Open Geospatial Consortium Inc.

Date: 2006-04-25

Reference number of this document: OGC® 06-049r1

Version: 1.0

Category: OpenGIS® Implementation Specification Profile

Editor: Panagiotis (Peter) A. Vretanos

Geography Markup Language (GML) simple features profile

Document type: OpenGIS® Implementation Specification
Document subtype: GML 3.1.1 Profile
Document stage: Version 1.0

Contents

Page

Tables and Figures

i. Preface

This approved OpenGIS® Implementation Specification defines a Simple Features profile of the Geography Markup Language version 3.1.1. This Simple Features Profile has been aligned with the OpenGIS Implementation Specification Simple Features for SQL version 1.1. Simple Features include: Point, Curve (LineString), Surface (Polygon), Geometry, MultiPoint, MultiCurve, MultiSurface, and MulitGeometry. The detailed abstract model for OGC features and geometry can be found in the OGC Abstract Specification, Topic Volume 1: Features (which is equivalent to ISO 19107).

This Simple Features profile of GML began as 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.

The Open Geospatial Consortium (OGC) is an international industry consortium of more than 300 companies, government agencies, and universities participating in a consensus process to develop publicly available geo-processing specifications.

ii. Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2: Rules for the structure and drafting of International Standards. In particular, the word "shall" (not "must") is the verb form used to indicate a requirement to be strictly followed to comply with this specification.

iii. 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

iv. 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**
Arctur, David	ESRI
Bruce, Craig	CubeWerx Inc.
Burggraf, David	Galdos Inc.
Cox, Simon	CSIRO
Daisey, Paul	U.S. Census Bureau
Daly, Martin	Cadcorp Ltd.

Contact	Company
Danko, David	ESRI
Herring, John	Oracle
Keens, Stephen	PCI Ltd.
Keighan, Edric	CubeWerx Inc.
Kyle, Martin	Galdos Inc.
May, Brian	Cubewerx Inc.
Lake, Ron	Galdos Inc.
Lemieux, Sylvain	PCI Ltd.
Pomakis, Keith	CubeWerx Inc.
Portele, Clemens	Interactive Instrument GmbH
Ramsey, Paul	Refractions Research Inc.
Scarponcini, Paul	Bentley Systems Inc.
Sonnet, Jerome	IONIC
Whiteside, Arliss	BAE Systems E&IS

v. Revision history

See Annex F.

vi. Changes to the OpenGIS® Abstract Specification

None.

vii. Changes to the OpenGIS® Implementation Specifications

The following table lists the WFS operations and parameters that should be changed to accommodate this profile.

Additional values for inputFormat and outputFormat parameters

Request	AttributeName	MIME Type	Description
DescribeFeatureType	outputFormat	text/xml; subType=*gmlSubtype*[1]	The WFS should respond by generating a GML application schema that complies with this profile specification
GetFeature	outputFormat	text/xml; subType=*gmlSubtype*[1]	The WFS should respond by generating an instance document that validates against a schema document that complies with this profile
Transaction (Insert)	inputFormat	text/xml; subType=*gmlSubtype*[1]	The feature instance in the insert action validates against a schema that complies with this profile.

[1] *gmlSubtype* is a placeholder for a string with the value **gml/3.1.1/profiles/gmlsf/1.0.0/[0|1|2]**.

Examples

The following examples show fragments from the capabilities document of a WFS implementation that can generate schemas that comply with this GML 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/profiles/gmlsf/1.0.0/0</Value>
10.         <ows:Value>text/xml; subType=gml/3.1.1/profiles/gmlsf/1.0.0/1</Value>
10.      </ows:Parameter>
11.   </ows:Operation>
```

Lines 9 and 10 indicate that this WFS can generate a GML application schema that complies with levels 0 and 1 of this profile.

DescribeFeatureType XML-encoded request:

The following example XML fragment shows a DescribeFeatureType request indicating that the response should comply with level 1 of this profile:

```
<DescribeFeatureType outputFormat="text/xml; subType=gml/3.1.1/profiles/gmlsf/1.0.0/1"/>
```

If the server does not support the requested compliance level then an exception should be raised.

DescribeFeatureType KVP-encode request:

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

```
http://www.BlueOx.org/wfs/wfs.cgi?service=WFS&version=1.1.0&request=DescribeFeatureType&o
utputFormat=text/xml;+subType=gml/3.1.1/profiles/gmlsf/1.0.0/0
```

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

The table above 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.

viii. Future Work Items

a) Harmonize null handling with GML 3.2 when it becomes an adopted specification.

Foreword

This document cancels and replaces OGC Discussion Paper 03-003r10, titled “Level 0 Profile of GML3 for WFS”.

This document includes five annexes; Annexes A, D, and E are normative, and Annexes B and C are informative.

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 GML specification declares a large number of XML elements and attributes meant to support a wide variety of capabilities. For example, the GML specification can encode dynamic features, spatial and temporal topology, complex geometric property types and coverages. With such a wide scope, interoperability can only be achieved by defining profiles of GML that deal with a restricted subset of GML capabilities. Such profiles limit the number of GML object types that can appear in compliant schemas and consequently are easier to process.

The generation and parsing of Geographic Markup Language (GML) [OGC 03-105r1] and XML Schema [W3C XML-1, W3C XML-2] code are required in the implementation of many components that deal with GML encoded content. 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.

Development of this profile does not reduce the need for distinct communities of users to develop 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.

Geography Markup Language (GML) simple features profile

1 Scope

This OpenGIS® specification defines a simplified profile of GML 3.1.1 that supports GML features and a limited set of linearly interpolated geometric types. A set of application schema encoding rules is defined that allow features to be encoded 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 referenced by 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 specifies three compliance levels for this profile. Level 0 compliance is the simplest and easiest to understand, but does not support the entire set of GML features. Level 1 compliance is somewhat more complex and difficult to understand, but includes more feature encoding abilities sometimes needed. The “full” (level 2) compliance is the most complex, and is intended to include all the feature encoding abilities provided by the OpenGIS® Simple Features for SQL Implementation Specification (version 1.1).

This document is applicable to any client wishing to interpret GML application schema documents in an ad hoc manner in order to interact with data encoded in GML. That is, 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, as well as the structure of those feature types (in terms of property names and data types).

2 Conformance

2.1 Introduction

This document defines three compliance levels called SF-0, SF-1, and SF-2.

Compliance level SF-0 limits the property types of both spatial and non-spatial properties. Non-spatial properties are limited to being of type: integer, measurement, date, boolean, binary, URI, character or real; and cardinality of non-spatial properties is limited to at most one. Spatial properties are limited to being of type: point, linearly interpolated curve, planar surface, or aggregates thereof. In GML instance documents compliant with level SF-0, values for a particular property may be encoded inline or by-reference but not both. By-reference property values are only supported through the use of the type gml:ReferenceType. The more generalized GML property-type pattern allowing mixed inline and by-reference encoded property values within the same instance document is disallowed in this level.

Compliance level SF-1 supports everything in compliance level SF-0 but removes the restriction on value cardinality and allows user defined non-spatial property types to be used. As with compliance level SF-0, in GML instance documents compliant with level SF-1, values for a particular property may be encoded inline or by-reference but not both. By-reference property values are only supported through the use of the type gml:ReferenceType. The more generalized GML property-type pattern allowing mixed inline and by-reference encoded property values within the same instance document is disallowed in this level.

Finally, compliance level SF-2 essentially corresponds to the OGC Simple Features specification [1]. There are no restrictions placed on non-spatial properties. Spatial properties, however, are limited to the set of geometric types supported by compliance levels SF-0 and SF-1. In instance documents, the restriction on remote or referenced geometric property values, found in compliance levels SF-0 and SF-1, is removed.

The rationale for these specific compliance levels and associated restrictions is based on practical experience by implementers gained since 2003. Level SF-0 is the most restrictive, but much existing geospatial data can be represented within its stated restrictions. Building servers and clients to compliance level SF-0 will create greater opportunity for interoperable data exchange, because of the reduced number of optional aspects permitted and the corresponding reduction in vendor-specific functionality likely to occur. Compliance level SF-1 has been found to be useful in projects requiring multi-valued (non-spatial) properties of features. And compliance level SF-2 allows servers to host sophisticated information models without complicating the spatial properties beyond those allowed in levels SF-0 and SF-1. This level allows all Simple Features for SQL (SF-SQL) databases to represent any data they may have. The benefit of restricting the geometry model to the types included here, is to foster maximum interoperability of the geospatial location data. Table 1 summarizes the capabilities of each compliance level.

Table 1 — Compliance level capabilities matrix

	Level SF-0	Level SF-1	Level SF-2
restricted set of built-in non-spatial property types	Yes[1]	Yes[1]	No
restricted set of spatial property types	Yes[2]	Yes[2]	Yes[2]
user-defined property types	No	Yes	Yes
cardinality of properties	0 or 1	0…unbounded	0…unbounded
non-spatial property values references	Yes[3]	Yes[3]	Yes
spatial property values references	Yes[3]	Yes[3]	Yes
1. string, integer, measurement, date, real, binary, boolean, URI 2. Point, Curve (LineString), Surface (Polygon), Geometry, MultiPoint, MultiCurve, MultiSurface, MultiGeometry 3. In levels 0 and 1, remote values for properties are supported **only** through the use of the type gml:ReferenceType. The more generalized GML property-type pattern allowing mixed inline and by-reference encoded property values within the same instance document is disallowed.			

2.2 Compliance levels

Table 2 lists the compliance levels defined in this document. The table specifies a numeric identifier for each compliance level, the relevant subclauses in clauses 7 and 8 and annexes D and E that apply to the compliance level, and the compliance tests in ANNEX A that shall be satisfied by each level.

Table 2 — Compliance levels

Compliance level Identifier	Relevant subclauses and annexes	Compliance tests
SF-0	7, 8, D.1, E	A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.8, A.9, A.10.1, A.10.2, A.10.3, A.10.4, A.10.5, A.10.6, A.10.7, A.10.8, A.10.9, A.10.10, A.10.11, A.10.12
SF-1	7, 9, D.1, E	All compliance tests defined in Annex A.
SF-2	7, 10, D.2, E	A.1, A.2, A.3, A.4, A.5, A.6, A.7, A.9, A.10.8

NOTE Some subclauses and compliance tests listed above contain provisions that depend on the compliance level.

Compliance with this profile shall be checked using all the relevant tests in Annex A, as specified in Table 2 for each compliance level.

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.

OGC 03-105r1, OpenGIS® Geography Markup Language (GML) Implementation Specification, version 3.1

OGC 04-092r4, normative schema document package for [OGC 03-105r1]

OGC 04-094, OpenGIS Web Feature Service Implementation Specification version 1.1

OGC 05-008, OGC Web Service Common Specification, version 1.0

OGC 05-099r1, GML 3.1.1 simple dictionary profile, version 1.0

W3C XML-0, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, 6 October 2000, http://www.w3.org/TR/REC-xml

W3C XML-1, XML Schema Part 1: Structures, http://www.w3.org/TR/xmlschema-1

W3C XML-2, XML Schema Part 2: Datatypes, http://www.w3.org/TR/xmlschema-2

W3C, Namespaces in XML, http://www.w3.org/TR/1999/REC-xml-names-19990114

Recommended Reading

As this is a profile of the Geography Markup Language [OGC 03-105r1], the reader is expected to have some familiarity with GML. The GML specification is a large document covering many topics. The purpose of this section is to direct the reader to specific clauses in the GML specification that should be reviewed in order to better understand the content of this document. The following table lists the clauses and specific subclauses from the GML specification [OGC 03-105r1] that should be reviewed:

Table 3 — Relevant GML 3.1.1 specification clauses and subclauses

Clause from GML specification	Relevant sub-clauses
1. Scope	All
4. Terms and definitions	All
5. Conventions	5.1, 5.2, 5.4
6. Overview of GML schemas	All
7. GML schemas – general rules and base schemas	7.1, 7.2, 7.3.1, 7.3.3.2, 7.3.3.3, 7.4, 7.5.1, 7.5.2, 7.5.3, 7.5.4, 7.5.6.2, 7.5.6.3, 7.5.6.5,
8. GML schemas – feature model	8.1, 8.2.1, 8.2.2, 8.2.3, 8.2.6, 8.2.7, 8.3
9. GML schemas basic geometry	9.1, 9.1.1, 9.1.2, 9.1.3, 9.1.5, 9.2.1, 9.2.2
10. GML schemas – more geometric primitives	102.1.1, 10.2.1.2, 10.2.1.3, 10.2.1.5, 10.2.2.1, 10.2.2.2, 10.2.2.3, 10.2.2.5
11. GML schemas – geometric complex, geometric composites and geometric aggregates	11.3.1, 11.3.2, 11.3.3,11.3.4
12. Coordinate reference systems schemas	12.1.2, 12.1.3
22. Profiles	All
23. Rules for application schemas	23.1 to 23.5

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.

5 Conventions

5.1 Abbreviated terms

Most of the abbreviated terms listed in sub-clause 5.1 of the OGC Web Service Common Specification [OGC 05-008] 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 [OGC 05-008].

5.3 Notations used in XML Schema fragments

The following notations are used in XML Schema fragment presented in this document:

1) Brackets ([]) are used to denote constructs that can be optionally specified. In the following example:

```
<xsd:element name="MyElement" minOccurs="0" [maxOccurs="1"]>
```

the brackets around *maxOccurs="1"* mean that this construct is optional and can be omitted.

2) The vertical bar (|) notation is used to separate alternatives. In the following example:

```
<xsd:element name="MyElement" minOccurs="0" maxOccurs="0|1|unbounded">
```

the value of maxOccurs can be one of 0, 1 or unbounded.

3) Italics are used in XML Schema fragments to denote values that must be specified by the process or person creating the application schema. In the following example:

```
http://…schema repository URL…/schemas/mySchema.xsd
```

the italic string *"...schema repository URL…"* indicates the actual schema repository URL would be substituted by the person or process creating the application schema.

5.4 Document terms and definitions

The following specification terms and definitions are used in this document:

a) shall – verb form used to indicate a requirement to be strictly followed to comply to this specification, from which no deviation is permitted

b) should – verb form used to indicate desirable ability or use, without mentioning or excluding other possibilities

c) may – verb form used to indicate an action permissible within the limits of this specification

d) can – verb form used for statements of possibility

e) informative – a part of a document that is provided for explanation, but is not required

f) normative – a part of a standards document that is required

g) annex – an auxiliary part of a document

h) clause – a major part of a document

i) subclause – a secondary part of a clause or annex

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 can then be inserted into the feature store using a WFS Transaction request.

6.5 Update features with modified geometry and/or property 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.

6.6 Create GML application schema corresponding to known spatial data format

An application schema designer is able create a GML application schema that corresponds to data stored in a file with a well know spatial data format. This application schema can then be used to validate GML instance documents generated from the data in source file.

6.7 Create GML application schemas based on a simple UML class diagram

An application schema designer creates a GML application schema based on a UML simple class diagram. A simple UML model is one that defines classes that would correspond to geographic features whose attributes could be mapped to the property types supported by one of the compliance levels defined in this profile.

7 Schema coding patterns common to all compliance levels

7.1 Introduction

This clause contains discussion and schema coding patterns that apply to all compliance levels defined in this profile.

7.2 Schema processing

Compliance to this profile can be viewed from two perspectives: from the perspective of application schema generators and the perspective of application schema consumers.

An application schema generator is any process 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 encode their data and may ignore the rest. For example, a web feature service that only serves point data will never need to output 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 one 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 shall be prepared to deal with all aspects of one or more compliance levels defined in 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 use heterogeneous geometry collections, but it shall be able to recognize this structure in the schema and possibly raise an exception.

7.3 Metadata handling

This profile prohibits use of **gml:metaDataProperty** elements for referencing metadata in instance documents. Instead, application schema designers may declare metadata feature property elements in a GML application schema, for recording metadata associated with features. If the metadata used to describe features is based on an external schema that does not comply with this profile, then such metadata shall be referenced from a complaint application schema. Otherwise, the metadata can be encoded inline in a complaint application schema.

7.4 Identifying the compliance level

This profile defines three compliance levels as described in clause 2. In order for a client to be able to properly interpret a schema, it shall be able to identify the compliance level of the schema. An XML annotation shall be used for this purpose. The annotation shall be placed at the top most nesting level of the schema such that the XPath expression:

