile:

```

1 <xsd:element name="propertyName" minOccurs="0|N" maxOccurs="0|N|unbounded">
2   <xsd:simpleType>
3     <xsd:restriction base="xsd:double|xsd:decimal">
4       ... Optional facets described is sub-clause 8.4.3.2 ...
5       <xsd:totalDigits value="N"/>
6       <xsd:fractionDigits value="N" />
7     </xsd:restriction>
8   </xsd:simpleType>
9 </xsd:element>

```


SHORT FORM: `<xsd:element name="propertyName" type="xsd:double" minOccurs="0|N" maxOccurs="0|N|unbounded">`

The value of the mandatory **name** attribute, *propertyName*, is a placeholder for the name of the element being declared which should match the name of feature property being encoded.

The use of the attributes **minOccurs** and **maxOccurs** is described in section 8.4.3.3.

A real-valued property (or an element with real content) must be of type **xsd:double** or type **xsd:decimal**. If the type is **xsd:decimal** then the facets *totalDigits* and *fractionDigits* listed in lines 5 and 6 must be specified. If the type is **xsd:double**, then the facets *totalDigits* and *fractionDigits* listed in lines 5 and 6 must not be specified.

The declaration of an element with real content may also include one or more of the optional facets described in sub-clause 8.4.3.2. If no optional facets are specified in the element declaration, then the *short form* schema fragment may be used instead.

8.4.3.5 Defining elements with character content

The following XML-Schema fragments show how to declare an element with character content:

```

1 <xsd:element name="propertyName" minOccurs="0|N" maxOccurs="0|N|unbounded">
2   <xsd:simpleType>
3     <xsd:restriction base="xsd:string">
4       <xsd:maxLength value="nCharacters"/>
5       ... Optional facets described in sub-clause 8.4.3.2 ...
6     </xsd:restriction>
7   </xsd:simpleType>
8 </xsd:element>

```

SHORT FORM: `<xsd:element name="propertyName" type="xsd:string" minOccurs="0|N" maxOccurs="0|N|unbounded">`

The value of the mandatory **name** attribute, *propertyName*, is a placeholder for the name of the element being declared which should match the name of feature property being encoded.

The use of the attributes **minOccurs** and **maxOccurs** is described in section 8.4.3.3.

A character string-valued property (or an element with character content) must be derived, by restriction, from the base XML-Schema type **xsd:string**.

The *nCharacters* value of the **value** attribute of the **maxLength** element may be used to indicate the maximum number of characters that the string may contain. Note that this value may **not** be the same as the maximum length in bytes of the character string. The length of the string in bytes may be longer than the *nDigits* value if a multi-byte character set is being used.

The declaration of an element with character content may also include one or more of the optional facets described in sub-clause 8.4.3.2. If no optional facets are specified in the element declaration, then the *short form* schema fragment may be used instead.

8.4.3.6 Defining elements with date content

The following XML-Schema fragments show how to encode an element with date content:

```

1 <xsd:element name="propertyName" minOccurs="0|N" maxOccurs="0|N|unbounded">
2   <xsd:simpleType>
3     <xsd:restriction base="xsd:date|xsd:dateTime">
4       ... Optional facets described is sub-clause 8.4.3.2 ...
5     </xsd:restriction>
6   </xsd:simpleType>
7 </xsd:element>

```

SHORT FORM: `<xsd:element name="propertyName" type="xsd:date | xsd:dateTime" minOccurs="0|N" maxOccurs="0|N|unbounded">`

The value of the mandatory **name** attribute, *propertyName*, is a placeholder for the name of the element being declared which should match the name of feature property being encoded.

The use of the attributes **minOccurs** and **maxOccurs** is described in section 8.4.3.3.

An element that contains date content can be of type **xsd:date** or **xsd:dateTime** depending on whether time is important or not. The actual instances of date values must be encoded according to the ISO8601 standard.

The declaration of an element with date content may also include one or more of the optional facets described in sub-clause 8.4.3.2. . If no optional facets are specified in the element declaration, then the *short form* schema fragment may be used instead.

8.4.3.7 Defining elements with Boolean content

The following XML-Schema fragment shows how to declare an element with Boolean content in a GML application schema that conforms to this profile:

```

1 <xsd:element name="propertyName"
2   type="xsd:boolean"
3   minOccurs="0|N" maxOccurs="0|N|unbounded"/>

```

The value of the mandatory **name** attribute, *propertyName*, is a placeholder for the name of the element being declared which should match the name of feature property being encoded.

The use of the attributes **minOccurs** and **maxOccurs** is described in section 8.4.3.3.

The value space of boolean content is {true, 1, false, 0}.

8.4.3.8 Defining elements with binary content

The following XML-Schema fragment shows how to declare an element with binary content in a GML application schema that conforms to this profile:

```

1 <xsd:element name="propertyName" minOccurs="0|N" maxOccurs="0|N|unbounded">
2   <xsd:complexType>

```

```

3     <xsd:simpleContent>
4         <xsd:extension base="xsd:base64Binary|xsd:hexBinary">
5             <xsd:attribute name="url" type="xsd:anyURI" use="optional"/>
6             <xsd:attribute name="mimeType" type="xsd:string" use="required"/>
7             <xsd:attribute name="role" type="xsd:string" use="optional"/>
8             <xsd:attribute name="length" type="xsd:positiveInteger" use="optional"/>
9         </xsd:extension>
10    </xsd:simpleContent>
11 </xsd:complexType>
12 </xsd:element>

```

The value of the mandatory **name** attribute, *propertyName*, is a placeholder for the name of the element being declared which should match the name of feature property being encoded.

The use of the attributes **minOccurs** and **maxOccurs** is described in section 8.4.3.3.

Binary content can either be referenced from an external URI or encoded inline in base64 or hex format.

When binary data is referenced from an external URI, the **url** attribute must be used to point to the location of the data. The **mimeType** attribute must also be specified to indicate the type or format of binary data that is being referenced. Finally, the optional **role** attribute can be used to assign a user-defined role to the data. The role attribute allows complex binary formats like HDF/EOS, which contains multiple independent binary components, to be supported.

When binary data is encoded inline, the **mimeType** attribute must be specified to indicate the type or format of the binary data. The optional **role** attribute can also be specified to assign a user-defined role to the data. In lined binary data is either encoded in base64 format (indicated by defining the type of the element as **xsd:base64Binary**) or hex (indicated by defining the type of the element as **xsd:hexBinary**). The optional **length** attribute may be used to specify the length of the content of the element in an instance document.

8.4.3.9 Defining elements with geometric content

The following XML-Schema fragment shows how to declare an element with geometric content:

```

1 <xsd:element name="propertyName"
2     type="gml_geometric_propertyType | gml:boundingShapeType"
3     minOccurs="0|N" maxOccurs="0|N|unbounded">

```

The value of the mandatory **name** attribute, *propertyName*, is a placeholder for the name of the element being declared which should match the name of feature property being encoded.

The use of the attributes **minOccurs** and **maxOccurs** is described in section 8.4.3.3.

The value *gml_geometric_propertyType* for the **type** attribute is a placeholder for one of the geometric property types defined in Table 2, sub-clause 7.2. An element with geometric content may also be defined as a **gml:boundingShapeType**.

8.4.3.10 Defining elements that reference other resources

The following XML-Schema fragment shows how to declare an element whose value is a reference to a local or remote resource:

```

1 <xsd:element name="propertyName" type=" gml:ReferenceType" minOccurs="0|1">
2   <xsd:annotation>
3     <xsd:appinfo source="urn:x-gml:targetElement">target element</xsd:appinfo>
4   </xsd:annotation>
5 </xsd:element>

```

The value of the mandatory **name** attribute, *propertyName*, is a placeholder for the name of the element being declared which should match the name of feature property being encoded.

The use of the attributes **minOccurs** and **maxOccurs** is described in section 8.4.3.3.