```
/xsd:schema/xsd:annotation/xsd:appinfo/gmlsf:ComplianceLevel
```

can be used to determine the compliance level of the application schema.

The following schema fragment shows how this annotation shall be declared in an application schema:

```
1 <xsd:annotation>
2    <xsd:appinfo
source="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gmlsfLevels.xsd"
>
3       <gmlsf:ComplianceLevel>0|1|2</gmlsf:ComplianceLevel>
4
<gmlsf:GMLProfileSchema>http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/
gmlsf[2].xsd</gmlsf:GMLProfileSchema>
5    </xsd:appinfo>
6 </xsd:annotation>
```

In line 2, the **appinfo** element is used to contain the compliance level declaration for the application schema. The source attribute shall point to the gmlsfLevels.xsd schema, where the **ComplianceLevel** element is declared.

Line 3 uses the **gmlsf:ComplianceLevel** element to declare the compliance level of an application schema. The contents of the **ComplianceLevel** element shall be 0 for schemas that comply with level SF-0, 1 for schemas that comply with level SF-1 and 2 for schemas that comply with level SF-2.

Line 4 uses the **gmlsf:GMLProfileSchema** element to specify the URL of the schema location of the XML Schema document describing the GML Profile. Application schemas that declare compliance to level SF-0 and SF-1 shall provide the schema location of a schema file whose contents are the same as the schema in ANNEX D.1 (called *gmlsf.xsd* in this document). Application schemas that declare compliance to level SF-2 shall provide the schema location of a schema file whose contents are the same as the schema in ANNEX D.2 (called *gmlsf2.xsd* in this document).

8 Schema coding patterns for compliance level SF-0

8.1 Introduction

This clause describes a rigid coding pattern for GML application schemas that comply to level SF-0 of this profile. The main motivation behind this pattern is to limit the set of XML-Schema and GML capabilities that can be used to code a GML application schema. This in turn simplifies the task of building clients that can ingest schema documents that comply with this coding pattern and understand the structure of the feature types defined within.

The schema fragments defined in the following sub-clauses shall be combined to create a complete GML application schema that complies to level SF-0 of this profile. The schema fragments shall be structurally encoded exactly as presented in the document. This means that all mandatory elements and attributes presented in the fragment shall 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 or the order in which elements or type definitions are declared. 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 encoding

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"
6   xmlns:gmlsf="http://www.opengis.net/gmlsf"
7   elementFormDefault="qualified"
8   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 application schema document. The value *target_name_space* is a placeholder for the actual namespace identifier. As a GML application schema is revised, all first and second level revision changes shall require a new namespace identifier and thus a new value for the *target_name_space* placeholder.

EXAMPLE To illustrate this requirement, consider a GML application schema which declares that its initial version is 2.3.0, and that the elements in the schema belong to a namespace with the identifier *http://www.someserver.com/…some path…/2.3.0*. If the next revision of the schema changes the version from 2.3.0 to 2.3.1 then the namespace identifier can remain the same. If, however, the next revision of the schema changes the version from 2.3.0 to 2.4.0 then a new namespace identifier must be used for the new revision of the schema (e.g. http*://www.someserver.com/…some path…/2.4.0* for this example).

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

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 complies with 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 compliant schema may also declare the XML-Schema namespace as the default namespace as long as the default namespace has not been previously bound.

NOTE Like GML, the application schema examples in Annex C use no prefix for the XML-Schema namespace (http://www.w3.org/2001/XMLSchema).

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 at it is correctly bound to the GML namespace for prefixes (http://www.opengis.net/gml).

Line 6 declares the mandatory prefix **gmlsf** for the namespace of the **ComplianceLevel element** declared in *gmlsfLevels.xsd*. As described in Subclause 7.6, the **ComplianceLevel** and the **GMLProfileSchema** elements shall be used inside an **appinfo** element to declare the compliance level of an application schema. Although this profile recommends that this prefix be named **gmlsf**, a compliant schema may use any name that conforms to XML-Schema for this prefix as long at it is correctly bound to the namespace *http://www.opengis.net/gmlsf*.

Line 7 sets the default value for the **elementFormDefault** 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 application schema document being generated. The value ‘’99.99.99’’ is a placeholder for any version value, in any format, that has meaning to the entity or organization creating the schema document.

8.3 Importing and including schemas

8.3.1 Importing the GML schema

The following XML fragment imports the GML schema:

```
<xsd:import namespace="http://www.opengis.net/gml"
 schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
```

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

The schema to be imported shall be the complete GML schema, denoted above using the file name *gml.xsd*. The XML Schema describing the GML Profile (i.e. *gmlsf.xsd*) shall be provided in an **appinfo** annotation as specified in clause 7.4.

The value of the **schemaLocation** attribute shall be a URL (absolute or relative) that resolves to the contents of the GML Schema. The schema fragment above shows a URL that points to the physical file, *schemas.opengis.net/gml/3.1.1/base/gml.xsd*, but any URL may be used (e.g. an HTTP GET request that resolves to the appropriate 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 shall still specify the **schemaLocation** attribute.

8.3.2 Importing the compliance levels schema

The following XML fragment imports the schema that declares the **ComplianceLevel** element used to declare the compliance level of an application schema:

```
<xsd:import namespace="http://www.opengis.net/gmlsf"
     schemaLocation="http://…schema repository path…/gmlsfLevels.xsd"/>
```

The value of the **namespace** attribute shall match the **gmlsf** namespace declaration in the root element.

The value of the **schemaLocation** attribute shall be a URL (absolute or relative) that resolves to the contents of the schema file *gmlsfLevels.xsd*. The schema fragment above shows a URL that points to the physical file, *gmlsfLevels.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 shall specify the **schemaLocation** attribute.

8.3.3 Importing and including other application schemas

An application schema compliant with this profile may import or include other application schemas based on this profile as long as the imported schemas have a compliance level declaration that is less than or equal to the importing schema's compliance level and are also compliant with this profile. The only exceptions are those imported schemas required for compliance by this specification (i.e. gml.xsd). The coding pattern for importing is the same as that used in clause 8.3 except that the **namespace** and **schemaLocation** attributes shall be correct for 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 XML documents with nested elements, and thus assists checking of the Object/property pattern by developers and users.

8.4.2 Defining feature collections

The following XML-Schema fragment shows how to define a feature collection in a GML application schema that complies with level SF-0 of this profile:

```
1   <element name="FeatureCollectionName"
2            type="[prefix:]FeatureCollectionNameType"
3            substitutionGroup="gml:_GML"/>
4   <complexType name="FeatureCollectionNameType">
5      <complexContent>
6         <extension base="gml:AbstractFeatureType">
7            <sequence minOccurs="0" maxOccurs="unbounded">
8               <element name="featureMember">
9                  <complexType>
10                    <sequence>
11                        <element ref="gml:_Feature"/>
12                    </sequence>
13                 </complexType>
14              </element>
15           </sequence>
16        </extension>
17     </complexContent>
18  </complexType>
```

The global element of the feature collection is declared in line 1. The value *FeatureCollectionName* is a placeholder for the actual name of the feature collection. The root element of the collection shall be defined to be of type *FeatureCollectionNameType,* and shall be substitutable for the substitution group head gml:_GML.

NOTE 1 This *FeatureCollectionName* element is NOT defined in the gml:_Feature substitutionGroup, so that this feature collection element cannot be included in another feature collection.

Line 4 begins the definition of the XML complex type that defines the feature collection. This complex type shall be an extension of the GML abstract feature type (line 6), and shall contain zero or more featureMember elements. Each of these featureMember elements shall contain one concrete feature element of any feature type that is defined in the gml:_Feature substitutionGroup. This feature collection shall not include any feature property element declarations.

NOTE 2 This *FeatureCollectionNameType* does NOT extend gml:AbstractFeatureCollectionType because that type has been deprecated in GML 3.2. This extension of gml:AbstractFeatureType does not use gml:featureMember because a feature collection using this profile must directly include (not reference) each of the features included. In addition, the gml:featureMember element has been deprecated in GML 3.2. The only feature member container supported by this profile, for compliance level SF-0, shall be [prefix:]featureMember. Use of a feature container corresponding to gml:featureMembers in GML 3.1.1 is not allowed by this specification for compliance level SF-0.

No more than one feature collection shall be defined in each Application Schema. When used, that feature collection shall be the root element in a XML document.

8.4.3 Defining feature types

The following XML-Schema fragment shows how to define a feature type in a GML application schema that complies with level SF-0 of this profile:

```
1  <xsd:element name="FeatureTypeName"
2              type="[prefix:]FeatureTypeNameType"
3              substitutionGroup="gml:_Feature"/>
4
```

```
5  <xsd:complexType name="FeatureTypeNameType">
6     <xsd:complexContent>
7        <xsd:extension base="gml:AbstractFeatureType">
8           <xsd:sequence>
9
10          <!-- … zero or more element declarations as described in 8.4.4 … -->
11
12          </xsd:sequence>
13       </xsd:extension>
14    </xsd:complexContent>
15 </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 shall be defined to be of type FeatureTypeNameType and shall 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 shall be an extension of the GML abstract feature type (line 7) and shall contain zero or more element declarations that represent the properties of the feature type. The use of the sequence construct in lines 8-12 means that the order in which elements appear in an instance document shall match the order in which they are declared in a compliant application schema document. Subclause 8.4.4 describes how to code elements of the various supported content types.

In this profile, the gml:id attribute in the AbstractFeatureType is changed from optional in the GML specification [OGC 03-105r1] to mandatory in this profile.

8.4.4 Feature property encoding

8.4.4.1 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 (many having a different set of supported basic types), this document limits the set of available basic types for compliance level 0 to a smaller subset. The reasoning is that a smaller common set of supported basic data types is likely to be more interoperable.

In addition to the GML geometric property types defined in Table 6, this profile supports the following list of basic data types:

a) Integers (with optional specified value range and number of digits)

c) Measurements, that include a reference to units of measurement

d) Character strings (with optional specified maximum length or fixed length)

e) Date

f) Boolean

g) Binary data

h) URI references to local or remote resources

i) References to other resources

j) Character strings whose values are from a code list

k) Reals (with optional specified number of digits, decimal precision and value range)

The following subclauses specify XML 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, shall be included in an application schema that complies with this profile.

8.4.4.2 Common facets

Table 4 summarizes the common facets that may be used when declaring elements that encode properties in application schemas that comply with level SF-0 of this profile.

Table 4 — Allowed element facets

XML schema element facet name	Applicable data types	Encoding pattern
minInclusive	integer, measurement, date, dateTime, real	<xsd:minInclusive value="*min value*"/>
minExclusive	integer, measurement, date, dateTime, real	<xsd:minExclusive value="*min value*"/>
maxInclusive	integer, measurement, date, dateTime, real	<xsd:maxInclusive value="*max value*"/>
maxExclusive	integer, measurement, date, dateTime, real	<xsd:maxExclusive value="*max value*"/>
Enumeration	integer, measurement, date, dateTime, real , string, URI, reference	<xsd:enumeration value="*value*"/>

8.4.4.3 Multiplicity and null values

The attributes **minOccurs** and **maxOccurs** may be specified to indicate the minimum and maximum number of times that a property element shall 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). For compliance level SF-0, the attribute **minOccurs** can have the values 0 or 1 and **maxOccurs** can only have the value 1.

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

Although XML-Schema [W3C XML-1] allows **minOccurs=maxOccurs=0**, this profile prohibits this combination. In addition, whether or not the value of **minOccurs** or **maxOccurs** can be equal to or greater than 1 depends on the declared compliance level of the application schemas.

Table 5 — 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 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.

8.4.4.4 Declaring elements with integer content

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

```
<xsd:element name="propertyName" [minOccurs="0|1"] [maxOccurs="1"]>
   <xsd:simpleType>
      <xsd:restriction base="xsd:integer">
         <xsd:totalDigits value="nDigits"/>
         <!-- … Optional facets described in sub-clause 8.4.4.2 …-->
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

SHORT FORM: <xsd:element name="propertyName" type="xsd:integer"
                         [minOccurs="0|1"] [maxOccurs="1"]/>
```

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

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

The use of the attributes **minOccurs** and **maxOccurs** is described in Subclause 8.4.4.2.

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 of more of the optional facets described in sub-clause 8.4.4.2. If no optional facets are specified in the element declaration, then the *short form* schema fragment may be used instead.

8.4.4.5 Declaring elements with content that is a measurement

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

```
<xsd:element name="propertyName" [minOccurs="0|1"] [maxOccurs="1"]>
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:restriction base="gml:MeasureType">
            <!-- … Optional facets described in sub-clause 8.4.4.2 … -->
         </xsd:restriction>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>

SHORT FORM: <xsd:element name="propertyName" type="gml:MeasureType"
                         [minOccurs="0|1"] [maxOccurs="1"]/>
```

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

The use of the attributes **minOccurs** and **maxOccurs** is described in Subclause 8.4.4.2.

A measurement property shall be of type **gml:MeasureType**.

The declaration of an element with a measurement may also include one of more of the optional facets described in sub-clause 8.4.4.2. If no optional facets are specified in the element declaration, then the *short form* schema fragment may be used instead.

In an instance document, if the unit of measure is not known, then the value of the **uom** attribute may be set to **unknown** to indicate this. It should also be noted that if the value of a property is a ratio, then the unit type is **scale** with example values such as unity, percent and parts-per-million.

8.4.4.6 Declaring elements with string content

The following XML-Schema fragments show how to declare a property element with character string content:

```
<xsd:element name="propertyName" [minOccurs="0|1"] [maxOccurs="1"]>
   <xsd:simpleType>
      <xsd:restriction base="xsd:string">
         <xsd:maxLength value="nCharacters"/>
         <!-- - or - -->
         <xsd:length value="nCharacters"/>
         <!--… Optional facets described in sub-clause 8.4.4.2 … -->
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

SHORT FORM: <xsd:element name="propertyName" type="xsd:string"
                         [minOccurs="0|1"] [maxOccurs="1"]/>
```

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

The use of the attributes **minOccurs** and **maxOccurs** is described in Subclause 8.4.4.2.

A character string valued property shall 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. Similarly, the *nCharacters* value of the **value** attribute of the **length** element may be used to indicate the fixed number of characters that the string shall contain. Note that *nCharacters* value may **not** be the same as the length in bytes of the character string. The length of the string in bytes may be longer than the *nCharacters* value if a multi-byte character set is being used.

The declaration of an element with character string content may also include the optional "Enumeration" facet described in sub-clause 8.4.4.2. If no optional facets are specified in the element declaration, then the *short form* schema fragment may be used instead.

8.4.4.7 Declaring elements with date content

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

```
<xsd:element name="propertyName" [minOccurs="0|1"] [maxOccurs="1"]>
   <xsd:simpleType>
      <xsd:restriction base="xsd:date|xsd:dateTime">
         <!-- … Optional facets described in sub-clause 8.4.4.2 … -->
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

SHORT FORM: <xsd:element name="propertyName" type="xsd:date | xsd:dateTime"
                             [minOccurs="0|1"] [maxOccurs="1"]/>
```

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

The use of the attributes **minOccurs** and **maxOccurs** is described in Subclause 8.4.4.2.

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 shall be encoded according to the ISO8601 standard.

The declaration of an element with date content may also include one of more of the optional facets described in sub-clause 8.4.4.2. If no optional facets are specified in the element declaration, then the *short form* schema fragment may be used instead.

8.4.4.8 Declaring elements with boolean content

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

```
<xsd:element name="propertyName"
             type="xsd:boolean"
             [minOccurs="0|1"] [maxOccurs="1"]/>
```

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

The use of the attributes **minOccurs** and **maxOccurs** is described in Subclause 8.4.4.2.

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

8.4.4.9 Declaring elements with binary content

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

```
<xsd:element name="propertyName" [minOccurs="0|1"] [maxOccurs="1"]>
   <xsd:complexType>
      <xsd:simpleContent>
         <xsd:extension base="xsd:base64Binary|xsd:hexBinary">
            <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:attribute name="length"   type="xsd:positiveInteger" use="optional"/>
         </xsd:extension>
      </xsd:simpleContent>
   </xsd:complexType>
</xsd:element>
```

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

The use of the attributes **minOccurs** and **maxOccurs** is described in Subclause 8.4.4.2.

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 shall be used to point to the location of the data. The **mimeType** attribute shall 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 shall 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. Binary data included inline 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.

8.4.4.10 Declaring elements with geometric content

8.4.4.10.1 Encoding properties with geometric content

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

```
<xsd:element name="propertyName"
             type="gml_geometric_propertyType"
             [minOccurs="0|1"] [maxOccurs="1"]>
```

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

The use of the attributes **minOccurs** and **maxOccurs** is described in Subclause 8.4.4.2.

The value *gml_geometric_propertyType* for the **type** attribute is a placeholder for one of the geometric property types defined in Table 6 below.

8.4.4.10.2 Supported geometric property types

Table 6 lists the GML geometry property types can be used in a GML application schema that complies with level SF-0:

Table 6 — Supported GML Geometric Property Types

GML geometric property type	Defined in GML schema document	Restrictions
gml:PointPropertyType	geometryBasic0d1d.xsd	None
gml:CurvePropertyType	geometryBasic0d1d.xsd	Supported property values are gml:LineString or gml:Curve with gml:LineStringSegment segments.
gml:SurfacePropertyType	geometryBasic2d.xsd	Supported property values are gml:Polygon or gml:Surface with gml:PolygonPatch patches.
gml:GeometryPropertyType	geometryBasic0d1d.xsd	Supported property values are gml:Point,gml:LineString, gml:Curve, gml:Polygon, gml:Surface, gml:MultiPoint, gml:MultiCurve, gml:MultiSurface
gml:MultiPointPropertyType	geometryAggregates.xsd	None
gml:MultiCurvePropertyType	geometryAggregates.xsd	Supported members are same as CurvePropertyType
gml:MultiSurfacePropertyType	geometryAggregates.xsd	Supported members are same as SurfacePropertyType
gml:MultiGeometryPropertyType	geometryAggregates.xsd	Supported members are same as GeometryPropertyType

Additional requirements for Geometry include:

a) in all cases, geometry coordinates shall only be specified using the **gml:pos** for **gml:Point** or **gml:posList** elements for all other types

l) coordinate reference systems can have 1, 2 or 3 dimensions

m) features may have any number of geometric properties

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

For the aggregate geometric property types **gml:MultiPointPropertyType**, **gml:MultiCurvePropertyType**, **gml:MultiSurfacePropertyType** and **gml:MultiGeometryPropertyType**, the GML specification defines two containing elements for individual geometry members. The following table lists which geometry member containers are defined for each aggregate geometric type in this profile:

Table 7 — Geometry member containers

Aggregate geometry property type	Geometry member container element
Gml:MultiPointPropertyType	gml:pointMember
Gml:MultiCurvePropertyType	gml:curveMember
Gml:MultiSurfacePropertyType	gml:surfaceMember
Gml:MultiGeometryPropertyType	gml:geometryMember

That is, this profile restricts instance documents to using the property containers **gml:pointMember**, **gml:curveMember**, **gml:surfaceMember** and **gml:geometryMember**.

8.4.4.10.3 CRS support

> This subclause discusses topics related to CRS handling within GML instance documents that validate against application schemas that comply with compliance level SF-0.

A client application needs to 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.

In this profile, the **srsName** attribute can be specified at three different levels in an instance document:

a) On the **gml:boundedBy** Envelope of a top level response container that is a feature collection. For example, the WFS response container, **wfs:FeatureCollection** defined in the Web Feature Service Implementation specification [OGC 04-094].

n) On the **gml:boundedBy** Envelope of each feature instance. The “object” level element of each feature type is defined using the pattern described in subclause 8.4.

o) On the geometry elements **gml:Point, gml:LineString, gml:Curve, gml:Polygon, gml:Surface, gml:Geometry, gml:MultiPoint, gml:MultiCurve, gml:MultiSurface, gml:MultiGeometry**.

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

```
<wfs:FeatureCollection>
   <gml:boundedBy>
      <gml:Envelope srsName="…"> … </gml:Envelope>
   </gml:boundedBy>
   <!-- … feature instances go here … -->
   <gml:featureMember>
      <myns:MyFeature>
         <gml:boundedBy>
            <gml:Envelope srsName="…"> … </gml:Envelope>
         </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. In the case were a CRS is not explicitly specified for a geometry instance, the CRS of the geometry instance shall be assumed to take on the value of the **srsName** attribute of the **gml:boundedBy** Envelope of the nearest containing element (i.e. feature instance or feature collection) that contains an Envelope. 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 shall be specified for each geometry instance of the corresponding feature type even if the CRS is the same for all geometry instances.

8.4.4.11 Declaring elements whose content is a URI

The following XML-Schema fragment shows how to declare a property element with URI content in a GML application schema that complies with this profile:

```
<xsd:element name="propertyName"
             type="xsd:anyURI"
             [minOccurs="0|1"] [maxOccurs="1"]/>
```

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

The use of the attributes **minOccurs** and **maxOccurs** is described in Subclause 8.4.4.2.

8.4.4.12 Declaring 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:

```
<xsd:element name="propertyName" type="gml:ReferenceType" minOccurs="0|1">
   <xsd:annotation>
      <xsd:appinfo source="urn:x-gml:targetElement">…target element…</xsd:appinfo>
    </xsd:annotation>
 </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 Subclause 8.4.4.2.

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

To explicitly describe that an association exists between two features, an XML Schema annotation shall be used, as shown above, to specify the namespace-qualified elements in each feature that encode the association. For example, consider the following association between two fictitious features called AIRPORT and BUILDING:

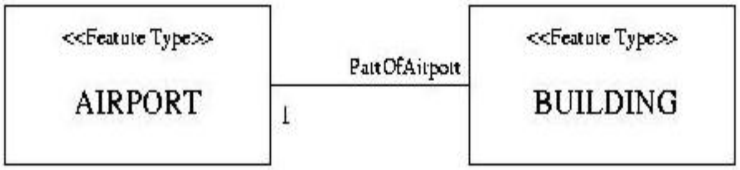

Figure 2 – Example association between two features

In this example, the BUILDING feature might contain the property declaration:

```
<xsd:element name="partOfAirport" type="gml:ReferenceType minOccurs="1">
   <xsd:annotation>
      <xsd:appinfo source="urn:x-gml:targetElement">myns:AIRPORT</xsd:appinfo>
   </xsd:annotation>
</xsd:element>
```

used to indicate in an instance of the BUILDING object to which airport the building belongs.

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
```

8.4.4.13 Declaring elements with string content from a code list

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

```
1 <xsd:element name="propertyName" [minOccurs="0|1"] [maxOccurs="1"]>
2    <xsd:complexType>
3       <xsd:simpleContent>
4          <xsd:restriction base="gml:CodeType">
5             <xsd:attribute name="codeSpace" type="xsd:anyURI" [use="optional"]
6                         default="…some uri…" | fixed="… some uri…"/>
7          </xsd:restriction>
8       <xsd:simpleContent>
9    </xsd:complexType>
10 </xsd:element>
```

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

The use of the attributes **minOccurs** and **maxOccurs** is described in Subclause 8.4.4.2.

A default GML dictionary for the code list can be specified as shown in line 5 using the **default** or **fixed** attribute on the **attribute** declaration pointing to the desired dictionary. If the desired value dictionary is specified in an application schema using the **default** attribute then an instance document may point to an alternate value dictionary at runtime using the **codeSpace** attribute. If, however, the **fixed** attribute is used in an application schema, then an instance document shall use values from the specified dictionary.

Instance documents that validate against application schemas that comply with this profile shall only reference GML dictionaries that comply with the GML 3.1.1 simple dictionary profile [OGC 05-099r1].

8.4.4.14 Declaring elements with real content

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

```
<xsd:element name="propertyName" [minOccurs="0|1"] [maxOccurs="1"]>
   <xsd:simpleType>
      <xsd:restriction base="xsd:double|xsd:decimal">
         <!-- … Optional facets described in sub-clause 8.4.4.2 … -->
         [<xsd:totalDigits value="N"/>]
         [<xsd:fractionDigits value="N" />]
      </xsd:restriction>
   </xsd:simpleType>
</xsd:element>

SHORT FORM: <xsd:element name="propertyName" type="xsd:double"
                         [minOccurs="0|1"] [maxOccurs="0|1"]/>
```

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

The use of the attributes **minOccurs** and **maxOccurs** is described in Subclause 8.4.4.2.

A real-valued property (or an element with real content) shall be of type **xsd:double** or type **xsd:decimal**. The type **xsd:decimal** is included in the profile for the encoding of real values and to support the mapping, into GML, of source formats where the precision and scale are precisely known. For example, when encoding numeric values from a table in a relational database, the definition of numeric columns can include both a precision and scale.

If the type is **xsd:decimal** then the facets *totalDigits* and *fractionDigits* listed in lines 5 and 6 shall be specified. If the type is **xsd:double**, then the facets *totalDigits* and *fractionDigits* listed in lines 5 and 6 shall not be specified.

The declaration of an element with real content may also include one of more of the optional facets described in sub-clause 8.4.4.2. If no optional facets are specified in the element declaration, then the *short form* schema fragment may be used instead.

In most cases, a real-valued property is used for encoding a measurement of some sort and as such the value has an associated unit of measure. For this reason, this specification strongly recommends that real-valued properties be encoded using the **gml:MeasureType** as described in subclause 8.4.4.5.

8.5 Comments and annotations

Application schemas that comply with level SF-0 of this profile may freely use XML comments and annotations, using the **annotation** element, wherever they are legally allowed by XML-Schema and GML.

9 Schema coding patterns for compliance level SF-1

9.1 Introduction

This clause describes a rigid coding pattern for GML application schemas that comply with level SF-1 of this profile. All the coding rules defined for compliance level SF-0, in clauses 7 and 8, apply to schemas that conform with level SF-1 except for the changes and additions noted in subclauses 9.2 and 9.3.

9.2 Changes from compliance level SF-0

9.2.1 Basic data types

In additions to the basic data types listed in subclause 7.7.4, schemas that comply with compliance level SF-1 of this profile also allow:

k) User defined types composed of elements with spatial and non-spatial scalar types as defined in this profile

Clause 9.3.2 describes how user defined types shall be declared in application schemas that comply with level SF-1.

9.2.2 Cardinality

The maximum cardinality that can be declared in an application schema that complies with level SF-0 is 1. This means that the attribute **minOccurs** can have values of 0 or 1 and the attribute **maxOccurs** can only have the value of 1.

This restriction is relaxed for application schemas that comply with level SF-1 of this profile. The attribute **minOccurs** can have a value of 0 or N, where N is a non-negative integer greater than zero. The attribute **maxOccurs** can have a value of 0, N or *unbounded*, where N is a non-negative integer greater than zero and the special value "*unbounded*" is used to indicate that the element may recur an unlimited number of times in an instance document.

In the XML schemas fragments presented as templates in clause 8, all instance of **minOccurs="0|1"** shall be replaced by **minOccurs="0|N"** and all instances of **maxOccurs="1"** shall be replaced by **maxOccurs="0|N|unbounded"** for compliance level SF-1.

9.2.3 Null values

The manner in which the value of a property is deemed to be null is almost the same for both compliance level SF-0 and compliance level SF-1. For compliance level SF-1, however, the fact that elements can have a cardinality of greater than 1 (i.e. the attributes **minOccurs** and **maxOccurs** can have a value greater than 1) must be taken into account.

Table 8, which is similar to Table 5 in clause 8.4.4.3, summarizes how the various combinations of **minOccurs**, **maxOccurs** and an element's content are interpreted to set the value of a feature property for GML application schemas that comply with level SF-1. The table takes into account the fact that elements can have a maximum cardinality that is greater than 1.

Table 8 — 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 GML instance document**	**Interpretation of feature property value**
0	0	EMPTY	Not allowed by this profile.
0	0	NOT EMPTY	Invalid combination
0	=1 or >1	EMPTY	Property value is interpreted as being null or absent.
0	=1 or >1	NOT EMPTY	Property value is the content of the element.
=1 or >1	0	EMPTY	Invalid according to XML Schema.

=1 or >1	0	NOT EMPTY	Invalid according to XML Schema.
=1 or >1	=1 or >1	EMPTY	Invalid according to XML Schema.
=1 or >1	= 1 or >1	NOT EMPTY	Property value is the content of the element.

9.3 Additions to compliance level SF-0

9.3.1 Introduction

Application schemas that comply with level SF-0 can only declare, with the exception of geometric property types, feature properties that have simple or scalar values. For compliance level SF-1, the restriction is eased and conformant schemas are allowed to declare compound or complex valued feature properties. For example, the following schema fragment:

```
<xsd:complexType name="IdentifierPropertyType">
   <xsd:sequence>
      <xsd:element ref="fw:Identifier"/>
   </xsd:sequence>
</xsd:complexType>
<xsd:element name="Identifier">
   <xsd:complexType>
      <xsd:sequence>
         <xsd:element name="identifier" type="xs:string"/>
         <xsd:element name="idAuthority" type="xs:string"
                      minOccurs="0" maxOccurs="1"/>
         <xsd:element name="description" type="xs:string"
                      minOccurs="0" maxOccurs="1"/>
      </xsd:sequence>
   </xsd:complexType>
</xsd:element>
```

declares a complex feature property called **Identifier** as an element that contains three sub-elements, **identifier**, **idAuthority** and **description**.

9.3.2 Declaring elements with complex content

The following XML-Schema fragment shows how to declare a property element with complex content.

```
<xsd:element name="propertyName"
             type="userDefinedNamePropertyType"
             [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 shall be the name of the feature property being encoded.

The value of the mandatory type attribute, *userDefinedNamePropertyType*, is a placeholder for the name of a user-defined type that shall be declared as described in Subclause 9.3.2.2.

The attributes minOccurs and maxOccurs may be specified and indicate the minimum and maximum number of times that the element shall 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.

9.3.3 Declaring a user defined type

The following XML-Schema fragments shows how to declare a used-defined complex type.

```
1  <xsd:complexType name="userDefinedNamePropertyType">
2     <xsd:sequence>
3        <xsd:element ref="tns:userDefinedName"/>
4     </xsd:sequence>
5  </xsd:complexType>
6
7  <xsd:element name="userDefinedName">
8     <xsd:complexType>
9        <xsd:sequence>
10
11       <!-- …one or more element declarations as described in clause 8.4.4… -->
12
13       </xsd:sequence>
14    </xsd:complexType>
15 </xsd:element>
```

The mandatory attribute name, in line 1, contains the name of the user-defined type. The value userDefinedName is a placeholder the name the user chooses for the type. This type name shall include the suffix PropertyType that is a convention used in GML to indicate that this type may be used to declare feature property elements. In order to adhere to the GML object-property rule, the user-defined type declaration shall reference an element with complex content that contains the desired fields for the complex type. As shown in line 3, the value of the mandatory ref element shall be a qualified element name. The value tns is a placeholder for the namespace of this element. The value userDefinedName is a placeholder for the actual name of the element with complex content. It is recommended that the name of the element match the value userDefinedName used in line 1 as the prefix for the name of the user-defined type.

Line 7 declares an element with complex content. The value of the mandatory name attribute shall match the name of the element referenced in line 3. The comment at line 11 is used to indicate that the element named in line 7 shall contain zero or more element declarations that represent the fields of the complex type. These element declarations shall match the scalar non-spatial or spatial property patterns defined in clause 8. In other words, a complex type cannot contain a field that is itself a complex type (with the exception of the spatial property types declared in Table 6).

The use of the sequence model group, at line 9, means that the order in which fields appear in an instance document shall match the order in which they are declared in a compliant application schema document.

It should be noted that the complex type defined at line 1, can be used to define any number of feature property elements of the same type.

10 Schema coding patterns for compliance level SF-2

10.1 Introduction

Compliance level SF-2 is the least restrictive of the compliance levels defined in this profile and is intended to include all the feature encoding abilities provided by the OpenGIS® Simple Features Specification for SQL, Version 1.1 [1]. Unlike compliance levels SF-0 and SF-1, application schemas that comply with compliance level SF-2 have no restrictions whatsoever placed on non-spatial properties. Spatial properties, however, are limited to the set of geometric property types supported by compliance levels SF-0 and SF-1 with the caveats noted in this clause.