The value of the mandatory **type** attribute must be **gml:ReferenceType**.

In order to indicate which element is the target of the reference, an XML Schema annotation must be used, as shown above, to specify the fully qualified target element name.

In instance documents, local objects may be identified using only the following shorthand xpointer expression template: a local resource is identified by setting the value of the **xlink:href** attribute to be the fragment separator ‘#’ and the **gml:id** value of the feature. The absence of a URI before the fragment separator indicates that the local document is being referenced. For example:

```
xlink:href="#1013"
```

Similarly, remote objects may be identified by setting the value of the **xlink:href** attribute to be the URI for the remote document, a fragment separator ‘#’ and the id of the object being referenced. For example:

```
xlink:href=http://www.someserver.com/somedoc.xml#1013
```

8.4.3.11 Elements with a code list

The following XML Schema fragment defines the encoding of an element whose content must be from a list of values stored in one or more external dictionaries.

```

1 <xsd:element name="propertyName" minOccurs="0|N" maxOccurs="0|N|unbounded">
2   <xsd:annotation>
3     <xsd:appinfo source="http://www.someserver.com/dictionary.com"/>
4   </xsd:annotation>
5   <xsd:simpleType>
6     <xsd:restriction base="gml:CodeType">
7       <xsd:attribute name="codeSpace" type="xsd:anyURI" use="required"/>

```

```

8     </xsd:restriction>
9     </xsd:simpleType>
10 </xsd:element>

```

The value of the mandatory **name** attribute, *propertyName*, is a placeholder for the name of the element being declared which should match the name of feature property being encoded.

The use of the attributes **minOccurs** and **maxOccurs** is described in section 8.4.3.3.

If the schema designer has a particular dictionary or set of dictionaries in mind for the code list(s), then one or more **appinfo** annotations, as shown at line 3, may be included with the **source** attribute on the **<appinfo>** element pointing to the desired dictionary. In this way, schema consumers may know what the list of codes may from the schema document.

8.4.3.12 Elements with content that is a measurement

The following XML-Schema fragments show how to define elements with content that is considered a measurement and includes a unit of measure in a GML application schema that conforms to this profile:

```

1 <xsd:element name="propertyName" minOccurs="0|N" maxOccurs="0|N|unbounded">
2   <xsd:simpleType>
3     <xsd:restriction base="gml:MeasureType">
4       ... Optional facets described in sub-clause 8.4.3.2 ...
5     </xsd:restriction>
6   </xsd:simpleType>
7 </xsd:element>

```

SHORT FORM: `<xsd:element name="propertyName" type="gml:MeasureType" minOccurs="0|N" maxOccurs="0|N|unbounded">`

The value of the mandatory **name** attribute, *propertyName*, is a placeholder for the name of the element being declared which should match the name of feature property being encoded.

The use of the attributes **minOccurs** and **maxOccurs** is described in section 8.4.3.3.

A real-valued property (or an element with real content) must be of type **gml:MeasureType**.

The declaration of an element with a measurement may also include one or more of the optional facets described in sub-clause 8.4.3.2. If no optional facets are specified in the element declaration, then the *short form* schema fragment may be used instead.

8.5 Comments and annotations

Application schemas that conform to this profile must include an **appinfo** annotation that identifies the application schemas as conforming to this profile. The exact form of the **appinfo** element must be: `<xsd:appinfo>SFGML+V1.0</xsd:appinfo>`. Otherwise, application schemas that conform to this profile may use XML comments and annotations, using the **<annotation>** element, wherever they are legally allowed by XML-Schema and GML.

Annex A (normative)

Conformance testing

A.1 Introduction

This annex outlines how a GML application schema can be tested for compliance with this profile.

It is assumed that a method exists for lexically scanning the application schema being tested in order to access all the elements and attributes contained therein.

The following description is only concerned with interpreting the schema definition in order to ascertain if the schema is compliant with this profile.

The best way to test a GML application schema for compliance to this profile is to build an XML-Schema interpreter that checks all the elements and attributes defined in the application schema for compliance.

The following clauses present each of the XML-Schema pattern templates defined in this profile and describe how to test them for conformance.

A.2 Elements to ignore

All XML comments may be ignored. All XML-Schema annotation elements and sub-elements may be ignored except as directed in this profile.

Attention is drawn to the fact that in this annex, as in the profile, the namespace prefix *xsd* is used for the XML-Schema namespace. This is purely for illustrative purposes and the prefix *xsd* should be considered a placeholder for the actual prefix defined in any particular schema instance.

A.3 General conformance rules

Only the elements and attributes defined or discussed in this profile may appear in a compliant schema regardless of the implicit elements and attributes that may be defined in XML-Schema or GML.

The order in which elements are declared is not important except where the order matters with regard to XML-Schema or GML.

Only opening tags or elements are described in this annex. It is assumed that the schema is well formed and the corresponding closing tags exist.

The order in which attributes appear in opening elements is not important. The order chosen in this document is purely for the sake of clarity.

A.4 Root element

A compliant application schema must be a valid GML application schema, which means that it must be a valid XML-Schema document. The following fragment defines the root element of a compliant XML-Schema document:

```

1 <xsd:schema
2   targetNamespace="target_name_space"
3   xmlns:prefix="target_name_space"
4   xmlns:xmns_prefix="http://www.w3.org/2001/XMLSchema"
5   xmlns:gml="http://www.opengis.net/gml/profile/sfgml/1.0"
6   elementFormDefault="qualified"
7   attributeFormDefault="unqualified"
8   version="0.0.11">

```

Conformance rules:

- 1) the root element must be **xsd:schema** (line 1)
 - a) the attribute **targetNamespace** must be present (line 2)
 - i) its value is user defined
 - b) the attribute **xmlns:prefix** must be present (line 3)
 - i) the value of *prefix* is user defined
 - ii) the namespace value must be same as target namespace(line 2)
 - c) the attribute **xmlns:xmns_prefix** must be present (line 4)
 - i) the value *xmns_prefix* is user defined and is the alias for the XML-Schema namespace; this profile uses **xsd** as an example
 - ii) its value must be 'http://www.w3.org/2001/XMLSchema'
 - d) the attribute **xmlns:gml** must be present (line 5)
 - i) its value must be 'http://www.opengis.net/gml/profile/sfgml/1.0'
 - ii) the namespace alias **gml** is recommended but may be redefined by the user; this profile uses the prefix **gml** as an example
 - e) additional **xmlns:prefix** attributes may be present declaring prefixes for imported namespaces
 - f) the attribute **elementFormDefault** may be present
 - i) its value must be 'qualified'
 - g) the attribute **attributeFormDefault** must be present
 - i) its value must be 'unqualified'
 - h) the attribute **version** must be present
 - i) its value is user defined

A.5 Importing the profile schemas

All features in a compliant application schema must be substitutable for `gml:_Feature` and their definition must be derived from `gml:AbstractFeatureType`.

This implies that a compliant application schema must import the GML profile for simple features schema (i.e. `gmlSimpleFeaturesProfile.xsd`). The following element must appear in a compliant application schema document:

```
1 <xsd:import
2   namespace="http://www.opengis.net/gml/profile/sfgml/1.0"
3   schemaLocation="... SOME_URI.../gmlSimpleFeaturesProfile.xsd"/>
```

Conformance rules:

- 1) an **xsd:import** element must be present
 - a) the attribute **namespace** must be present
 - i) its value must be 'http://www.opengis.net/gml/profile/sfgml/1.0'
 - b) the attribute **schemaLocation** must be present
 - i) its value must be a valid URI reference to `gmlSimpleFeaturesProfile.xsd`

A.6 Importing and including other schemas

A conformant schema may import other schemas as long as they too are compliant with this profile. The only exception is the GML schema which is describe in sub-clause A.4

```
1 <xsd:import
2   namespace="target_namespace_of_schema"
3   schemaLocation="URI_reference_to_schema"/>
```

Conformance rules:

- 1) an **xsd:import** element may be present
 - a) the attribute **namespace** must be present
 - i) its value is user defined
 - b) The attribute **schemaLocation** must be present
 - i) its value must be a valid URI reference to the imported schema

```
1 <xsd:include
2   schemaLocation="URI_reference_to_schema"/>
```

Conformance rules:

- 2) one or more **xsd:include** elements may be present
 - a) For each, the attribute **schemaLocation** must be present
 - i) its value must be a valid URI reference to the imported schema

A.7 Feature types

A.7.1 Global element

A compliant application schema must define one or more feature types by defining one or more global elements for those feature types.