10.2 Changes from compliance levels SF-0 and SF-1

Application schemas that comply with level SF-2 shall comply with all the subclauses in Clause 7 and in Subclauses 8.2 and 8.3, with one exception. In Subclause 8.3.1, instead of importing the gmlsf.xsd schema document in line 2 of the schema fragment, the gmlsf2.xsd schema document must be imported. In addition, application schemas that comply with level SF-2 shall, with one change described in the following paragraphs, comply with subclause 8.4.4.9 that describes which geometric property types are supported and how they are to be encoded in conformant application schemas.

For compliance levels SF-0 and SF-1, the only allowed feature member container is the element featureMember. This restriction is removed for compliance level SF-2. That is, the XML-Schema fragment in Subclause 8.4.2 is replaced by:

```
1   <element name="FeatureCollectionName"
2            type="[prefix:]FeatureCollectionNameType"/>
3            substitutionGroup="gml:_GML"/>
4   <complexType name="FeatureCollectionNameType">
5      <complexContent>
6         <extension base="gml:AbstractFeatureType">
7            <sequence maxOccurs="unbounded">
8               <element name="featureMember" minOccurs="0" maxOccurs="unbounded">
9                  <complexType>
10                    <sequence>
11                        <element ref="gml:_Feature"/>
12                    </sequence>
13                 </complexType>
14              </element>
15              <element name="featureMembers" minOccurs="0">
16                 <complexType>
17                    <sequence>
18                        <element ref="gml:_Feature" maxOccurs="unbounded"/>
19                    </sequence>
20                 </complexType>
21              </element>
22           </sequence>
23        </extension>
24     </complexContent>
25  </complexType>
```

Again, the global element of the feature collection is declared in line 1. The value *FeatureCollectionName* is a placeholder for the actual name of the feature collection. The root element of the collection shall be defined to be of type *FeatureCollectionNameType,* and shall be substitutable for the substitution group head gml:_GML.

Line 4 begins the definition of the XML complex type that defines the feature collection. This complex type shall be an extension of the GML abstract feature type (line 6), and can contain zero or more featureMember elements. Each of these featureMember elements shall contain one concrete feature element of any feature type that is defined in the gml:_Feature substitutionGroup.

Instead of, or in addition to, featureMember elements, this complex type can contain zero or one featureMembers element. This featureMembers element shall contain one or more concrete feature element of any feature type that is defined in the gml:_Feature substitutionGroup. This feature collection shall not include any feature property element declarations.

Again, no more than one feature collection shall be defined in each Application Schema. When used, that feature collection shall be the root element in a XML document.

Annex A
(normative)

Compliance testing

A.1 Introduction (Level SF-0, SF-1, SF-2)

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 compliance. Every effort has been made to ensure that the schema fragments presented in this ANNEX match those in the main body of this profile. If, however, a discrepancy is found then the fragments in the main body of this profile should be considered normative.

A.2 Elements to ignore (Level SF-0, SF-1, SF-2)

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 compliance rules (Level SF-0, SF-1, SF-2)

Only the elements and attributes defined or discussed in this profile shall 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 (Level SF-0, SF-1, SF-2)

A compliant application schema shall be a valid GML application schema, which means that is shall 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:xmlns_prefix="http://www.w3.org/2001/XMLSchema"
5    xmlns:gml="http://www.opengis.net/gml"
6    xmlns:gmlsf="http://www.opengis.net/gmlsf"
7    elementFormDefault="qualified"
8    version="99.99.99">
```

Compliance rules:

1) the root element shall be **xsd:schema** (line 1)
    - a) the attribute **targetNamespace** shall be present (line 2)
        - i) its value is user defined
    - b) the attribute **xmlns:*prefix*** shall be present (line 3)
        - i) the value of ***prefix*** is user defined and represents the target namespace
        - ii) the namespace value shall be same as target namespace(line 2)
    - c) the attribute **xmlns:xmls_prefix** shall be present (line 4)
        - i) the value ***xmls_prefix*** is user defined and is the alias for the XML-Schema namespace; this profile uses **xsd** as an example
        - ii) its value shall be 'http://www.w3.org/2001/XMLSchema'
    - d) the attribute **xmlns:gml** shall be present (line 5)
        - i) its value shall be 'http://www.opengis.net/gml'
        - ii) the namespace alias **gml** is recommended but may be redefined by the user; this profile uses the prefix **gml** as an example
    - e) the attribute **xmlns:gmlsf** shall be present (line 6)
        - i) its value shall be ‘http://www.opengis.net/gmlsf’
        - **ii)** the namespace alias shall be **gmlsf**
    - f) additional **xmlns:*prefix*** attributes may be present declaring prefixes for imported namespaces
    - g) the attribute **elementFormDefault** should be present

        i) its value shall be 'qualified'

    h) the attribute **version** shall be present

        i) its value is user defined

A.5 Identifying the compliance level (Level SF-0, SF-1, SF-2)

An application schema shall use an appinfo annotation to declare its compliance level (defined in sub-clause 7.4).

```
1 <xsd:annotation>
2    <xsd:appinfo source="…SOME URI…/gmlsfLevels.xsd">
3       <gmlsf:ComplianceLevel>0|1|2<gmlsf:ComplianceLevel>
4 
<gmlsf:GMLProfileSchema>http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/
gmlsf[2].xsd</gmlsf:GMLProfileSchema>
5    </xsd:appinfo>
6 </xsd:annotation>
```

1) the element **xsd:annotation** shall be present (line 1)
    a) the element **xsd:annotation** shall be present (line 2)
        i. the attribute **source** shall be present (line 2)
        ii. its value shall be some URI that resolves to the contents of *gmlsfLevels.xsd* (line 2)
        iii. the element **gmlsf:ComplianceLevel** shall be present (line 3)
            1. its value shall be one of *0,1 or 2* (line 3)
                a. the value shall be 0 for application schema that claim compliance with level SF-0
                b. the value shall be 1 for application schemas that claim compliance with level SF-1
                c. the value value be 2 for application schemas that claim compliance with level SF-2
        iv. the element **gmlsf:GMLProfileSchema** shall be present (line 4)
            1. its value shall be a valid URL reference to *gmlsf.xsd* for application schemas that comply with levels 0 and 1
            2. its value shall be a valid URL reference to *gmlsf2.xsd* for application schemas that comply with level 2 (line 4)

A.6 Importing the profile schemas (Level SF-0, SF-1, SF-2)

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

This implies that a compliant application schema shall import the GML profile for simple features schema (i.e. gmlsf.xsd or gmlsf2.xsd). The following element shall appear in a compliant application schema document:

```
<xsd:import
   namespace="http://www.opengis.net/gml"
   schemaLocation="http://… SOME URI…/[gml.xsd|gmlsf.xsd|gmlsf2.xsd]"/>
```

Compliance rules:

1) an **xsd:import** element shall be present (line 1)
    a) the attribute **namespace** shall be present (line 2)
        i) its value shall he 'http://www.opengis.net/gml'
    b) the attribute **schemaLocation** shall be present (line 3)
        i) its value shall be a valid URL to either the complete GML Schema or the profile schemas gmlsf.xsd or gmlsf2.xsd

A.7 Importing and including other schemas (Level SF-0, SF-1, SF-2)

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.6

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

Compliance rules:

1) an **xsd:import** element may be present (line 1)
    a) the attribute **namespace** shall be present (line 2)
        i) its value is user defined
    b) The attribute **schemaLocation** shall be present (line 2)
        i) its value shall be a valid URI reference to the imported schema

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

Compliance rules:

2) one or more **xsd:include** elements may be present (line 1)
    a) For each, the attribute **schemaLocation** shall be present (line 2)
        i) its value shall be a valid URI reference to the included schema

A.8 Feature collections (Level SF-0, SF-1)

A.8.1 Element declaration

A compliant application schema shall 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   <element name="FeatureCollectionName"
2            type="[prefix:]FeatureCollectionNameType"
3            substitutionGroup="gml:_GML"/>
```

Compliance rules:

1) an **xsd:element** element shall be present for each feature collection defined in the application schema (line 1)
   a) the attribute **name** shall be present (line 1)
      i) its value is user defined and represents the name of the feature collection
   b) the attribute **type** shall be present (line 2)
      i) its value shall be the name of a complex type defined elsewhere in the document that substitutes for **gml:AbstractFeatureCollectionType**
      ii) the value shall follow the following pattern: 'prefix:FeatureCollectionNameType'
      iii) the prefix shall match the target namespace prefix defined in the root element of the schema document
      iv) the *FeatureCollectionName* is the same as the value of the name attribute
      v) the suffix shall be the literal 'Type'
   c) the attribute **substitutionGroup** shall be present (line 3)
      i) its value shall be '**gml:_GML**'
   d) the prefix **gml** is recommended but it may be

A.8.2 Complex type

A complex type shall 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="FeatureCollectionNameType">
2     <xsd:complexContent>
3        <xsd:extension base="gml:AbstractFeatureType">
4           <xsd:sequence minOccurs="0" maxOccurs="unbounded">
5              <xsd:element name="featureMember">
6                 <xsd:complexType>
7                    <xsd:sequence>
8                       <xsd:element ref="gml:_Feature"/>
9                    </xsd:sequence>
10                </xsd:complexType>
11             </xsd:element>
12          </xsd:sequence>
13       </xsd:extension>
14    </xsd:complexContent>
15 </xsd:complexType>
```

Compliance rules:

1) a **complexType** element shall be present to define the XML type of each feature type
   a) the attribute **name** shall be present (line 1)
      i) the value shall follow the pattern in validation rule A.6.1(b,ii,iii,iv)
2) the element **xsd:complexContent** shall be present (line 2)
3) the element **xsd:extension** shall be present (line 3)
   a) the attribute **base** shall be present

   b) its value shall be ‘gml:AbstractFeatureType’
   c) the prefix **gml** is recommended but it may be redefined

4) the element **xsd:sequence** must be present
   a) the attribute **minOccurs** must be present
      i) its value must be zero
   b) the attribute **maxOccurs** must be present
      i) its value must be 'unbounded'
5) the element **xsd:element** must be present (line 5)
   a) the attribute **name** must be present (line 5)
      i) its value must be **featureMember** (line 5)
6) the element **xsd:complexType** must be present (line 6)
7) the element **xsd:sequence** must be present (line 7)
8) the element **xsd:element** must be present (line 8)
   a) the attribute **ref** must be present (line 8)
      i) its value must be 'gml:_Feature'

A.9 Feature types (Level SF-0, SF-1, SF-2)

A.9.1 Element declaration

A compliant application schema shall 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"/>
```

Compliance rules:

1) an **xsd:element** element shall be present for each feature type defined in the application schemaa (line 1)
   a) the attribute **name** shall be present (line 1)
      i) its value is user defined and represents the name of the feature type
   b) the attribute **type** shall be present (line 2)
      i) its value shall be the name of a complex type defined elsewhere in the document that substitutes for **gml:AbstractFeatureType**
      ii) the value shall follow the following pattern: 'prefix:FeatureTypeNameType'
      iii) the *prefix* shall 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 shall be the literal 'Type'
   c) the attribute **substitutionGroup** shall be present (line 3)

i) its value shall be 'gml:_Feature'

d) the prefix **gml** is recommended but it may be redefined

A.9.2 Complex type

A complex type shall 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>
11    </xsd:complexContent>
12 </xsd:complexType>
```

Compliance rules:

9) a **complexType** element shall be present to define the XML type of each feature type
   - a) the attribute **name** shall be present (line 1)
     - i) the value shall follow the pattern in validation rule A.9.1(b,ii,iii,iv)

10) the element **xsd:complexContent** shall be present (line 2)

11) the element **xsd:extension** shall be present (line 3)
   - a) the attribute **base** shall be present
   - b) its value shall be ‘gml:AbstractFeatureType’
   - c) the prefix **gml** is recommended but it may be redefined

12) the element **xsd:sequence** shall be present (line 4)

13) zero or more property definitions shall follow (line 6)
   - a) for compliance level SF-0 property definitions are described in clauses 8.4.4.1 through 8.4.4.14
   - b) for compliance level SF-1 property definitions are described in clauses 8.4.4.1 through 8.4.4.14 with the provisos in clauses 9.2 and 9.3
   - c) for compliance level SF-2 any property definitions are allowed following any desired pattern except for spatial properties which must conform to clause 8.4.4.10

A.10 Properties

A.10.1 Integer valued properties (Level SF-0, SF-1)

Each integer-valued property shall be defined using the following XML-Schema fragments:

SHORT FORM

```
1   <xsd:element name-“propertyName” type=”xsd:integer”
2                [minOccurs="0|N"] [maxOccurs="1|N|unbounded"]/>
```

Compliance rules:

1) the element **xsd:element** shall be present (line 1)
    a) the attribute **name** shall be present (line 1)
        i) its value is user defined and represents the name of property
    b) the attribute **type** shall present (line 1)
        i) its value shall be the qualified name **integer**
    c) the attribute **minOccurs** may be present (line 2)
        i) for application schemas that claim compliance to level SF-0
        ii) if it is present, its value shall be 0 or some integer N
            (1) for compliance level 0, N shall equal 1
            (2) for compliance level 1, N shall be >= 1
        iii) 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 shall be 1, or some value N
            (1) for compliance level 0, N shall be 1
            (2) for compliance level 1, N shall be >=1 or the value unbounded

LONG FORM

```
1   <xsd:element name="propertyName"
2                [minOccurs="0|N"] [maxOccurs="1|N|unbounded"]>
3      <xsd:simpleType>
4         <xsd:restriction base="xsd:integer">
5            [<xsd:totalDigits value="nDigits"/>]
6            … Optional facets describe in sub-clause 8.4.4.1 …
7         </xsd:restriction>
8      </xsd:simpleType>
9   </xsd:element>
```

Compliance rules:

2) the element **xsd:element** shall be present (line 1)
    a) the attribute **name** shall be present (line 1)
        i) it 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 shall be 0 or some integer N
            (1) for compliance level 0, N shall be 1
            (2) for compliance level 1, N shall be >= 1
        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 shall be 1 or some integer N
            (1) for compliance level 0, N shall be 1

(2) for compliance level 1, N shall be >=1 or the value *unbounded*

ii) if it is not present, the default value is 1

3) the element **xsd:simpleType** shall be present (line 3)

4) the element **xsd:restriction** shall be present (line 4)

a) the attribute **base** shall be present (line 4)

i) its value shall be 'xsd:integer'

5) the element **xsd:totalDigits** may be present (line 5)

a) if **xsd:totalDigits** is present, the attribute **value** shall be present (line 5)

i) its value is user defined and shall represent the number of digits in the integer

6) one or more optional facets (sub-clause 8.4.4.1) may be present (line 6)

a) only the facets defined in sub-clause 8.4.4.1 may be included

b) for any additional facet, only the attributes defined in sub-clause 8.4.4.1 may be present

A.10.2 Real valued properties (Level SF-0, SF-1)

SHORT FORM

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

Compliance rules:

7) the element **xsd:element** shall be present (line 1)

a) the attribute **name** shall be present (line 1)

i) its value is user defined and represents the name of property

b) the attribute **type** shall be present (line 1)

i) its value shall be the qualified name **double**

c) the attribute **minOccurs** may be present (line 2)

i) if it is present, its value shall be 0 or some integer N

(1) for compliance level 0, N shall be 1

(2) for compliance level 1, N shall be >= 1

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 shall be 1 or some integer N

(1) for compliance level 0, N shall be 1

(2) for compliance level 1, N shall be >=1 or *unbounded*

ii) if it not present, the default value is 1

LONG FORM

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

```
3       <xsd:restriction base="xsd:double|xsd:decimal">
4          … Optional facets described in sub-clause 8.4.4.1 …
5          [<xsd:totalDigits value="N"/>]
6          [<xsd:fractionDigits value="N" />]
7       </xsd:restiction>
8    </xsd :simpleType>
9 </xsd:element>
```

Compliance rules:

1) the element **xsd:element** shall be present (line 1)
    a) the attribute **name** shall be present (line 1)
        i) it 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 shall be 0 or some integer N>0
        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 shall 1 or some integer N>1 or *unbounded*
        ii) if it is not present, the default value is 1
2) the element **xsd:simpleType** shall be present (line 3)
3) the element **xsd:restriction** shall be present (line 4)
    a) the attribute **base** shall be present (line 4)
        i) its value shall be **'xsd:decimal'** or '**xsd:double**'
            (1) if the value of the type attribute is **xsd:decimal** then the facets *totalDigits* and *fractionDigits* can 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* sjall not be specified (lines 5 & 6)
4) one or more optional facets (sub-clause 8.4.4.1) may be present (line 7)
    a) only the facets defined in sub-clause 8.4.4.1 may be included
    b) for any additional facet, only the attributes defined in sub-clause 8.4.4.1 may be present

A.10.3 Character string valued properties (Level SF-0, SF-1)

SHORT FORM

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

Compliance rules:

1) the element **xsd:element** shall be present (line 1)
    a) the attribute **name** shall be present (line 1)
        i) its value is user defined and represents the name of property

b) the attribute type shall be present (line 1)

i) its value shall be the qualified name string

c) the attribute minOccurs may be present (line 2)

i) if it is present, its value shall be 0 or some integer N

(1) for compliance level 0, N shall be 1

(2) for compliance level 1, N shall be >=1

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 shall be 1, some value N

(1) for compliance level 0, N shall be >=1 or *unbounded*

LONG FORM

```
1 <xsd:element name="propertyName"
2             [minOccurs="0|N"] [maxOccurs="1|N|unbounded"]>
2    <xsd:simpleType>
3       <xsd:restriction base="xsd:string">
4          <xsd:maxLength value="nCharacters"/> -or- <xsd:length value="nCharacters"/>
5          … Optional facets described in sub-clause 8.4.4.1 …
6       </xsd:restriction>
7    </xsd:simpleType>
8 </xsd:element>
```

Compliance rules:

1) the element **xsd:element** shall be present (line 1)

a) the attribute **name** shall be present (line 1)

i) it 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 shall be 0 or some integer N

(1) for compliance level 0, N shall be 1

(2) for compliance level 1, N shall be >= 1

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 shall 0 or some integer N

(1) for compliance level 0 N shall be 1

(2) for compliance level 1 N shall be >=1 or *unbounded*

ii) if it is not present the default value is 1

2) the element **xsd:simpleType** shall be present (line 3)

3) the element **xsd:restriction** shall be present (line 4)

a) the attribute **base** shall be present (line 4)

i) its value shall be 'xsd:string'

4) the element **xsd:maxLength** OR **xsd:length** shall be present (line 5)

   a) if present, the attribute **value** shall be present (line 5)
      i) its value is user defined and represents the max number of chars that the string can be (**xsd:maxLength**) or the fixed length of the character string (**xsd:length**)
   b) if not present, the max length of the string is undefined

5) one or more optional facets (sub-clause 8.4.4.1) may be present (line 7)
   a) only the facets defined in sub-clause 8.4.4.1 may be included
   b) for any additional facet, only the attributes defined in sub-clause 8.4.4.1 may be present

A.10.4 Date valued properties (Level SF-0, SF-1)

SHORT FORM

```
1 <xsd:element name=”propertyName” type=”xsd:date | xsd:dateTime”
2              [minOccurs=”0|N”] [maxOccurs=”0|N|unbounded”]>
```

1) the element **xsd:element** shall be present (line 1)
   a) the attribute **name** shall be present (line 1)
      i) its value is user defined and represents the name of property
   b) the attribute **type** shall be present (line 1)
      i) its value shall be one of the qualified names **date** or **dateTime**
   c) the attribute **minOccurs** may be present (line 2)
      i) if it is present, its value shall be 0 or some integer N
         (1) for compliance level 0 N shall be 1
         (2) for compliance level 1, N shall be >=1
      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 shall be 1, some value N
         (1) for compliance level 0 N shall be 1
         (2) for compliance level 1 N shall be >=1 or unbounded
      ii) if it is not present, the default value is 1

LONG FORM

```
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 8.4.4.1. …
6       </xsd:restriction>
7    </xsd:simpleType>
8 </xsd:element>
```

Compliance rules:

1) the element **xsd:element** shall be present (line 1)
   a) the attribute **name** shall 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 shall be 0 or some integer N
      ii) if it is not present, the default value shall be 1
   c) the attribute **maxOccurs** may be present (line 2)
      i) if it is present its value shall be 0 or some integer N or *unbounded*
      ii) if it is not present the default value shall be 1
2) the element **xsd:simpleType** shall be present (line 3)
3) the element **xsd:restri**ction shall be present (line 4)
   a) the attribute **base** shall be present (line 15)
      i) its values shall be **xsd:date** or **xsd:dateTime**
4) one or more optional facets (sub-clause 8.4.4.1) may be present (line 7)
   a) only the facets defined in sub-clause 8.4.4.1 may be included
   b) for any additional facet, only the attributes defined in sub-clause 8.4.4.1 may be present

A.10.5 Boolean valued properties (Level SF-0, SF-1)

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

Compliance rules:

1) the element **xsd:element** shall be present (line 1)
   a) the attribute **name** shall be present (line 1)
      i) it value is user defined and represents the name of property
   b) the attribute **type** shall be present (line 1)
      i) it value shall be **xsd:boolean**
   c) the attribute **minOccurs** may be present (line 2)
      i) if it is present, its value shall be 0 or some integer N
         (1) for compliance level 0, N shall be 1
         (2) for compliance level 1, N shall be >=1
      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 shall 0 or some integer N
         (1) for compliance level 0, N shall be 1
         (2) for compliance level 1, N shall be >=1 or the value *unbounded*
      ii) if it is not present, the default value is 1

A.10.6 Binary valued properties (Level SF-0, SF-1)

```
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:attribute name="length"
13                            type="xsd:positiveInteger" use="optional"
12          </xsd:extension>
13       </xsd:simpleContent>
14    </xsd:complexType>
15 </xsd:element>
```

Compliance rules:

1) the element **xsd:element** shall be present (line 1)

a) the attribute **name** shall be present (line 1)

i) it value is user defined and represents the name of property

b) the attribute **minOccurs** , be present (line 2)

i) if it is present, its value shall be 0 or some integer N

(1) for compliance level 0 N shall be 1

(2) for compliance level 1, N shall be >= 1

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 shall 0 or some integer N

(1) for compliance level 1, N shall be >=1 or the value *unbounded*

ii) if it is not present, the default value is 1

2) the element **xsd:complexType** shall be present (line 3)

3) the element **xsd:simpleContent** shall be present (line 4)

4) the element **xsd:extension** shall be present (line 5)

a) the attribute **base** shall be present (line 5)

i) its value shall one or **xsd:base64Binary** or **xsd:hexBinary**

5) the element **xsd:attribute** shall be present (line 6)

a) the attribute **name** shall be present (line 6)

i) its value shall be 'url'

b) the attribute **type** shall be present (line 7)

i) its value shall be 'xsd:anyURI'

c) the attribute **use** shall be present (line 7)

i) its value shall be 'optional'

6) the element **xsd:attribute** shall be present (line 8)

a) the attribute **name** shall be present (line 8)

i) its value shall be 'mimeType'

b) the attribute **type** shall be present (line 9)

i) its value shall be 'xsd:string'

c) the attribute **use** shall be present (line 9)

i) its value shall be 'required'

7) the element **xsd:attribute** shall be present (line 10)

a) the attribute **name** shall be present (line 10)

i) its value shall be 'role'

b) the attribute **type** shall be present (line 11)

i) its value shall be 'xsd:string'

c) the attribute **use** shall be present (line 11)

i) its value shall be 'optional'

8) the element **xsd:attribute** shall be present (line 12)

a) the attibute **name** shall be present (line 12)

i) its value shall be ‘length’

b) the attribute **type** shall be present (line 13)

i) its value shall be ‘positiveInteger’

c) the attribute **use** shall be present (line 13)

i) its value shall be ‘optional’

A.10.7 Geometry valued properties (Level SF-0, SF-1)

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

Compliance rules:

1) the element **xsd:element** shall be present (line 1)

a) the attribute **name** shall be present (line 1)

i) it value is user defined and represents the name of property

b) the attribute **type** shall be present (line 2)

i) it value shall be one of: gml:PointPropertyType, gml:CurvePropertyType, gml:SurfacePropertyType, gml:GeometryPropertyType, gml:MultiPointPropertyType, gml:MultiCurvePropertyType, gml:MultiSurfacePropertyType, gml:MultiGeometryPropertyType,

c) the attribute **minOccurs** may be present (line 3)

i) if it is present, its value shall 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 shall 0 or some integer N or 'unbounded'

ii) if it is not present, the default value is 1

A.10.8 Properties with URI valued content (Level SF-0, SF-1, SF-2)

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

1) the element **xsd:element** shall be present (line 1)

a) the attribute **name** shall be present (line 1)

i. its value is user defined and represents the name of property

b) the attribute **type** shall present (line 1)

i. its value shall be the qualified name **anyURI**

c) the attribute **minOccurs** may be present (line 2)

i. if it is present, its value shall be 0 or some integer N

(1) for compliance level 0, N shall be 1

(2) for compliance level 1, N shall be >=1

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 shall be 1, some value N

(1) for compliance level 0, N shall be 1

(2) for compliance level 1, N shall be >=1 or *unbounded*

A.10.9 Properties with references to other resources (Level SF-0, SF-1)

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

Compliance rules:

1) the element **xsd:element** shall be present (line 1)

a) the attribute **name** shall be present (line 1)

i) its value is user defined and represents the name of property

b) the attribute **type** shall be present (line 1)

i) its value shall be **gml:ReferenceType**

c) the attribute minOccurs may be present (line 3)

i) if it is present, its value shall be 0 or 1

ii) if it is not present, the default value is 1

2) the element **xsd:annotation** shall be present (line 3)

3) the element **xsd:appinfo** shall be present (line 4)

a) the **source** attribute shall be present (line 4)

i) its value shall be *urn:x-gml:targetElement*

b) the content of the **xsd:appinfo** element shall be the fully qualified name of the element being referenced

A.10.10 Properties with string values from a code list (Level 0,1)

```
 1 <xsd:element name="propertyName" [minOccurs="0|N"] [maxOccurs="1|N|unbounded"]>
 2    <xsd:complexType>
 3       <xsd:simpleContent>
 4          <xsd:restriction base="gml:CodeType">
 5             <xsd:attribute name="codeSpace" type="xsd:anyURI"
 6                            [use="optional"] default|fixed="…SOME URI…"/>
 7          </xsd:restriction>
 8       </xsd:simpleContent>
 9    </xsd:complexType>
10 </xsd:element>
```

Compliance rules:

1) the element **xsd:element** shall be present (line 1)
   a) the attribute **name** shall 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 shall be 0 or 1
      ii) if it is not present, the default value is 1
2) the element **xsd:complexType** shall be present (line 2)
3) the element **xsd:simpleContent** shall be present (line 3)
4) the element **xsd:restriction** shall be present (line 4)
   a) the attribute **base** shall be present (line 4)
      i) its value shall be **gml:CodeType**
5) the element **xsd:attribute** shall be present (line 5)
   a) the attribute **name** shall be present (line 5)
      i) its value shall be **codeSpace**
   b) the attribute **type** shall be present (line 5)
      i) its value shall be **xsd:anyURI**
   c) the attribute **use** may be present (line 6)
      i) its value shall be **optional**
      ii) if the use attribute is not present, the default value is **optional**
   d) one of the attributes **default** or **fixed** shall be present (line 6)
      i) its value shall be a URI that resolved to a GML simple dictionary profile document

A.10.11 Measurements (Level SF-0, SF-1)

```
1  <xsd:element name="propertyName"
2                [minOccurs="0|N"] [maxOccurs="0|N|unbounded"]>
3     <xsd:complexType>
4        <xsd:simpleContent>
5           <xsd:restriction base="xsd:double">
```

```
6             … Optional facets described in sub-clause 8.4.4.1 …
7          </xsd:restiction>
8        </xsd :simpleType>
9     </xsd:complexType>
10 </xsd :element>
```

Compliance rules:

1) the element **xsd:element** shall be present (line 1)
    a) the attribute **name** shall be present (line 1)
        i) it value is user defined and represents the name of property
    b) the attribute **minOccurs** may be present (line 2)
        ii) if it is present, its value shall be 0 or some integer N
            (1) for compliance level 0, N shall be 1
            (2) for compliance level 1, N shall be >=1 or the value *unbounded*
        iii) if it is not present, the default value is 1
    c) the attribute **maxOccurs** may be present (line 2)
        iv) if it is present, its value shall be 0 or some integer N
            (1) for compliance level 0, N shall be 1
            (2) for compliance level 1, N shall be >=1 or the value *unbounded*
        v) if it is not present, the default value is 1
2) the element **xsd:complexType** shall be present (line 3)
3) the element **xsd:simpleContent** shall be present (line 4)
4) the element **xsd:restriction** shall be present (line 5)
    a. the attribute **base** shall be present (line 5)
        i) its value shall be **'gml:MeasureType**'
5) one or more optional facets (sub-clause 8.4.4.1) may be present (line 6)
    a. only the facets defined in sub-clause 8.4.4.1 may be included
    b. for any additional facet, only the attributes defined in sub-clause 8.4.4.1 may be present

A.10.12 Complex-value properties (Level SF-1)

A.10.12.1 Introduction

The pattern described here following GML object-property rule, which necessitates the element nesting, described in this clause.

A.10.12.2 User defined property declaration

The following pattern declares a property of a feature of user-defined type:

```
1  <xsd:element name="propertyName"
2               type="userDefinedNamePropertyType"
3               [minOccurs="0|N"] [maxOccurs="0|N|unbounded"]>
```

Compliance rules:

1) the element xsd:element shall be present (line 1)

a) the attribute name shall be present (line 1)

i) it value is user define and represents the name of the property

b) the attribute type shall be present (line 2)

i) its value is used define and shall be the of a type define elsewhere in the application schema

c) the attribute minOccurs may be present (line 3)

i) it if is present, its value shall be 0 or some integer N

(1) for compliance level SF-0, N shall be 1

(2) for compliance level SF-1, N shall be >=1 or the value *unbounded*

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 shall be 0 or some integer N

(1) for compliance level SF-0, N shall be 1

(2) for compliance level SF-1, N shall be >= 1 or the value *unbounded*

A.10.12.3 User-defined complex type declaration

The following XML Schema fragment describes the pattern for defining a user-defined complex type. Following the GML object-property pattern, the user-defined type references an element that declares the properties of the complex type.