The following XML-Schema fragment defines the global element for a feature type:

```

1 <xsd:element name="FeatureTypeName"
2           type="prefix:FeatureTypeNameType"
3           substitutionGroup="gml:_Feature"/>

```

Conformance rules:

- 1) an **xsd:element** element must be present for each feature type defined in the application schema
 - a) the attribute **name** must be present
 - i) its value is user defined and represents the name of the feature type
 - b) the attribute **type** must be present
 - i) its value must be the name of a complex type defined elsewhere in the document
 - ii) the value must follow the following pattern: 'prefix:FeatureTypeNameType'
 - iii) the *prefix* must match the target namespace prefix defined in the root element of the schema document
 - iv) the *FeatureTypeName* is the same as the value of the **name** attribute
 - v) the suffix must be the literal 'Type'
 - c) the attribute **substitutionGroup** must be present
 - i) its value must be 'gml:_Feature'
 - d) the prefix **gml** is recommended but it may be redefined

A.7.2 Complex type

A complex type must be defined that corresponds to the value of the **type** attribute in the definition of the global element of each feature type.

```

1 <xsd:complexType name="FeatureTypeNameType">
2   <xsd:complexContent>
3     <xsd:extension base="gml:AbstractFeatureType">
4       <xsd:sequence>
5
6         <!-- ...one or more element declarations as described in sec. 8.4.3 ... -->
7
8       </xsd:sequence>
9     </xsd:extension>
10  </xsd:complexContent>
11 </xsd:complexType>

```

Conformance rules:

- 1) a **complexType** element must be present to define the XML type of each feature type

- a) the attribute **name** must be present
 - i) the value must follow the pattern in validation rule A.6.1(b,ii,iii,iv)
- 2) the element **xsd:complexContent** must be present
- 3) the element **xsd:extension** must be present
 - a) the attribute **base** must be present
 - b) its value must be 'gml:AbstractFeatureType'
 - c) the prefix **gml** is recommended but it may be redefined
- 4) the element **xsd:sequence** must be present
- 5) zero or more property definitions must follow

A.8 Properties

A.8.1 Integer valued properties

Each integer valued property must be defined using the following XML-Schema fragment:

```

1  <xsd:element name="propertyName"
2      minOccurs="0|N" maxOccurs="0|N|unbounded">
3      <xsd:simpleType>
4          <xsd:restriction base="xsd:integer">
5              <xsd:totalDigits value="nDigits"/>
6              ... Optional facets describe in sub-clause 8.4.3.2 ...
7          </xsd:restriction>
8      </xsd:simpleType>
9  </xsd:element>

```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a) the attribute **name** must be present (line 1)
 - i) its value is user defined and represents the name of property
 - b) the attribute **minOccurs** may be present (line 2)
 - i) if it is present, its value must be 0 or some integer N
 - ii) if it is not present, the default value is 1
 - c) the attribute **maxOccurs** may be present (line 2)
 - i) if it is present, its value must 0 or some integer N or 'unbounded'
 - ii) if it is not present, the default value is 1
- 2) the element **xsd:simpleType** must be present (line 3)
- 3) the element **xsd:restriction** must be present (line 4)
 - a) the attribute **base** must be present (line 4)
 - i) its value must be 'xsd:integer'
- 4) the element **xsd:totalDigits** may be present (line 5)
 - a) if **xsd:totalDigits** is present, the attribute **value** must be present (line 5)
 - i) its value is user defined and represent the number of digits in the integer

- 5) one or more optional facets (sub-clause 8.4.3.2) may be present (line 6)
 - a) only the facets defined in sub-clause 8.4.3.2 may be included
 - b) for any additional facet, only the attributes defined in sub-clause 8.4.3.2 may be present

A.8.2 Real valued properties

```

1 <xsd:element name="propertyName" minOccurs="0|N" maxOccurs="0|N|unbounded">
2   <xsd:simpleType>
3     <xsd:restriction base="xsd:double|xsd:decimal">
4       ... Optional facets described is sub-clause 8.4.3.2 ...
5       <xsd:totalDigits value="N"/>
6       <xsd:fractionDigits value="N" />
7     </xsd:restriction>
8   </xsd:simpleType>
9 </xsd:element>

```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a) the attribute **name** must be present (line 1)
 - i) its value is user defined and represents the name of property
 - b) the attribute **minOccurs** may be present (line 2)
 - i) if it is present, its value must be 0 or some integer N
 - ii) if it is not present, the default value is 1
 - c) the attribute **maxOccurs** may be present (line 2)
 - i) if it is present, its value must 0 or some integer N or 'unbounded'
 - ii) if it is not present, the default value is 1
- 2) the element **xsd:simpleType** must be present (line 3)
- 3) the element **xsd:restriction** must be present (line 4)
 - a) the attribute **base** must be present (line 4)
 - i) its value may be '**xsd:decimal**' or '**xsd:double**'
 - (1) if the value of the type attribute is **xsd:decimal** then the facets *totalDigits* and *fractionDigits* must be specified (lines 5 & 6)
 - (a) the only allowed attribute on both the *totalDigits* and *fractionDigits* facets is **value**
 - (2) if the value of the type attribute is **xsd:double** then the facts *totalDigits* and *fractionDigits* must not be specified (lines 5 & 6)
- 4) one or more optional facets (sub-clause 8.4.3.2) may be present (line 7)
 - a) only the facets defined in sub-clause 8.4.3.2 may be included
 - b) for any additional facet, only the attributes defined in sub-clause 8.4.3.2 may be present

A.8.4 Character valued properties

```

1 <xsd:element name="propertyName"
2   minOccurs="0|N" maxOccurs="0|N|unbounded">

```

```

2   <xsd:simpleType>
3     <xsd:restriction base="xsd:string">
4       <xsd:maxLength value="nCharacters"/>
5       ... Optional facets described in sub-clause 8.4.3.2 ...
6     </xsd:restriction>
7   </xsd:simpleType>
8 </xsd:element>

```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a) the attribute **name** must be present (line 1)
 - i) its value is user defined and represents the name of property
 - b) the attribute **minOccurs** may be present (line 2)
 - i) if it is present, its value must be 0 or some integer N
 - ii) if it is not present, the default value is 1
 - c) the attribute **maxOccurs** may be present (line 2)
 - i) if it is present its value must 0 or some integer N or 'unbounded'
 - ii) if it is not present the default value is 1
- 2) the element **xsd:simpleType** must be present (line 3)
- 3) the element **xsd:restriction** must be present (line 4)
 - a) the attribute **base** must be present (line 4)
 - i) its value must be 'xsd:string'
- 4) the element **xsd:maxLength** may be present (line 5)
 - a) if present, the attribute **value** must be present (line 5)
 - i) its value is user defined and represent the max number of chars
 - b) if not present, the max length of the string is undefined
- 5) one or more optional facets (sub-clause 8.4.3.2) may be present (line 7)
 - a) only the facets defined in sub-clause 8.4.3.2 may be included
 - b) for any additional facet, only the attributes defined in sub-clause 8.4.3.2 may be present

A.8.5 Character valued properties with a dynamic list of values

```

1 <xsd:element name="propertyName" type="prefix:ListOfValuesType"
2     minOccurs="0|N" maxOccurs="0|N|unbounded"/>
3 <xsd:simpleType name="ListOfValuesType">
4   <xsd:union
5     memberTypes="prefix:ListOfValuesEnumType
6     prefix:ListOfValuesOtherType"/>
7 </xsd:simpleType>
8 <xsd:simpleType name="ListOfValuesEnumType">
9   <xsd:restriction base="xsd:string">
10    <xsd:enumeration value="string"/>
11  </xsd:restriction>
12 </xsd:simpleType>
13 <xsd:simpleType name="ListOfValuesOtherType">
14   <xsd:restriction base="string">
15    <xsd:pattern value="other: \w{2,}"/>
16  </xsd:restriction>
17 </xsd:simpleType>
18 </xsd:element>

```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a) the attribute **name** must be present (line 1)
 - i) its value is user defined and represents the name of property
 - b) the attribute **type** must be present (line 1)
 - i) its value is user defined and is the qualified name of the type that defines the list of values
 - ii) the pattern for the type name must be *prefix:ListOfValuesType*
 - (1) *prefix* is the user-defined target namespace for elements defined in this schema
 - (2) the type name is composed of a user-defined name for the list of values, '*ListOfValues*', and the suffix 'Type'
 - c) the attribute **minOccurs** may be present (line 2)
 - i) if it is present, its value must be 0 or some integer N
 - ii) if it is not present, the default value is 1
 - d) the attribute **maxOccurs** may be present (line 2)
 - i) if it is present its value must 0 or some integer N or 'unbounded'
 - ii) if it is not present the default value is 1
- 2) the element **xsd:simpleType** must be present (line 8)
 - a) the attribute **name** must be present (line 8)
 - i) its value is used defined and must be that same name specified for the type in line 1 above
- 3) the element **xsd:union** must be present (line 4)
 - a) the attribute **memberTypes** must be present (line 5)
 - i) its value must be a list of 2 type names
 - (1) the first name is the name of the type that defines the fixed list of values and is composed of a user-defined name for the values, '*ListOfValues*', and the suffix 'EnumType' (line 5)
 - (2) the second name is the name of the type that defines the dynamic component of the list of values and is composed of a user-defined name for the values, '*ListOfValues*', and the suffix 'OtherType' (line 6)
- 4) the element **xsd:simpleType** must be present (line 8)
 - a) the attribute **name** must be present (line 8)
 - i) its value is user-defined and must be the same as the type name specified in the **memberTypes** attribute (line 5) for the fixed list of enumerated values (*ListOfValuesType*)
- 5) the element **xsd:restriction** must be present (line 9)
 - a) the attribute **base** must be present (line 9)
 - i) its values must be '**xsd:string**'
- 6) one or more **xsd:enumeration** element must be specified (line 10)

- a) the attribute **value** must be present for each **xsd:enumeration** element (line 10) and its values must represent one member of the fixed list of values
- 7) the element **xsd:simpleType** must be present (line 14)
 - a) the attribute **name** must be present (line 14)
 - i) its value is user-defined and must be the same as the type name specified in the **memberTypes** attribute (line 6) for the dynamic components of the list of values (*ListOfValuesOtherType*)
- 8) the element **xsd:restriction** must be present (line 15)
 - a) the attribute **base** must be present (line 15)
 - i) its values must be **xsd:string**
- 9) the element **xsd:pattern** must be present (line 16)
 - a) the attribute **value** must be present (line 16)
 - i) its values must be '*other: \w{2,}*'

A.8.6 Date valued properties

```

1 <xsd:element name="propertyName"
2     minOccurs="0|N" maxOccurs="0|N|unbounded">
3   <xsd:simpleType>
4     <xsd:restriction base="xsd:date | xsd:dateTime">
5       ... Optional facets described in sub-clause 8.4.3.2. ...
6     </xsd:restriction>
7   </xsd:simpleType>
8 </xsd:element>

```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a) the attribute **name** must be present (line 1)
 - i) its value is user defined and represents the name of property
 - b) the attribute **minOccurs** may be present (line 2)
 - i) if it is present, its value must be 0 or some integer N
 - ii) if it is not present, the default value is 1
 - c) the attribute **maxOccurs** may be present (line 2)
 - i) if it is present its value must 0 or some integer N or 'unbounded'
 - ii) if it is not present the default value is 1
- 2) the element **xsd:simpleType** must be present (line 3)
- 3) the element **xsd:restriction** must be present (line 4)
 - a) the attribute **base** must be present (line 15)
 - i) its values must be **xsd:date** or **xsd:dateTime**
- 4) one or more optional facets (sub-clause 8.4.3.2) may be present (line 7)
 - a) only the facets defined in sub-clause 8.4.3.2 may be included
 - b) for any additional facet, only the attributes defined in sub-clause 8.4.3.2 may be present

A.8.7 Boolean valued properties

```
1 <xsd:element name="propertyName" type="xsd:boolean"
2     minOccurs="0|N" maxOccurs="0|N|unbounded"/>
```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a) the attribute **name** must be present (line 1)
 - i) its value is user defined and represents the name of property
 - b) the attribute **type** must be present (line 1)
 - i) its value must be **xsd:boolean**
 - c) the attribute **minOccurs** may be present (line 2)
 - i) if it is present, its value must be 0 or some integer N
 - ii) if it is not present, the default value is 1
 - d) the attribute **maxOccurs** may be present (line 2)
 - i) if it is present, its value must be 0 or some integer N or 'unbounded'
 - ii) if it is not present, the default value is 1

A.8.8 Binary valued properties

```
1 <xsd:element name="propertyName"
2     minOccurs="0|N" maxOccurs="0|N|unbounded">
3   <xsd:complexType>
4     <xsd:simpleContent>
5       <xsd:extension base="xsd:base64Binary|xsd:hexBinary">
6         <xsd:attribute name="url"
7             type="xsd:anyURI" use="optional"/>
8         <xsd:attribute name="mimeType"
9             type="xsd:string" use="required"/>
10        <xsd:attribute name="role"
11            type="xsd:string" use="optional"/>
12      </xsd:extension>
13    </xsd:simpleContent>
14  </xsd:complexType>
15 </xsd:element>
```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a) the attribute **name** must be present (line 1)
 - i) its value is user defined and represents the name of property
 - b) the attribute **minOccurs** may be present (line 2)
 - i) if it is present, its value must be 0 or some integer N
 - ii) if it is not present, the default value is 1
 - c) the attribute **maxOccurs** may be present (line 2)
 - i) if it is present, its value must be 0 or some integer N or 'unbounded'
 - ii) if it is not present, the default value is 1
- 2) the element **xsd:complexType** must be present (line 3)

- 3) the element **xsd:simpleContent** must be present (line 4)
- 4) the element **xsd:extension** must be present (line 5)
 - a) the attribute **base** must be present (line 5)
 - i) its value must one or **xsd:base64Binary** or **xsd:hexBinary**
- 5) the element **xsd:attribute** must be present (line 6)
 - a) the attribute **name** must be present (line 6)
 - i) its value must be 'url'
 - b) the attribute **type** must be present (line 7)
 - i) its value must be 'xsd:anyURI'
 - c) the attribute **use** must be present (line 7)
 - i) its value must be 'optional'
- 6) the element **xsd:attribute** must be present (line 8)
 - a) the attribute **name** must be present (line 8)
 - i) its value must be 'mimeType'
 - b) the attribute **type** must be present (line 9)
 - i) its value must be 'xsd:string'
 - c) the attribute **use** must be present (line 9)
 - i) its value must be 'required'
- 7) the element **xsd:attribute** must be present (line 10)
 - a) the attribute **name** must be present (line 10)
 - i) its value must be 'role'
 - b) the attribute **type** must be present (line 11)
 - i) its value must be 'xsd:string'
 - c) the attribute **use** must be present (line 11)
 - i) its value must be 'optional'