```
1  <xsd:complexType name="userDefineNamePropertyType">
2     <xsd:sequence>
3        <xsd:element ref="[prefix:]userDefinedName"/>
4     </xsd:sequence>
5  </xsd:complexType>
6
7  <xsd:element name="userDefinedName">
8     <xsd:complexType>
9        <xsd:sequence>
10
11       <!-- …one or more element declarations as described in clause 8.4.4… -->
12
13       </xsd:sequence>
14    </xsd:complexType>
15 </xsd:element>
```

Compliance rules:

1) the element **xsd:complexType** shall be present (line 1)

a) the attribute **name** shall be present (line 1)

i) the value is used-defined and represents the name of user-defined type

ii) the type name shall have the suffix 'PropertyType'

2) the element **xsd:sequence** shall be present (line 2)

3) the element **xsd:element** shall be present (line 3)

   a) the attribute **ref** shall be present
   b) the value shall reference an element declared elsewhere in the application schema (line 3) that declares the properties of the complex type
4) the element **xsd:element** shall be present (line 7)
   a) the attribute **name** shall be present
   b) its value shall be the name of the element (line 7) referenced in line 3
5) the element **xsd:complexType** shall be present (line 8)
6) the element **xsd:sequence** shall be present (line 9)
7) one or more element declarations as described in clause 8.4.4 shall follow

Annex B
(informative)

Mapping to SQL/MM geometric types

The following table maps all the geometric property types supported by this profile to their equivalent SQL/MM geometry types. The information in Table B.1 was compiled from the mapping presented in Table 14 of the SQL/MM specification [7].

Table B.1 — 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"?>
<schema
   targetNamespace="http://www.cubewerx.com/cw"
   xmlns:cw="http://www.cubewerx.com/cw"
   xmlns="http://www.w3.org/2001/XMLSchema"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:gmlsf="http://www.opengis.net/gmlsf"
   elementFormDefault="qualified"
   version="1.0">

   <annotation>
      <appinfo
source="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gmlsfLevels.xsd"
>
         <gmlsf:ComplianceLevel>0</gmlsf:ComplianceLevel>

<gmlsf:GMLProfileSchema>http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/
gmlsf.xsd</gmlsf:GMLProfileSchema>
      </appinfo>
   </annotation>

   <!-- ============================================================= -->
   <!-- === includes and imports                                  === -->
   <!-- ============================================================= -->
   <import namespace="http://www.opengis.net/gml"
           schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
   <import namespace="http://www.opengis.net/gmlsf"

schemaLocation="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gmlsfLev
els.xsd"/>

   <!-- =========================================================
        define feature types
        ========================================================= -->
   <element name="Reporter" type="cw:ReporterType"
            substitutionGroup="gml:_Feature"/>
   <!-- ============================================================= -->
   <complexType name="ReporterType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="reporterId">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="9"/>
                     </restriction>
                  </simpleType>
               </element>
               <element name="firstName" minOccurs="0" maxOccurs="1">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="20"/>
                     </restriction>
                  </simpleType>
               </element>
               <element name="lastName" minOccurs="0" maxOccurs="1">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="20"/>
                     </restriction>
                  </simpleType>
               </element>
               <element name="organization" minOccurs="0" maxOccurs="1">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="50"/>
                     </restriction>
                  </simpleType>
               </element>
               <element name="email" minOccurs="0" maxOccurs="1">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="50"/>
                     </restriction>
                  </simpleType>
               </element>
               <element name="age" minOccurs="0" maxOccurs="1">
                  <simpleType>
                     <restriction base="integer">
                        <totalDigits value="10"/>
                     </restriction>
                  </simpleType>
               </element>
               <element name="photo" minOccurs="0" maxOccurs="1">
                  <complexType>
                     <simpleContent>
                        <extension base="base64Binary">
                           <attribute name="url" type="anyURI" use="optional"/>
                           <attribute name="mimeType" type="string"
                              use="required"/>
                           <attribute name="role" type="string" use="optional"/>
                        </extension>
                     </simpleContent>
                  </complexType>
               </element>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================ -->
   <element name="NewsItem" type="cw:NewsItemType"
      substitutionGroup="gml:_Feature"/>
   <!-- ============================================================ -->
   <complexType name="NewsItemType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="location" type="gml:PointPropertyType"
                  minOccurs="1" maxOccurs="1"/>
               <element name="reporterId" minOccurs="1" maxOccurs="1">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="9"/>
                     </restriction>
                  </simpleType>
               </element>
               <element name="eventDate" type="dateTime" minOccurs="1"
                  maxOccurs="1"/>
               <element name="byLine" minOccurs="1" maxOccurs="1">
                  <simpleType>
                     <restriction base="string">
```

```
                        <maxLength value="30"/>
                     </restriction>
                  </simpleType>
               </element>
               <element name="details" minOccurs="1" maxOccurs="1">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="20000"/>
                     </restriction>
                  </simpleType>
               </element>
               <element name="image" minOccurs="0" maxOccurs="5">
                  <complexType>
                     <simpleContent>
                        <extension base="base64Binary">
                           <attribute name="url" type="anyURI" use="optional"/>
                           <attribute name="mimeType" type="string"
                              use="required"/>
                           <attribute name="role" type="string" use="optional"/>
                        </extension>
                     </simpleContent>
                  </complexType>
               </element>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================= -->
   <!-- Define a feature collection for a set of news items            -->
   <!-- ============================================================= -->
   <element name="NewsItems"
            type="cw:NewsItemsType"
            substitutionGroup="gml:_GML"/>
   <complexType name="NewsItemsType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence minOccurs="1" maxOccurs="unbounded">
               <element name="featureMember">
                  <complexType>
                     <sequence>
                        <element ref="gml:_Feature"/>
                     </sequence>
                  </complexType>
               </element>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
</schema>
```

C.2 Roads_bts example from the CIPI1.2 testbed

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<schema
   targetNamespace="http://www.opengis.org/cipi1.2/level0/bts"
   xmlns:bts="http://www.opengis.org/cipi1.2/level0/bts"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:gmlsf="http://www.opengis.net/gmlsf"
   xmlns="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified"
   version="1.0">
   <annotation>
      <appinfo
source="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gmlsfLevels.xsd"
>
      <gmlsf:ComplianceLevel>0</gmlsf:ComplianceLevel>

<gmlsf:GMLProfileSchema>http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/
gmlsf.xsd</gmlsf:GMLProfileSchema>
      </appinfo>
   </annotation>
   <!-- ============================================================= -->
   <!-- === includes and imports                                  === -->
   <!-- ============================================================= -->
   <import namespace="http://www.opengis.net/gml"
           schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
   <import namespace="http://www.opengis.net/gmlsf"

schemaLocation="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gmlsfLev
els.xsd"/>
   <!-- =========================================================
        define feature types
        ========================================================= -->
   <element name="Roads_bts" type="bts:Roads_btsType"
            substitutionGroup="gml:_Feature"/>
   <!-- ============================================================= -->
   <complexType name="Roads_btsType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element maxOccurs="1" minOccurs="1" name="Objectid_1">
                  <simpleType>
                     <restriction base="integer">
                        <totalDigits value="10"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="1" name="Objectid">
                  <simpleType>
                     <restriction base="integer">
                        <totalDigits value="10"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="1" name="FNode_">
                  <simpleType>
                     <restriction base="integer">
                        <totalDigits value="10"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="1" name="TNode_">
                  <simpleType>
                     <restriction base="integer">
                        <totalDigits value="10"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="0" name="LPoly_">
                  <simpleType>
                     <restriction base="integer">
                        <totalDigits value="10"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="0" name="RPoly_">
                  <simpleType>
                     <restriction base="integer">
                        <totalDigits value="10"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="0" name="Length"
                  type="gml:MeasureType"/>
               <element maxOccurs="1" minOccurs="1" name="Bdt_roads_">
                  <simpleType>
                     <restriction base="integer">
                        <totalDigits value="10"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="1" name="Bdt_roads1">
                  <simpleType>
                     <restriction base="integer">
                        <totalDigits value="10"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="0" name="Prefix">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="2"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="0" name="Name">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="30"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="0" name="Type">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="4"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="0" name="Suffix">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="2"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="1" name="Fcc">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="3"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="1" name="Fips">
                  <simpleType>
                     <restriction base="string">
                        <maxLength value="11"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="0" name="Shape_len">
                  <simpleType>
                     <restriction base="decimal">
                        <totalDigits value="30"/>
                        <fractionDigits value="15"/>
                     </restriction>
                  </simpleType>
               </element>
               <element maxOccurs="1" minOccurs="0" name="Geometry"
                  type="gml:CurvePropertyType"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================= -->
   <!-- Define a feature collection for a set of Roads_bts            -->
   <!-- ============================================================= -->
   <element name="Roads"
            type="bts:RoadsType"
            substitutionGroup="gml:_GML"/>
   <complexType name="RoadsType">
      <complexContent>
```

```
         <extension base="gml:AbstractFeatureType">
            <sequence minOccurs="1" maxOccurs="unbounded">
               <element name="featureMember">
                  <complexType>
                     <sequence>
                        <element ref="gml:_Feature"/>
                     </sequence>
                  </complexType>
               </element>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
</schema>
```

C.3 Hydrography model example

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<schema
   targetNamespace="http://www.fgdc.gov/framework/073004/hydro"
   xmlns:hyd="http://www.fgdc.gov/framework/073004/hydro"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:gmlsf="http://www.opengis.net/gmlsf"
   xmlns="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified"
   version="0.0.4">

   <annotation>
      <appinfo
source="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gmlsfLevels.xsd"
>
         <gmlsf:ComplianceLevel>1</gmlsf:ComplianceLevel>

<gmlsf:GMLProfileSchema>http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/
gmlsf.xsd</gmlsf:GMLProfileSchema>
      </appinfo>
   </annotation>

   <!-- ============================================================= -->
   <!-- === includes and imports                                  === -->
   <!-- ============================================================= -->
   <import namespace="http://www.opengis.net/gml"
           schemaLocation="http://schemas.opengis.net/gml/3.1.1/base/gml.xsd"/>
   <import namespace="http://www.opengis.net/gmlsf"

schemaLocation="http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gmlsfLev
els.xsd"/>

   <!-- ============================================================= -->
   <!-- IDENTIFIER                                                    -->
   <!-- ============================================================= -->
   <complexType name="IdentifierPropertyType">
      <sequence>
         <element ref="hyd:Identifier"/>
      </sequence>
   </complexType>
   <!-- ============================================================= -->
   <element name="Identifier">
      <complexType>
         <sequence>
            <element name="identifier" type="string"/>
            <element name="idAuthority" type="string" minOccurs="0"
               maxOccurs="1"/>
            <element name="description" type="string" minOccurs="0"
               maxOccurs="1"/>
         </sequence>
      </complexType>
   </element>
   <!-- ============================================================= -->
```

```
<!-- EXTERNALRESOURCE                                              -->
<!-- ============================================================= -->
<complexType name="ExternalResourcePropertyType">
   <sequence>
      <element ref="hyd:ExternalResource"/>
   </sequence>
</complexType>
<!-- ============================================================= -->
<element name="ExternalResource">
   <complexType>
      <sequence>
         <element name="url" type="anyURI"/>
         <element name="urlType" minOccurs="0" maxOccurs="1">
            <complexType>
               <simpleContent>
                  <restriction base="gml:CodeType">
                     <attribute name="codeSpace" type="anyURI"
                        default="../Dictionaries/ResourceTypesDictionary.xml"/>
                  </restriction>
               </simpleContent>
            </complexType>
         </element>
         <element name="urlDescription" type="string" minOccurs="0"
            maxOccurs="1"/>
      </sequence>
   </complexType>
</element>
<!-- ============================================================= -->
<!-- RESPONSIBLEPARTY                                              -->
<!-- ============================================================= -->
<complexType name="ResponsiblePartyPropertyType">
   <sequence>
      <element ref="hyd:ResponsibleParty"/>
   </sequence>
</complexType>
<!-- ============================================================= -->
<element name="ResponsibleParty">
   <complexType>
      <sequence>
         <element name="individualName" type="string"/>
         <element name="organizationName" type="string" minOccurs="0"
            maxOccurs="1"/>
         <element name="positionName" type="string" minOccurs="0"
            maxOccurs="1"/>
         <element name="contactInfo" type="string" minOccurs="0"
            maxOccurs="1"/>
         <element name="role" type="string"/>
      </sequence>
   </complexType>
</element>
<!-- ============================================================= -->
<!-- EXTENDEDATTRIBUTE                                             -->
<!-- ============================================================= -->
<complexType name="ExtendedAttributePropertyType">
   <sequence>
      <element ref="hyd:ExtendedAttribute"/>
   </sequence>
</complexType>
<!-- ============================================================= -->
<element name="ExtendedAttribute">
   <complexType>
      <sequence>
         <element name="authority" type="string"/>
         <element name="link" type="hyd:ExternalResourcePropertyType"
            minOccurs="0" maxOccurs="1"/>
         <element name="name" type="string"/>
         <element name="type" minOccurs="0" maxOccurs="1">
            <complexType>
               <simpleContent>
                  <restriction base="gml:CodeType">
                     <attribute name="codeSpace" type="anyURI" use="optional"
```

```
                           default="../Dictionaries/DatatypeDictionary.xml"/>
                     </restriction>
                  </simpleContent>
               </complexType>
            </element>
            <element name="value" type="string"/>
         </sequence>
      </complexType>
   </element>
   <!-- ============================================================= -->
   <!-- NAME                                                          -->
   <!-- ============================================================= -->
   <complexType name="NamePropertyType">
      <sequence>
         <element ref="hyd:Name"/>
      </sequence>
   </complexType>
   <!-- ============================================================= -->
   <element name="Name">
      <complexType>
         <sequence>
            <element name="name" type="string"/>
            <element name="nameId" type="hyd:IdentifierPropertyType"
               minOccurs="0" maxOccurs="1"/>
         </sequence>
      </complexType>
   </element>
   <!-- ============================================================= -->
   <!-- REPRESENTATION                                                -->
   <!-- ============================================================= -->
   <complexType name="RepresentationPropertyType">
      <sequence>
         <element ref="hyd:Representation"/>
      </sequence>
   </complexType>
   <!-- ============================================================= -->
   <element name="Representation">
      <complexType>
         <sequence>
            <element name="representationId" type="hyd:IdentifierPropertyType"
               minOccurs="0" maxOccurs="1"/>
         </sequence>
      </complexType>
   </element>
   <!-- ============================================================= -->
   <!-- MEASUREMENT                                                   -->
   <!-- ============================================================= -->
   <complexType name="MeasurementPropertyType">
      <sequence>
         <element ref="hyd:Measurement"/>
      </sequence>
   </complexType>
   <!-- ============================================================= -->
   <element name="Measurement">
      <complexType>
         <sequence>
            <element name="accuracy" type="string" minOccurs="0" maxOccurs="1"/>
            <element name="reportingOrganization"
               type="hyd:ResponsiblePartyPropertyType"/>
            <element name="units">
               <complexType>
                  <simpleContent>
                     <restriction base="gml:CodeType">
                        <attribute name="codeSpace" type="anyURI" use="optional"
                           default="../Dictionaries/UnitsDictionary.xml"/>
                     </restriction>
                  </simpleContent>
               </complexType>
            </element>
            <element name="value" type="string"/>
         </sequence>
      </complexType>
   </element>
   <!-- ============================================================= -->
   <!-- ============================================================= -->
   <complexType name="ComputedNetworkValuesPropertyType">
      <sequence>
         <element ref="hyd:ComputedNetworkValues"/>
      </sequence>
   </complexType>
   <!-- ============================================================= -->
   <element name="ComputedNetworkValues">
      <complexType>
         <sequence>
            <element name="fromNode" type="integer"/>
            <element name="toNode" type="integer"/>
            <element name="hydrologicSequenceNumber" type="integer"/>
            <element name="startFlag" type="integer"/>
            <element name="terminalFlag" type="integer"/>
            <element name="terminalDrainId" type="integer"/>
            <element name="levelPathId" type="integer"/>
            <element name="arbolateSumKm" type="gml:MeasureType"/>
            <element name="pathLengthKm" type="gml:MeasureType"/>
            <element name="thinner" type="integer"/>
            <element name="divergenceFlag" type="integer"/>
            <element name="drainStreamLevel" type="integer"/>
            <element name="downstreamDrainLevel" type="integer"/>
            <element name="streamOrder" type="integer"/>
            <element name="upstreamLevelPathId" type="integer"/>
            <element name="upstreamHydrologicSequenceNumber" type="integer"/>
            <element name="upstreamMinimumHydrologicSequenceNumber"
               type="integer"/>
            <element name="downstreamLevelPathId" type="integer"/>
            <element name="downstreamDrainCount" type="integer"/>
            <element name="downstreamMinorHydrologicSequenceNumber"
               type="integer"/>
         </sequence>
      </complexType>
   </element>
   <!-- ============================================================= -->
   <!-- HYDRO COLLECTION                                              -->
   <!-- ============================================================= -->
   <element name="HydroCollection" type="hyd:HydroCollectionType"
      substitutionGroup="gml:_Feature"/>
   <!-- ============================================================= -->
   <complexType name="HydroCollectionType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="metadata" type="anyURI"
                        minOccurs="0" maxOccurs="unbounded"/>
               <element name="featureMember" maxOccurs="unbounded">
                  <complexType>
                     <annotation>
                        <appinfo source="urn:x-
gml:targetElement">hyd:HydroComplex/@gml:id</appinfo>
                     </annotation>
                     <sequence>
                        <element ref="gml:_Feature"/>
                     </sequence>
                  </complexType>
               </element>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================= -->
   <!-- HYDRO COMPLEX                                                 -->
   <!-- ============================================================= -->
   <element name="HydroComplex" type="hyd:HydroComplexType"
      substitutionGroup="gml:_Feature"/>
   <!-- ============================================================= -->
   <complexType name="HydroComplexType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="featureId" type="hyd:IdentifierPropertyType"/>
               <element name="linkedResource"
                  type="hyd:ExternalResourcePropertyType" minOccurs="0"
                  maxOccurs="1"/>
               <element name="metadata" type="anyURI" minOccurs="0"
                  maxOccurs="unbounded"/>
               <element name="featureDate" type="date"/>
               <element name="name" type="hyd:NamePropertyType" minOccurs="0"
                  maxOccurs="unbounded"/>
               <element name="measure" type="hyd:MeasurementPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
               <element name="representation"
                  type="hyd:RepresentationPropertyType" minOccurs="0"
                  maxOccurs="unbounded"/>
               <element name="attribute"
                  type="hyd:ExtendedAttributePropertyType" minOccurs="0"
                  maxOccurs="unbounded"/>
               <element name="compositeType" minOccurs="1" maxOccurs="1">
                  <complexType>
                     <simpleContent>
                        <restriction base="gml:CodeType">
                           <attribute name="codeSpace" type="anyURI"
                              use="optional"
                              default="../Dictionaries/CompositeTypeDictionary.xml"/>
                        </restriction>
                     </simpleContent>
                  </complexType>
               </element>
               <element name="element" type="gml:ReferenceType" minOccurs="1"
                  maxOccurs="unbounded">
                  <annotation>
                     <appinfo source="urn:x-
gml:targetElement">hyd:HydroElement/@gml:id</appinfo>
                  </annotation>
               </element>
            </sequence>
         </extension>
      </complexContent>
      <!-- ============================================================= -->
   </complexType>
   <!-- ============================================================= -->
   <!-- HYDRO ELEMENT                                                 -->
   <!-- ============================================================= -->
   <element name="HydroElement" type="hyd:HydroElementType"
      substitutionGroup="gml:_Feature"/>
   <complexType name="HydroElementType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="featureId" type="hyd:IdentifierPropertyType"/>
               <element name="linkedResource"
                  type="hyd:ExternalResourcePropertyType" minOccurs="0"
                  maxOccurs="1"/>
               <element name="metadata" type="anyURI" minOccurs="0"
                  maxOccurs="unbounded"/>
               <element name="featureDate" type="date"/>
               <element name="name" type="hyd:NamePropertyType" minOccurs="0"
                  maxOccurs="unbounded"/>
               <element name="measure" type="hyd:MeasurementPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
               <element name="representation"
                  type="hyd:RepresentationPropertyType" minOccurs="0"
                  maxOccurs="unbounded"/>
               <element name="attribute"
                  type="hyd:ExtendedAttributePropertyType" minOccurs="0"
                  maxOccurs="unbounded"/>
               <!-- ====================================================== -->
               <element name="featureType">
                  <complexType>
                     <simpleContent>
                        <restriction base="gml:CodeType">
                           <attribute name="codeSpace" type="anyURI"
                              use="optional"
                              default="../Dictionaries/HydroFeatureTypeDictionary.xml"/>
                        </restriction>
                     </simpleContent>
                  </complexType>
               </element>
               <element name="featureCode" minOccurs="0" maxOccurs="1">
                  <complexType>
                     <simpleContent>
                        <restriction base="gml:CodeType">
                           <attribute name="codeSpace" type="anyURI"
                              use="optional"

default="../Dictionaries/HydroFeatureCodeTypeDictionary.xml"/>
                        </restriction>
                     </simpleContent>
                  </complexType>
               </element>
               <element name="geometry" type="gml:GeometryPropertyType"
                  minOccurs="1" maxOccurs="1"/>
               <element name="flowDirection" minOccurs="0" maxOccurs="1">
                  <complexType>
                     <simpleContent>
                        <restriction base="gml:CodeType">
                           <attribute name="codeSpace" type="anyURI"
                              use="optional"
                              default="../Dictionaries/FlowCodeDictionary.xml"/>
                        </restriction>
                     </simpleContent>
                  </complexType>
               </element>
               <!-- ====================================================== -->
               <element name="computedNetworkValues"
                  type="hyd:ComputedNetworkValuesPropertyType" minOccurs="0"
                  maxOccurs="1"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================ -->
   <!-- FEATURERELATIONSHIP                                          -->
   <!-- ============================================================ -->
   <element name="FeatureRelationship" type="hyd:FeatureRelationshipType"
      substitutionGroup="gml:_Feature"/>
   <complexType name="FeatureRelationshipType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="sourceFeature" type="gml:ReferenceType">
                  <annotation>
                     <appinfo source="urn:x-
gml:targetElement">hyd:HydroFeature/@gml:id</appinfo>
                  </annotation>
               </element>
               <element name="targetFeature" type="gml:ReferenceType"
                  minOccurs="1" maxOccurs="unbounded">
                  <annotation>
                     <appinfo source="urn:x-
gml:targetElement">hyd:HydroFeature/@gml:id</appinfo>
                  </annotation>
               </element>
               <element name="type">
                  <complexType>
                     <simpleContent>
                        <restriction base="gml:CodeType">
                           <attribute name="codeSpace" type="anyURI"
```

```
                              use="optional"
                              default="../Dictionaries/RelationshipTypeDictionary.xml"/>
                        </restriction>
                     </simpleContent>
                  </complexType>
               </element>
               <element name="attribute"
                  type="hyd:ExtendedAttributePropertyType" minOccurs="0"
                  maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================= -->
   <!-- EVENT                                                         -->
   <!-- ============================================================= -->
   <element name="Event" type="hyd:EventType" substitutionGroup="gml:_Feature"/>
   <complexType name="EventType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="date" type="date" minOccurs="0" maxOccurs="1"/>
               <element name="eventID" type="gml:ReferenceType" minOccurs="1"
                  maxOccurs="1">
                  <annotation>
                     <appinfo source="urn:x-
gml:targetElement">hyd:Identifier/@gml:id</appinfo>
                  </annotation>
               </element>
               <element name="relatedURL" type="anyURI" minOccurs="0"
                  maxOccurs="unbounded"/>
               <element name="metadata" type="anyURI" minOccurs="0"
                  maxOccurs="1"/>
               <element name="eventType">
                  <complexType>
                     <simpleContent>
                        <restriction base="gml:CodeType">
                           <attribute name="codeSpace" type="anyURI"
                              use="optional"
                              default="../Dictionaries/EventTypeDictionary.xml"/>
                        </restriction>
                     </simpleContent>
                  </complexType>
               </element>
               <element name="dataType">
                  <complexType>
                     <simpleContent>
                        <restriction base="gml:CodeType">
                           <attribute name="codeSpace" type="anyURI"
                              use="optional"
                              fixed="../Dictionaries/DatatypeDictionary.xml"/>
                        </restriction>
                     </simpleContent>
                  </complexType>
               </element>
               <element name="eventValue" type="string"/>
               <element name="location" type="gml:GeometryPropertyType"
                  minOccurs="0" maxOccurs="1"/>
               <element name="measure" type="hyd:MeasurementPropertyType"
                  minOccurs="0" maxOccurs="unbounded"/>
               <element name="attribute"
                  type="hyd:ExtendedAttributePropertyType" minOccurs="0"
                  maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================= -->
   <!-- UNMEASUREDEVENT                                               -->
   <!-- ============================================================= -->
   <element name="UnmeasuredEvent" type="hyd:UnmeasuredEventType"
```

```
      substitutionGroup="gml:_Feature"/>
   <complexType name="UnmeasuredEventType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="event" type="gml:ReferenceType">
                  <annotation>
                     <appinfo source="urn:x-
gml:targetElement">hyd:Event/@gml:id</appinfo>
                  </annotation>
               </element>
               <element name="location" type="gml:GeometryPropertyType"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================= -->
   <!-- MEASUREDEVENT                                                 -->
   <!-- ============================================================= -->
   <element name="MeasuredEvent" type="hyd:MeasuredEventType"
      substitutionGroup="gml:_Feature"/>
   <complexType name="MeasuredEventType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="event" type="gml:ReferenceType">
                  <annotation>
                     <appinfo source="urn:x-
gml:targetElement">hyd:Event/@gml:id</appinfo>
                  </annotation>
               </element>
               <element name="startPosition" type="gml:PointPropertyType"/>
               <element name="endPosition" type="gml:PointPropertyType"
                  minOccurs="0" maxOccurs="1"/>
               <element name="startOffset" type="gml:MeasureType" minOccurs="0"
                  maxOccurs="1"/>
               <element name="endOffset" type="gml:MeasureType" minOccurs="0"
                  maxOccurs="1"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================= -->
   <!-- EVENT ON                                                      -->
   <!-- ============================================================= -->
   <element name="EventOn" type="hyd:EventOnType"
      substitutionGroup="gml:_Feature"/>
   <complexType name="EventOnType">
      <complexContent>
         <extension base="gml:AbstractFeatureType">
            <sequence>
               <element name="feature" type="gml:ReferenceType" minOccurs="1"
                  maxOccurs="unbounded">
                  <annotation>
                     <appinfo source="urn:x-
gml:targetElement">hyd:HydroFeature/@gml:id</appinfo>
                  </annotation>
               </element>
               <element name="event" type="gml:ReferenceType" minOccurs="0"
                  maxOccurs="unbounded">
                  <annotation>
                     <appinfo source="urn:x-
gml:targetElement">hyd:Event/@gml:id</appinfo>
                  </annotation>
               </element>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
</schema>
```