A.8.9 Geometry valued properties

```

1 <xsd:element name="propertyName"
2           type="gml_geometric_propertyType | gml:BoundingShapeType"
3           minOccurs="0|N" maxOccurs="0|N|unbounded">

```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a) the attribute **name** must be present (line 1)
 - i) it value is user defined and represents the name of property
 - b) the attribute **type** must be present (line 2)
 - i) it value must be one of: **gml:PointPropertyType**, **gml:CurvePropertyType**, **gml:SurfacePropertyType**, **gml:GeometryPropertyType**, **gml:MultiPointPropertyType**, **gml:MultiCurvePropertyType**, **gml:MultiSurfacePropertyType**, **gml:MultiGeometryPropertyType**, **gml:BoundingShapeType**

- c) the attribute **minOccurs** may be present (line 3)
 - i) if it is present, its value must be 0 or some integer N
 - ii) if it is not present, the default value is 1
- d) the attribute **maxOccurs** may be present (line 3)
 - i) if it is present, its value must 0 or some integer N or 'unbounded'
 - ii) if it is not present, the default value is 1

A.8.10 Properties with references to other resources

```

1 <xsd:element name="propertyName" type=" gml:ReferenceType"
2           minOccurs="0|1">
3   <xsd:annotation>
4     <xsd:appinfo source="urn:x-gml:targetElement">target element</xsd:appinfo>
5   </xsd:annotation>
6 </xsd:element>

```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a) the attribute **name** must be present (line 1)
 - i) its value is user defined and represents the name of property
 - b) the attribute **type** must be present (line 1)
 - i) its value must be **gml:ReferenceType**
 - c) the attribute minOccurs may be present (line 3)
 - i) if it is present, its value must be 0 or 1
 - ii) if it is not present, the default value is 1
- 2) the element **xsd:annotation** must be present (line 3)
- 3) the element **xsd:appinfo** must be present (line 4)
 - a) the **source** attribute must be present (line 4)
 - i) its value must be *urn:x-gml:targetElement*
 - b) the content of the **xsd:appinfo** element must be the fully qualified name of the element being referenced

A.8.11 Properties with code lists

```

1 <xsd:element name="propertyName" minOccurs="0|N" maxOccurs="0|N|unbounded">
2   <xsd:annotation>
3     <xsd:appinfo source="http://www.someserver.com/dictionary.com"/>
4   </xsd :annotation>
5   <xsd :simpleType>
6     <xsd :restriction base="gml :CodeType">
7       <xsd:attribute name="codeSpace" type="xsd:anyURI" use="required"/>
8     </xsd :restriction>
9   </xsd :simpleType>
10 </xsd:element>

```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a) the attribute **name** must be present (line 1)
 - i) its value is user defined and represents the name of property
 - b) the attribute **minOccurs** may be present (line 1)
 - i) if it is present, its value must be 0 or 1
 - ii) if it is not present, the default value is 1
- 2) the element **xsd:annotation** may be present (line 2)
 - a) one or more **xsd:appinfo** elements may be present (line 3)
 - i) the **source** attribute must be present (line 3)
 - ii) its value must be of type **anyURI**.
- 3) the element **xsd:simpleType** must be present (line 5)
- 4) the element **xsd:restriction** must be present (line 6)
 - a) the attribute **base** must be present (line 6)
 - i) its value must be **gml:CodeType**
- 5) the element **xsd:attribute** must be present (line 7)
 - a) the attribute **name** must be present (line 7)
 - i) its value must be **codeSpace**
 - b) the attribute **type** must be present (line 7)
 - i) its value must be **xsd:anyURI**
 - c) the attribute **use** must be present (line 7)
 - i) its value must be **required**

A.8.12 Measurements

```

1 <xsd:element name="propertyName"
2           minOccurs="0|N" maxOccurs="0|N|unbounded">
3   <xsd:simpleType>
4     <xsd:restriction base="xsd:double">
5       ... Optional facets described in sub-clause 8.4.3.2 ...
6     </xsd:restriction>
7   </xsd:simpleType>
8 </xsd:element>

```

Conformance rules:

- 1) the element **xsd:element** must be present (line 1)
 - a. the attribute **name** must be present (line 1)
 - i. its value is user defined and represents the name of property
 - b. the attribute **minOccurs** may be present (line 2)
 - i. if it is present, its value must be 0 or some integer N
 - ii. if it is not present, the default value is 1

- c. the attribute **maxOccurs** may be present (line 2)
 - i. if it is present, its value must 0 or some integer N or 'unbounded'
 - ii. if it is not present, the default value is 1
- 2) the element **xsd:simpleType** must be present (line 3)
- 3) the element **xsd:restriction** must be present (line 4)
 - a. the attribute **base** must be present (line 4)
 - i. its value may be '**gml:MeasureType**'
- 4) one or more optional facets (sub-clause 8.4.3.2) may be present (line 7)
 - a. only the facets defined in sub-clause 8.4.3.2 may be included
 - b. for any additional facet, only the attributes defined in sub-clause 8.4.3.2 may be present

Annex B (Informative)

Mapping to SQL/MM Geometric Types

B.1 SQL/MM geometric property type mapping

The following tables maps all the geometric property types supported by this profile to their equivalent SQL/MM geometry types. The information in Table was compiled from the mapping presented in Table 14 of the SQL/MM specification [13].

Table 5 – GML geometric property types mapped to SQL/MM types.

| GML Geometric Property Type | Equivalent SQL/MM Type |
|-------------------------------|------------------------|
| gml:PointPropertyType | ST_Point |
| gml:CurvePropertyType | ST_LineString |
| gml:SurfacePropertyType | ST_Polygon |
| gml:GeometryPropertyType | ST_Geometry |
| gml:MultiPointPropertyType | ST_MultiPoint |
| gml:MultiCurvePropertyType | ST_MultiCurve |
| gml:MultiSurfacePropertyType | ST_MultiSurface |
| gml:MultiGeometryPropertyType | ST_GeomCollection |

Annex C (Informative)

Examples

C.1 News item example

The following is a GML application schema that defines two features types, *Reporter* and *NewsItem* and complies with the coding patterns described in this document:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema
  targetNamespace="http://www.cubewerx.com/cw"
  xmlns:cw="http://www.cubewerx.com/cw"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:gml="http://www.opengis.net/gml/profile/sfgml/1.0"
  elementFormDefault="qualified"
  version="1.0">
  <xsd:import namespace="http://www.opengis.net/gml/profile/sfgml/1.0"

  schemaLocation="http://schemas.cubewerx.com/schemas/gml/3.1.1/gmlSimpleFeaturesProfile.xsd"/>
  <!-- =====
  define feature types
  ===== -->
  <xsd:element name="Reporter"
    type="cw:ReporterType"
    substitutionGroup="gml:_Feature"/>
  <xsd:complexType name="ReporterType">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element name="reporterId">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:maxLength value="9"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="firstName"
            minOccurs="0" maxOccurs="1">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:maxLength value="20"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="lastName"
            minOccurs="0" maxOccurs="1">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:maxLength value="20"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="organization"
            minOccurs="0" maxOccurs="1">
            <xsd:simpleType>
              <xsd:restriction base="xsd:string">
                <xsd:maxLength value="50"/>
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>

```

```

</xsd:element>
<xsd:element name="email"
  minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="50"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="age"
  minOccurs="0" maxOccurs="1">
  <xsd:simpleType>
    <xsd:restriction base="xsd:integer">
      <xsd:totalDigits value="10"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element name="photo"
  minOccurs="0" maxOccurs="1">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:base64Binary">
        <xsd:attribute name="url"
          type="xsd:anyURI" use="optional"/>
        <xsd:attribute name="mimeType"
          type="xsd:string" use="required"/>
        <xsd:attribute name="role"
          type="xsd:string" use="optional"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:element name="NewsItem"
  type="cw:NewsItemType"
  substitutionGroup="gml:_Feature"/>
<xsd:complexType name="NewsItemType">
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractFeatureType">
      <xsd:sequence>
        <xsd:element name="location"
          type="gml:PointPropertyType"
          minOccurs="1" maxOccurs="1"/>
        <xsd:element name="reporterId"
          minOccurs="1" maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="9"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="eventDate"
          type="xsd:dateTime"
          minOccurs="1" maxOccurs="1"/>
        <xsd:element name="byLine"
          minOccurs="1" maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="30"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
        <xsd:element name="details"
          minOccurs="1" maxOccurs="1">
          <xsd:simpleType>
            <xsd:restriction base="xsd:string">
              <xsd:maxLength value="20000"/>
            </xsd:restriction>
          </xsd:simpleType>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>

```

```

        </xsd:simpleType>
    </xsd:element>
    <xsd:element name="image"
        minOccurs="0" maxOccurs="5">
        <xsd:complexType>
            <xsd:simpleContent>
                <xsd:extension base="xsd:base64Binary">
                    <xsd:attribute name="url"
                        type="xsd:anyURI" use="optional"/>
                    <xsd:attribute name="mimeType"
                        type="xsd:string" use="required"/>
                    <xsd:attribute name="role"
                        type="xsd:string" use="optional"/>
                </xsd:extension>
            </xsd:simpleContent>
        </xsd:complexType>
    </xsd:element>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

C.2 Roads_bts example from the CIPI1.2 testbed

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="qualified"
    targetNamespace="http://www.opengis.org/cipil.2/level0/bts"
    xmlns:bts="http://www.opengis.org/cipil.2/level0/bts"
    xmlns:gml="http://www.opengis.net/gml/profile/sfgml/1.0"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xsd:annotation>
        <xsd:documentation>roads_bts schema.</xsd:documentation>
    </xsd:annotation>
    <xsd:import namespace="http://www.opengis.net/gml/profile/sfgml/1.0"
        schemaLocation="http://www.someserver.com/some_directory/gmlSimpleFeaturesProfile.xsd"/>
    <xsd:element name="FeatureCollection"
        substitutionGroup="gml:_FeatureCollection"
        type="bts:FeatureCollectionType"/>
    <xsd:complexType name="FeatureCollectionType">
        <xsd:complexContent>
            <xsd:extension base="gml:AbstractFeatureCollectionType"/>
        </xsd:complexContent>
    </xsd:complexType>
    <xsd:element name="Roads_bts" substitutionGroup="gml:_Feature"
        type="bts:Roads_btsType"/>
    <xsd:complexType name="Roads_btsType">
        <xsd:complexContent>
            <xsd:extension base="gml:AbstractFeatureType">
                <xsd:sequence>
                    <xsd:element maxOccurs="1" minOccurs="1" name="Objectid_1">
                        <xsd:simpleType>
                            <xsd:restriction base="xsd:integer">
                                <xsd:totalDigits value="10"/>
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:element>
                    <xsd:element maxOccurs="1" minOccurs="1" name="Objectid">
                        <xsd:simpleType>
                            <xsd:restriction base="xsd:integer">
                                <xsd:totalDigits value="10"/>
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:element>
                    <xsd:element maxOccurs="1" minOccurs="1" name="FNode_">
                        <xsd:simpleType>
                            <xsd:restriction base="xsd:integer">
                                <xsd:totalDigits value="10"/>
                            </xsd:restriction>
                        </xsd:simpleType>
                    </xsd:element>
                </xsd:sequence>
            </xsd:extension>
        </xsd:complexContent>
    </xsd:complexType>

```



```

        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="1" name="TNode_">
    <xsd:simpleType>
        <xsd:restriction base="xsd:integer">
            <xsd:totalDigits value="10"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0" name="LPoly_">
    <xsd:simpleType>
        <xsd:restriction base="xsd:integer">
            <xsd:totalDigits value="10"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0" name="RPoly_">
    <xsd:simpleType>
        <xsd:restriction base="xsd:integer">
            <xsd:totalDigits value="10"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0" name="Length"
    type="gml:MeasureType"/>
<xsd:element maxOccurs="1" minOccurs="1" name="Bdt_roads_">
    <xsd:simpleType>
        <xsd:restriction base="xsd:integer">
            <xsd:totalDigits value="10"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="1" name="Bdt_roads1">
    <xsd:simpleType>
        <xsd:restriction base="xsd:integer">
            <xsd:totalDigits value="10"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0" name="Prefix">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="2"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0" name="Name">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="30"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0" name="Type">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="4"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0" name="Suffix">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">
            <xsd:maxLength value="2"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="1" name="Fcc">
    <xsd:simpleType>
        <xsd:restriction base="xsd:string">

```

```

        <xsd:maxLength value="3"/>
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="1" name="Fips">
  <xsd:simpleType>
    <xsd:restriction base="xsd:string">
      <xsd:maxLength value="11"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0" name="Shape_len">
  <xsd:simpleType>
    <xsd:restriction base="xsd:decimal">
      <xsd:totalDigits value="30"/>
      <xsd:fractionDigits value="15"/>
    </xsd:restriction>
  </xsd:simpleType>
</xsd:element>
<xsd:element maxOccurs="1" minOccurs="0" name="Geometry"
  type="gml:CurvePropertyType"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
</xsd:schema>

```

Annex D (Normative)

Reference schema for GML simple features profile

The following schema document, called `gmlSimpleFeaturesProfile.xsd`, contains only those element declarations from GML that are required to validate a GML application schema that complies with this profile. GML application schemas can ensure compliance with this profile by importing this schema document rather than schema documents from the standard GML 3.1.1 distribution.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<schema
  targetNamespace="http://www.opengis.net/gml/profile/sfgml/1.0"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:sch="http://www.ascc.net/xml/schematron"
  xmlns:gml="http://www.opengis.net/gml/profile/sfgml/1.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  elementFormDefault="qualified"
  version="3.1.1">

  <annotation>
  <appinfo source="urn:opengis:specification:gml:schema-xsd:feature:v3.1.1"/>
  <documentation>
    GML profile for simple features.
    Copyright (c) 2005 OGC, All Rights Reserved.
  </documentation>
  </annotation>

  <import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="http://schemas.opengis.net/gml/3.1.1/xlink/xlinks.xsd"/>

  <attribute name="id" type="ID"/>

  <attribute name="remoteSchema" type="anyURI"/>

  <attributeGroup name="AssociationAttributeGroup">
    <attributeGroup ref="xlink:simpleLink"/>
    <attribute ref="gml:remoteSchema" use="optional"/>
  </attributeGroup>

  <complexType name="MeasureType">
    <simpleContent>
      <extension base="double">
        <attribute name="uom" type="anyURI" use="required"/>
      </extension>
    </simpleContent>
  </complexType>

  <element name="_Object" abstract="true"/>

  <element name="_GML"
    type="gml:AbstractGMLType"
    abstract="true" substitutionGroup="gml:_Object"/>

  <complexType name="AbstractGMLType" abstract="true">
    <sequence>
      <group ref="gml:StandardObjectProperties"/>
    </sequence>
    <attribute ref="gml:id" use="required"/>
  </complexType>
```

```

<group name="StandardObjectProperties">
  <sequence>
    <element ref="gml:metaDataProperty"
      minOccurs="0" maxOccurs="unbounded"/>
    <element ref="gml:description" minOccurs="0"/>
    <element ref="gml:name" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</group>

<element name="metaDataProperty" type="gml:MetaDataPropertyType"/>
<complexType name="MetaDataPropertyType">
  <!-- <sequence minOccurs="0"> -->
    <!-- <element ref="gml:MetaData"/> -->
    <!-- <any processContents="lax"/> -->
  <!-- </sequence> -->
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attribute name="about" type="anyURI" use="optional"/>
</complexType>

<element name="description" type="gml:StringOrRefType"/>

<complexType name="StringOrRefType">
  <simpleContent>
    <extension base="string">
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </extension>
  </simpleContent>
</complexType>

<element name="name" type="gml:CodeType"/>
<complexType name="CodeType">
  <simpleContent>
    <extension base="string">
      <attribute name="codeSpace" type="anyURI" use="optional"/>
    </extension>
  </simpleContent>
</complexType>

<complexType name="ReferenceType">
  <sequence/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="Envelope" type="gml:EnvelopeType"/>
<complexType name="EnvelopeType">
  <sequence>
    <element name="lowerCorner" type="gml:DirectPositionType"/>
    <element name="upperCorner" type="gml:DirectPositionType"/>
  </sequence>
  <attributeGroup ref="gml:SRSReferenceGroup"/>
</complexType>

<element name="Null" type="gml:NullType"/>
<simpleType name="NullType">
  <union memberTypes="gml:NullEnumeration anyURI"/>
</simpleType>
<simpleType name="NullEnumeration">
  <annotation>
    <documentation>
      Some common reasons for a null value:
      innapplicable - the object does not have a value
      missing       - The correct value is not readily
                    available to the sender of this data.
                    Furthermore, a correct value may not exist.
      template     - the value will be available later
      unknown      - The correct value is not known to, and
                    not computable by, the sender of this data.
                    However, a correct value probably exists.
      withheld     - the value is not divulged
      other:reason - as indicated by "reason" string
      Specific communities may agree to assign more strict semantics
    </documentation>
  </annotation>
</simpleType>