Annex D
(normative)

Reference schemas for GML simple features profile

D.1 GML schema for compliance levels SF-0 and SF-1

The following schema document, called *gmlsf.xsd*, contains only those element declarations from GML that are required to validate a GML application schema that complies with compliance levels SF-0 or SF-1 as defined in this profile. GML application schemas can ensure compliance with this profile by importing this schema document rather 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"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified"
   version="1.0.0">

   <annotation>
      <appinfo
source="urn:opengis:specification:gml:schema-xsd:gmlsf:1.0.0">gmlsf.xsd</appinfo>
      <documentation>
         GML 3.1.1 Simplified Features profile Levels 0 and 1.
         Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.
      </documentation>
   </annotation>

   <!-- ============================================================= -->
   <!-- === includes and imports                                  === -->
   <!-- ============================================================= -->
   <import namespace="http://www.w3.org/1999/xlink"
   schemaLocation="http://schemas.opengis.net/xlink/1.0.0/xlinks.xsd"/>

   <!-- =========================================================== -->
   <!-- === Subset of geometryAggregates.xsd for this profile ===== -->
   <!-- =========================================================== -->
   <!-- aggregate geometry objects -->
   <!-- =========================================================== -->
   <element name="_GeometricAggregate"
            type="gml:AbstractGeometricAggregateType"
            substitutionGroup="gml:_Geometry"
            abstract="true">
      <annotation>
         <documentation>
            The "_GeometricAggregate" element is the abstract head of
            the substitution group for all geometric aggregates.
         </documentation>
      </annotation>
   </element>
   <!-- =========================================================== -->
   <complexType name="AbstractGeometricAggregateType" abstract="true">
      <annotation>
         <documentation>
            This is the abstract root type of the geometric aggregates.
         </documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractGeometryType"/>
      </complexContent>
   </complexType>
   <!-- =========================================================== -->
   <complexType name="MultiGeometryPropertyType">
      <annotation>
         <documentation>
            A property that has a geometric aggregate as its value domain
            shall contain an appropriate geometry element encapsulated in
            an element of this type.
         </documentation>
      </annotation>
      <sequence>
         <element ref="gml:_GeometricAggregate"/>
      </sequence>
   </complexType>
   <!-- =========================================================== -->
   <element name="MultiPoint"
            type="gml:MultiPointType"
            substitutionGroup="gml:_GeometricAggregate"/>
   <!-- =========================================================== -->
   <complexType name="MultiPointType">
      <annotation>
         <documentation>
            A MultiPoint is defined by one or more Points, referenced
            through pointMember elements.
         </documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
                <element ref="gml:pointMember"
                        minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- =========================================================== -->
   <complexType name="MultiPointPropertyType">
      <annotation>
         <documentation>
            A property that has a collection of points as its value
            domain shall contain an appropriate geometry element
            encapsulated in an element of this type.
         </documentation>
      </annotation>
      <sequence>
         <element ref="gml:MultiPoint"/>
      </sequence>
   </complexType>
   <!-- =========================================================== -->
   <element name="MultiCurve"
            type="gml:MultiCurveType"
            substitutionGroup="gml:_GeometricAggregate"/>
   <!-- =========================================================== -->
   <complexType name="MultiCurveType">
      <annotation>
         <documentation>
            A MultiCurve is defined by one or more Curves, referenced
            through curveMember elements.
         </documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
               <element ref="gml:curveMember"
                        minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================ -->
   <complexType name="MultiCurvePropertyType">
      <annotation>
         <documentation>
            A property that has a collection of curves as its value domain
            shall contain an appropriate geometry element encapsulated in
            an element of this type.
         </documentation>
      </annotation>
      <sequence>
         <element ref="gml:MultiCurve"/>
      </sequence>
   </complexType>
   <!-- ============================================================ -->
   <element name="MultiSurface"
            type="gml:MultiSurfaceType"
            substitutionGroup="gml:_GeometricAggregate"/>
   <!-- ============================================================ -->
   <complexType name="MultiSurfaceType">
      <annotation>
         <documentation>
            A MultiSurface is defined by one or more Surfaces,
            referenced through surfaceMember elements.
         </documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
               <element ref="gml:surfaceMember"
                        minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- ============================================================ -->
   <complexType name="MultiSurfacePropertyType">
      <annotation>
         <documentation>
            A property that has a collection of surfaces as its value
            domain shall contain an appropriate geometry element
            encapsulated in an element of this type.
         </documentation>
      </annotation>
      <sequence>
         <element ref="gml:MultiSurface"/>
      </sequence>
   </complexType>
   <!-- ============================================================ -->
   <element name="pointMember" type="gml:PointPropertyType">
      <annotation>
         <documentation>
            This property element contains the Point element.
         </documentation>
      </annotation>
   </element>
   <!-- ============================================================ -->
   <element name="surfaceMember" type="gml:SurfacePropertyType">
      <annotation>
         <documentation>
            This property element contains the surface element. A surface
            element is any element which is substitutable for "_Surface".
         </documentation>
      </annotation>
   </element>
   <!-- ============================================================ -->
   <!-- == Subset of geometryPrimitives.xsd for this profile ====== -->
   <!-- ============================================================ -->
   <element name="Curve" type="gml:CurveType" substitutionGroup="gml:_Curve"/>
```

```
<!-- ============================================================ -->
<complexType name="CurveType">
   <annotation>
      <documentation>
         Curve is a 1-dimensional primitive. Curves are continuous,
         connected, and have a measurable length in terms of the
         coordinate system.
         A curve is composed of one or more curve segments.
         The curve segments are connected to one another, with the
         end point of each segment except the last being the start
         point of the next segment in the segment list.
         The orientation of the curve is positive.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractCurveType">
         <sequence>
            <element ref="gml:segments">
               <annotation>
                  <documentation>
                     This element encapsulates the segments of the
                     curve.
                  </documentation>
               </annotation>
            </element>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- ============================================================ -->
<!-- curve segments (1-dimensional) -->
<!-- ============================================================ -->
<element name="_CurveSegment"
         type="gml:AbstractCurveSegmentType"
         abstract="true">
   <annotation>
      <documentation>
         The "_CurveSegment" element is the abstract head of the
         substituition group for all curve segment elements,
         i.e. continuous segments of the same interpolation
         mechanism.
      </documentation>
   </annotation>
</element>
<!-- ============================================================ -->
<complexType name="AbstractCurveSegmentType" abstract="true">
   <annotation>
      <documentation>
         Curve segment defines a homogeneous segment of a curve.
      </documentation>
   </annotation>
   <sequence/>
</complexType>
<!-- ============================================================ -->
<element name="segments" type="gml:CurveSegmentArrayPropertyType">
   <annotation>
      <documentation>
         This property element contains a list of curve segments.
         The order of the elements is significant and shall be
         preserved when processing the array.
      </documentation>
   </annotation>
</element>
<!-- ============================================================ -->
<complexType name="CurveSegmentArrayPropertyType">
   <annotation>
      <documentation>
         A container for an array of curve segments.
      </documentation>
   </annotation>
   <sequence>
```