```

```

        when these terms are used in a particular context.
    </documentation>
</annotation>
<union>
  <simpleType>
    <restriction base="string">
      <enumeration value="inapplicable"/>
      <enumeration value="missing"/>
      <enumeration value="template"/>
      <enumeration value="unknown"/>
      <enumeration value="withheld"/>
    </restriction>
  </simpleType>
  <simpleType>
    <restriction base="string">
      <pattern value="other:\w{2,}"/>
    </restriction>
  </simpleType>
</union>
</simpleType>

<element name="boundedBy" type="gml:BoundingShapeType"/>
<complexType name="BoundingShapeType">
  <sequence>
    <choice>
      <element ref="gml:Envelope"/>
      <element ref="gml:Null"/>
    </choice>
  </sequence>
</complexType>

<!-- ===== -->
<!-- ==                                     == -->
<!-- ===== -->

<element name="_Feature"
  type="gml:AbstractFeatureType"
  abstract="true"
  substitutionGroup="gml:_GML"/>
<complexType name="AbstractFeatureType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:boundedBy" minOccurs="0"/>
        <!-- additional properties must be specified
           in an application schema -->
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="FeatureCollection"
  type="gml:FeatureCollectionType"
  substitutionGroup="gml:_Feature"/>
<complexType name="FeatureCollectionType">
  <complexContent>
    <extension base="gml:AbstractFeatureCollectionType"/>
  </complexContent>
</complexType>
<complexType name="AbstractFeatureCollectionType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:featureMember"
          minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:featureMembers" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="featureMember" type="gml:FeaturePropertyType"/>

```

```

<complexType name="FeaturePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:_Feature"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<element name="featureMembers" type="gml:FeatureArrayPropertyType"/>
<complexType name="FeatureArrayPropertyType">
  <sequence>
    <element ref="gml:_Feature" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<!-- ===== -->
<!-- == == -->
<!-- ===== -->
<simpleType name="doubleList">
  <list itemType="double"/>
</simpleType>
<simpleType name="NCNameList">
  <list itemType="NCName"/>
</simpleType>
<element name="pos" type="gml:DirectPositionType"/>
<complexType name="DirectPositionType">
  <simpleContent>
    <extension base="gml:doubleList">
      <attributeGroup ref="gml:SRSReferenceGroup"/>
    </extension>
  </simpleContent>
</complexType>
<element name="posList" type="gml:DirectPositionListType"/>
<complexType name="DirectPositionListType">
  <simpleContent>
    <extension base="gml:doubleList">
      <attributeGroup ref="gml:SRSReferenceGroup"/>
      <attribute name="count" type="positiveInteger" use="optional"/>
    </extension>
  </simpleContent>
</complexType>

<attributeGroup name="SRSReferenceGroup">
  <attribute name="srsName" type="anyURI" use="optional"/>
  <attribute name="srsDimension" type="positiveInteger" use="optional"/>
  <attributeGroup ref="gml:SRSInformationGroup"/>
</attributeGroup>

<attributeGroup name="SRSInformationGroup">
  <attribute name="axisLabels" type="gml:NCNameList" use="optional"/>
  <attribute name="uomLabels" type="gml:NCNameList" use="optional"/>
</attributeGroup>

<!-- ===== -->
<!-- == == -->
<!-- ===== -->

<simpleType name="CurveInterpolationType">
  <restriction base="string">
    <enumeration value="linear"/>
  </restriction>
</simpleType>
<simpleType name="SurfaceInterpolationType">
  <restriction base="string">
    <enumeration value="planar"/>
  </restriction>
</simpleType>

<element name=" Geometry"
  type="gml:AbstractGeometryType"
  substitutionGroup="gml:_GML" abstract="true"/>
<complexType name="AbstractGeometryType" abstract="true">
  <complexContent>

```

```

    <extension base="gml:AbstractGMLType">
      <attributeGroup ref="gml:SRSReferenceGroup"/>
    </extension>
  </complexContent>
</complexType>

<element name="_GeometricPrimitive"
  type="gml:AbstractGeometricPrimitiveType"
  substitutionGroup="gml:_Geometry" abstract="true"/>
<complexType name="AbstractGeometricPrimitiveType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGeometryType"/>
  </complexContent>
</complexType>

<element name="_GeometricAggregate"
  type="gml:AbstractGeometricAggregateType"
  substitutionGroup="gml:_Geometry" abstract="true"/>
<complexType name="AbstractGeometricAggregateType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGeometryType"/>
  </complexContent>
</complexType>

<element name="_Ring"
  type="gml:AbstractRingType"
  substitutionGroup="gml:_Geometry" abstract="true"/>
<complexType name="AbstractRingType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGeometryType"/>
  </complexContent>
</complexType>

<element name="_Surface"
  type="gml:AbstractSurfaceType"
  substitutionGroup="gml:_GeometricPrimitive" abstract="true"/>
<complexType name="AbstractSurfaceType">
  <complexContent>
    <extension base="gml:AbstractGeometricPrimitiveType"/>
  </complexContent>
</complexType>

<element name="_SurfacePatch"
  type="gml:AbstractSurfacePatchType" abstract="true"/>
<complexType name="AbstractSurfacePatchType" abstract="true"/>

<element name="_Curve"
  type="gml:AbstractCurveType"
  substitutionGroup="gml:_GeometricPrimitive" abstract="true"/>
<complexType name="AbstractCurveType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGeometricPrimitiveType"/>
  </complexContent>
</complexType>

<element name="_CurveSegment"
  type="gml:AbstractCurveSegmentType" abstract="true"/>
<complexType name="AbstractCurveSegmentType" abstract="true">
  <sequence/>
  <attribute name="numDerivativesAtStart"
    type="integer" use="optional" fixed="0"/>
  <attribute name="numDerivativesAtEnd"
    type="integer" use="optional" fixed="0"/>
  <attribute name="numDerivativeInterior"
    type="integer" use="optional" fixed="0"/>
</complexType>

<!-- ===== -->
<!-- == == -->
<!-- ===== -->
<complexType name="PointPropertyType">

```

```

    <sequence minOccurs="0">
      <element ref="gml:Point"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <complexType name="PointArrayPropertyType">
    <sequence>
      <element ref="gml:Point" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <element name="Point"
    type="gml:PointType"
    substitutionGroup="gml:_GeometricPrimitive"/>
  <complexType name="PointType">
    <complexContent>
      <extension base="gml:AbstractGeometricPrimitiveType">
        <sequence>
          <element ref="gml:pos"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="CurvePropertyType">
    <sequence minOccurs="0">
      <element ref="gml:_Curve"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <complexType name="CurveArrayPropertyType">
    <sequence>
      <element ref="gml:_Curve" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

  <!--
  <element name="LineString"
    type="gml:LineStringType"
    substitutionGroup="gml:_Curve"/>
  <complexType name="LineStringType">
    <complexContent>
      <extension base="gml:AbstractCurveType">
        <sequence>
          <element ref="gml:posList"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  -->

  <element name="LineStringSegment"
    type="gml:LineStringSegmentType"
    substitutionGroup="gml:_CurveSegment"/>
  <complexType name="LineStringSegmentType">
    <complexContent>
      <extension base="gml:AbstractCurveSegmentType">
        <sequence>
          <element ref="gml:posList"/>
        </sequence>
        <attribute name="interpolation"
          type="gml:CurveInterpolationType" fixed="linear"/>
      </extension>
    </complexContent>
  </complexType>

  <element name="segments" type="gml:CurveSegmentArrayPropertyType"/>
  <complexType name="CurveSegmentArrayPropertyType">
    <sequence>
      <element ref="gml:_CurveSegment" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>

```



```

<element name="Curve" type="gml:CurveType"
      substitutionGroup="gml:_Curve"/>
<complexType name="CurveType">
  <complexContent>
    <extension base="gml:AbstractCurveType">
      <sequence>
        <element ref="gml:segments"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="SurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:_Surface"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<complexType name="SurfaceArrayPropertyType">
  <sequence>
    <element ref="gml:_Surface" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<element name="Surface" type="gml:SurfaceType"
      substitutionGroup="gml:_Surface"/>
<complexType name="SurfaceType">
  <complexContent>
    <extension base="gml:AbstractSurfaceType">
      <sequence>
        <element ref="gml:patches"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="patches" type="gml:SurfacePatchArrayPropertyType"/>
<complexType name="SurfacePatchArrayPropertyType">
  <sequence>
    <element ref="gml:_SurfacePatch" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<element name="PolygonPatch" type="gml:PolygonPatchType"
      substitutionGroup="gml:_SurfacePatch"/>
<complexType name="PolygonPatchType">
  <complexContent>
    <extension base="gml:AbstractSurfacePatchType">
      <sequence>
        <element ref="gml:exterior" minOccurs="0"/>
        <element ref="gml:interior" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="interpolation"
        type="gml:SurfaceInterpolationType"
        fixed="planar"/>
    </extension>
  </complexContent>
</complexType>

<!--
<element name="Polygon"
  type="gml:PolygonType"
  substitutionGroup="gml:_Surface"/>
<complexType name="PolygonType">
  <complexContent>
    <extension base="gml:AbstractSurfaceType">
      <sequence>
        <element ref="gml:exterior"
          minOccurs="0"/>

```

```

        <element ref="gml:interior"
            minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</extension>
</complexContent>
</complexType>
-->

<element name="exterior" type="gml:AbstractRingPropertyType"/>
<element name="interior" type="gml:AbstractRingPropertyType"/>
<complexType name="AbstractRingPropertyType">
    <sequence>
        <element ref="gml:LinearRing"/>
    </sequence>
</complexType>
<element name="LinearRing"
    type="gml:LinearRingType"
    substitutionGroup="gml:_Ring"/>
<complexType name="LinearRingType">
    <complexContent>
        <extension base="gml:AbstractRingType">
            <sequence>
                <element ref="gml:posList"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="GeometryPropertyType">
    <sequence minOccurs="0">
        <element ref="gml:_Geometry"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<complexType name="GeometryArrayPropertyType">
    <sequence>
        <element ref="gml:_Geometry" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
</complexType>
<!-- ===== -->
<!-- == == == -->
<!-- ===== -->
<complexType name="MultiPointPropertyType">
    <sequence minOccurs="0">
        <element ref="gml:MultiPoint"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<element name="MultiPoint"
    type="gml:MultiPointType"
    substitutionGroup="gml:_GeometricAggregate"/>
<complexType name="MultiPointType">
    <complexContent>
        <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
                <element ref="gml:pointMember"
                    minOccurs="0" maxOccurs="unbounded"/>
                <element ref="gml:pointMembers" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<element name="pointMember" type="gml:PointPropertyType"/>
<element name="pointMembers" type="gml:PointArrayPropertyType"/>

<complexType name="MultiCurvePropertyType">
    <sequence minOccurs="0">
        <element ref="gml:MultiCurve"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

```

```

<element name="MultiCurve"
  type="gml:MultiCurveType"
  substitutionGroup="gml:_GeometricAggregate"/>
<complexType name="MultiCurveType">
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>
        <element ref="gml:curveMember"
          minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:curveMembers" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="curveMember" type="gml:CurvePropertyType"/>
<element name="curveMembers" type="gml:CurveArrayPropertyType"/>

<complexType name="MultiSurfacePropertyType">
  <sequence minOccurs="0">
    <element ref="gml:MultiSurface"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="MultiSurface"
  type="gml:MultiSurfaceType"
  substitutionGroup="gml:_GeometricAggregate"/>
<complexType name="MultiSurfaceType">
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>
        <element ref="gml:surfaceMember"
          minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:surfaceMembers" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="surfaceMember" type="gml:SurfacePropertyType"/>
<element name="surfaceMembers" type="gml:SurfaceArrayPropertyType"/>

<complexType name="MultiGeometryPropertyType">
  <sequence minOccurs="0">
    <element ref="gml:_GeometricAggregate"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="MultiGeometry"
  type="gml:MultiGeometryType"
  substitutionGroup="gml:_GeometricAggregate"/>
<complexType name="MultiGeometryType">
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>
        <element ref="gml:geometryMember"
          minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:geometryMembers" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="geometryMember" type="gml:GeometryPropertyType"/>
<element name="geometryMembers" type="gml:GeometryArrayPropertyType"/>

<!-- ===== -->
<!-- ===== -->
<!-- ===== -->

<element name="Definition"

```

```

        type="gml:DefinitionType"
        substitutionGroup="gml:_GML"/>
<complexType name="DefinitionType">
  <complexContent>
    <restriction base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:metaDataProperty"
          minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:description" minOccurs="0"/>
        <element ref="gml:name" maxOccurs="unbounded"/>
      </sequence>
      <attribute ref="gml:id" use="required"/>
    </restriction>
  </complexContent>
</complexType>

<element name="Dictionary"
  type="gml:DictionaryType"
  substitutionGroup="gml:Definition"/>
<complexType name="DictionaryType">
  <complexContent>
    <extension base="gml:DefinitionType">
      <sequence minOccurs="0" maxOccurs="unbounded">
        <choice>
          <element ref="gml:dictionaryEntry"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="dictionaryEntry" type="gml:DictionaryEntryType"/>
<complexType name="DictionaryEntryType">
  <sequence minOccurs="0">
    <element ref="gml:Definition"/>
  </sequence>
  <!-- <attributeGroup ref="gml:AssociationAttributeGroup"/> -->
</complexType>
</schema>

```

Bibliography

- [9] OGC Document 04-024, OpenGIS Web Map Service Implementation Specification version 1.3, 2004.
- [10] OGC Document 02-004, Patterns in GML, 2002.
- [11] OGC Document 01-025, Simple GML: Geography Markup Language (GML) 2.0 Implementation Profile, 2001
- [12] OGC Document 99-049, OpenGIS® Simple Features Specification for SQL, Revision 1.1, 1999.
- [13] OGC Document 05-010, URNs of definitions in ogc namespace, 2005
- [14] IETF RFC 1521, N. Borenstein, et al., MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies, <http://www.ietf.org/rfc/rfc1952.txt>, 1993.
- [15] ISO/IEC JTC 1/SC 32/WG 4, SQL Multimedia and Application Packages (SQL/MM) – Part 3: Spatial, Mark Ashworth, March 1999