```
      <element ref="gml:_CurveSegment" minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
</complexType>
<!-- =========================================================== -->
<element name="curveMember" type="gml:CurvePropertyType">
   <annotation>
      <documentation>
         This property element contains the curve element. A curve
         element is any element which is substitutable for "_Curve".
      </documentation>
   </annotation>
</element>
<!-- ============================================================ -->
<!-- === global element in "_CurveSegment" substitution group === -->
<!-- ============================================================ -->
<element name="LineStringSegment"
         type="gml:LineStringSegmentType"
         substitutionGroup="gml:_CurveSegment"/>
<!-- ============================================================ -->
<complexType name="LineStringSegmentType">
   <annotation>
      <documentation>
         A LineStringSegment is a curve segment that is defined by
         two or more coordinate tuples, with linear interpolation
         between them.
         Note: LineStringSegment implements GM_LineString of ISO 19107.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractCurveSegmentType">
         <sequence>
            <element ref="gml:posList"/>
         </sequence>
         <attribute name="interpolation"
                    type="gml:CurveInterpolationType"
                    fixed="linear">
            <annotation>
               <documentation>
                  The attribute "interpolation" specifies
                  the curve interpolation mechanism used
                  for this segment. This mechanism uses
                  the control points and control parameters
                  to determine the position of this curve
                  segment. For a LineStringSegment the
                  interpolation is fixed as "linear".
               </documentation>
            </annotation>
         </attribute>
      </extension>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<!-- surface patches (2-dimensional) -->
<!-- =========================================================== -->
<element name="_SurfacePatch"
         type="gml:AbstractSurfacePatchType"
         abstract="true">
   <annotation>
      <documentation>
         The "_SurfacePatch" element is the abstract head of the
         substituition group for all surface pach elements describing
         a continuous portion of a surface.
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<complexType name="AbstractSurfacePatchType" abstract="true">
   <annotation>
      <documentation>
         A surface patch defines a homogenuous portion of a surface.
      </documentation>
   </annotation>
   <sequence/>
</complexType>
<!-- =========================================================== -->
<element name="patches" type="gml:SurfacePatchArrayPropertyType">
   <annotation>
      <documentation>
          This property element contains a list of surface patches.
          The order of the elements is significant and shall be
          preserved when processing the array.
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<complexType name="SurfacePatchArrayPropertyType">
   <annotation>
      <documentation>
         A container for an array of surface patches.
      </documentation>
   </annotation>
   <sequence>
      <element ref="gml:_SurfacePatch"
               minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
</complexType>
<!-- =========================================================== -->
<element name="PolygonPatch"
         type="gml:PolygonPatchType"
         substitutionGroup="gml:_SurfacePatch"/>
<!-- =========================================================== -->
<complexType name="PolygonPatchType">
   <annotation>
      <documentation>
         A PolygonPatch is a surface patch that is defined by
         a set of boundary curves and an underlying surface to
         which these curves adhere. The curves are coplanar and
         the polygon uses planar interpolation in its interior.
         Implements GM_Polygon of ISO 19107.
      </documentation>
   </annotation>
   <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">
            <annotation>
               <documentation>
                  The attribute "interpolation" specifies the
                  interpolation mechanism used for this surface
                  patch. Currently only planar surface patches
                  are defined in GML 3, the attribute is fixed
                  to "planar", i.e. the interpolation method
                  shall return points on a single plane. The
                  boundary of the patch shall be contained within
                  that plane.
               </documentation>
            </annotation>
         </attribute>
      </extension>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<element name="Surface"
              type="gml:SurfaceType"
              substitutionGroup="gml:_Surface"/>
<!-- =========================================================== -->
```

```
<complexType name="SurfaceType">
   <annotation>
      <documentation>
         A Surface is a 2-dimensional primitive and is composed
         of one or more surface patches. The surface patches are
         connected to one another.
         The orientation of the surface is positive ("up"). The
         orientation of a surface chooses an "up" direction
         through the choice of the upward normal, which, if the
         surface is not a cycle, is the side of the surface from
         which the exterior boundary appears counterclockwise.
         Reversal of the surface orientation reverses the curve
         orientation of each boundary component, and interchanges
         the conceptual "up" and "down" direction of the surface.
         If the surface is the boundary of a solid, the "up"
         direction is usually outward. For closed surfaces, which
         have no boundary, the up direction is that of the surface
         patches, which must be consistent with one another. Its
         included surface patches describe the interior structure
         of the Surface.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractSurfaceType">
         <sequence>
            <element ref="gml:patches">
               <annotation>
                  <documentation>
                     This element encapsulates the patches of the
                     surface.
                  </documentation>
               </annotation>
            </element>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<simpleType name="CurveInterpolationType">
   <annotation>
      <documentation>
         CurveInterpolationType is a list of codes that may be
         used to identify the interpolation mechanisms specified
         by an schema.
      </documentation>
   </annotation>
   <restriction base="string">
      <enumeration value="linear"/>
   </restriction>
</simpleType>
<simpleType name="SurfaceInterpolationType">
   <annotation>
      <documentation>
         SurfaceInterpolationType is a list of codes that may be
         used to identify the interpolation mechanisms specified
         by an application schema.
      </documentation>
   </annotation>
   <restriction base="string">
      <enumeration value="planar"/>
   </restriction>
</simpleType>
<!-- =========================================================== -->
<!-- == Subset of geometryBasic2d.xsd for this profile ========= -->
<!-- =========================================================== -->
<!-- == primitive geometry objects (2-dimensional) ============= -->
<!-- =========================================================== -->
<element name="_Surface"
         type="gml:AbstractSurfaceType"
         substitutionGroup="gml:_GeometricPrimitive"
         abstract="true">
   <annotation>
      <documentation>
         The "_Surface" element is the abstract head of the
         substituition group for all (continuous) surface elements.
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<complexType name="AbstractSurfaceType">
   <annotation>
      <documentation>
         An abstraction of a surface to support the different
         levels of complexity. A surface is always a continuous
         region of a plane.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGeometricPrimitiveType"/>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<complexType name="SurfacePropertyType">
   <annotation>
      <documentation>
         A property that has a surface as its value domain shall contain
         an appropriate geometry element encapsulated in an element
         of this type.
      </documentation>
   </annotation>
   <sequence>
      <element ref="gml:_Surface"/>
   </sequence>
</complexType>
<!-- =========================================================== -->
<element name="Polygon"
         type="gml:PolygonType"
         substitutionGroup="gml:_Surface"/>
<!-- =========================================================== -->
<complexType name="PolygonType">
   <annotation>
      <documentation>
         A Polygon is a special surface that is defined by a single
         surface patch. The boundary of this patch is coplanar and
         the polygon uses planar interpolation in its interior. It
         is backwards compatible with the Polygon of GML 2.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractSurfaceType">
         <sequence>
            <element ref="gml:exterior" minOccurs="0"/>
            <element ref="gml:interior" minOccurs="0" maxOccurs="unbounded"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<!-- rings (closed curves for surface boundaries) -->
<!-- =========================================================== -->
<element name="exterior" type="gml:AbstractRingPropertyType">
   <annotation>
      <documentation>
         A boundary of a surface consists of a number of rings.
         In the normal 2D case, one of these rings is distinguished
         as being the exterior boundary. In a general manifold this
         is not always possible, in which case all boundaries shall
         be listed as interior boundaries, and the exterior will be
         empty.
      </documentation>
   </annotation>
</element>
```

```
<!-- ============================================================ -->
<element name="interior" type="gml:AbstractRingPropertyType">
   <annotation>
      <documentation>
         A boundary of a surface consists of a number of rings. The
         "interior" rings seperate the surface / surface patch from
         the area enclosed by the rings.
      </documentation>
   </annotation>
</element>
<!-- ============================================================ -->
<complexType name="AbstractRingPropertyType">
   <annotation>
      <documentation>
         Encapsulates a ring to represent the surface boundary property
         of a surface.
      </documentation>
   </annotation>
   <sequence>
      <element ref="gml:LinearRing"/>
   </sequence>
</complexType>
<!-- ============================================================ -->
<element name="LinearRing" type="gml:LinearRingType"
   substitutionGroup="gml:_Geometry"/>
<!-- ============================================================ -->
<complexType name="LinearRingType">
   <annotation>
      <documentation>
         A LinearRing is defined by four or more coordinate tuples,
         with linear interpolation between them; the first and last
         coordinates must be coincident.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGeometryType">
         <sequence>
            <element ref="gml:posList">
               <annotation>
                  <documentation>
                     The "posList" element provides a compact way to
                     specifiy the coordinates of the control points,
                     if all control points are in the same coordinate
                     reference systems and belong to this ring only.
                     The number of direct positions in the list must
                     be at least four.
                  </documentation>
               </annotation>
            </element>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- ============================================================ -->
<!-- == Subset of geometryBasic0d1d.xsd for this profile ======= -->
<!-- ============================================================ -->
<!-- ============================================================ -->
<!-- == abstract supertype for geometry objects ================ -->
<!-- ============================================================ -->
<element name="_Geometry" type="gml:AbstractGeometryType"
   substitutionGroup="gml:_GML" abstract="true">
   <annotation>
      <documentation>
         The "_Geometry" element is the abstract head of the
         substituition group for all geometry elements of GML 3.
         This includes pre-defined and user-defined geometry elements.
         Any geometry element must be a direct or indirect
         extension/restriction  of AbstractGeometryType
         and must be directly or indirectly in the substitution
         group of "_Geometry".
      </documentation>
   </annotation>
</element>
<!-- ============================================================== -->
<complexType name="GeometryPropertyType">
   <annotation>
      <documentation>
         A geometric property shall contain any geometry element
         encapsulated in an element of this type.
      </documentation>
   </annotation>
   <sequence>
      <element ref="gml:_Geometry"/>
   </sequence>
</complexType>
<!-- ============================================================== -->
<complexType name="AbstractGeometryType" abstract="true">
   <annotation>
      <documentation>
         All geometry elements are derived directly or indirectly
         from this abstract supertype. A geometry element may
         have an identifying attribute ("gml:id"), a name (attribute
         "name") and a description (attribute "description"). It may
         be associated with a spatial reference system (attribute
         "srsName"). The following rules shall be adhered: - Every
         geometry type shall derive from this abstract type. - Every
         geometry element (i.e. an element of a geometry type) shall
         be directly or indirectly in the substitution group of _Geometry.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGMLType">
          <attribute name="srsName" type="anyURI" use="optional">
              <annotation>
                  <documentation>
         In general this reference points to a CRS instance of
         gml:CoordinateReferenceSystemType (see
         coordinateReferenceSystems.xsd). For well known
         references it is not required that the CRS description
         exists at the location the URI points to. If no srsName
         attribute is given, the CRS must be specified as part of
         the larger context this geometry element is part of, e.g.
         a geometric element like point, curve, etc. It is expected
         that this attribute will be specified at the direct position
         level only in rare cases.
                  </documentation>
              </annotation>
          </attribute>
      </extension>
   </complexContent>
</complexType>
<!-- ============================================================== -->
<element name="_GeometricPrimitive" type="gml:AbstractGeometricPrimitiveType"
   abstract="true" substitutionGroup="gml:_Geometry">
   <annotation>
      <documentation>
         The "_GeometricPrimitive" element is the abstract head of
         the substituition group for all (pre- and user-defined)
         geometric primitives.
      </documentation>
   </annotation>
</element>
<!-- ============================================================== -->
<complexType name="AbstractGeometricPrimitiveType" abstract="true">
   <annotation>
      <documentation>
         This is the abstract root type of the geometric primitives.
         A geometric primitive is a geometric object that is not
         decomposed further into other primitives in the system.
         All primitives are oriented in the direction implied by
         the sequence of their coordinate tuples.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGeometryType"/>
   </complexContent>
</complexType>
<!-- ============================================================ -->
<!-- primitive geometry objects (0-dimensional) -->
<!-- ============================================================ -->
<element name="Point"
         type="gml:PointType"
         substitutionGroup="gml:_GeometricPrimitive"/>
<!-- ============================================================ -->
<complexType name="PointType">
   <annotation>
      <documentation>
         A Point is defined by a single coordinate tuple.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGeometricPrimitiveType">
         <sequence>
            <element ref="gml:pos"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- ============================================================ -->
<complexType name="PointPropertyType">
   <annotation>
      <documentation>
         A property that has a point as its value domain shall contain
         an appropriate geometry element encapsulated in an element
         of this type.
      </documentation>
   </annotation>
   <sequence>
      <element ref="gml:Point"/>
   </sequence>
</complexType>
<!-- ============================================================ -->
<!-- primitive geometry objects (1-dimensional) -->
<!-- ============================================================ -->
<element name="_Curve" type="gml:AbstractCurveType"
   substitutionGroup="gml:_GeometricPrimitive" abstract="true">
   <annotation>
      <documentation>
         The "_Curve" element is the abstract head of the
         substituition group for all (continuous) curve elements.
      </documentation>
   </annotation>
</element>
<!-- ============================================================ -->
<complexType name="AbstractCurveType" abstract="true">
   <annotation>
      <documentation>
         An abstraction of a curve to support the different levels
         of complexity. The curve can always be viewed as a geometric
         primitive, i.e. is continuous.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGeometricPrimitiveType"/>
   </complexContent>
</complexType>
<!-- ============================================================ -->
<complexType name="CurvePropertyType">
   <annotation>
      <documentation>
         A property that has a curve as its value domain shall contain
         an appropriate geometry element encapsulated in an element
         of this type.
      </documentation>
   </annotation>
   <sequence>
      <element ref="gml:_Curve"/>
   </sequence>
</complexType>
<!-- =========================================================== -->
<element name="LineString"
         type="gml:LineStringType"
         substitutionGroup="gml:_Curve"/>
<!-- =========================================================== -->
<complexType name="LineStringType">
   <annotation>
      <documentation>
         A LineString is a special curve that consists of a
         single segment with linear interpolation. It is defined
         by two or more coordinate tuples, with linear interpolation
         between them. It is backwards compatible with the LineString
         of GML 2.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractCurveType">
         <sequence>
            <element ref="gml:posList"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<!-- positions -->
<!-- =========================================================== -->
<element name="pos" type="gml:DirectPositionType"/>
<!-- =========================================================== -->
<complexType name="DirectPositionType">
   <annotation>
      <documentation>
         DirectPosition instances hold the coordinates for one position
         in the coordinate reference system (CRS) referenced in a
         larger element. In this case, the CRS shall be assumed to be
         the value referenced in the containing object's CRS.
      </documentation>
   </annotation>
   <simpleContent>
      <extension base="gml:doubleList"/>
   </simpleContent>
</complexType>
<!-- =========================================================== -->
<element name="posList" type="gml:DirectPositionListType"/>
<!-- =========================================================== -->
<complexType name="DirectPositionListType">
   <annotation>
      <documentation>
         DirectPositionList instances hold the coordinates for a
         sequence of direct positions within the same coordinate
         reference system (CRS).
      </documentation>
   </annotation>
   <simpleContent>
      <extension base="gml:doubleList"/>
   </simpleContent>
</complexType>
<!-- =========================================================== -->
<!-- Envelope -->
<!-- =========================================================== -->
<element name="Envelope" type="gml:EnvelopeType"/>
<!-- =========================================================== -->
<complexType name="EnvelopeType">
   <annotation>
      <documentation>
         Envelope defines an extent using a pair of positions
         defining opposite corners in arbitrary dimensions. The
         first direct position is the "lower corner" (a coordinate
         position consisting of all the minimal ordinates for each
         dimension for all points within the envelope), the second
         one the "upper corner" (a coordinate position consisting
         of all the maximal ordinates for each dimension for all
         points within the envelope).
      </documentation>
   </annotation>
   <sequence>
      <element name="lowerCorner" type="gml:DirectPositionType"/>
      <element name="upperCorner" type="gml:DirectPositionType"/>
   </sequence>
   <attribute name="srsName" type="anyURI" use="required">
     <annotation>
         <documentation>
         In general this reference points to a CRS instance of
         gml:CoordinateReferenceSystemType (see
         coordinateReferenceSystems.xsd). For well known
         references it is not required that the CRS description
         exists at the location the URI points to.
         </documentation>
     </annotation>
   </attribute>
</complexType>
<!-- ========================================================== -->
<!-- == Subset of feature.xsd for this profile ================ -->
<!-- ========================================================== -->
<!-- ========================================================== -->
<element name="_Feature" type="gml:AbstractFeatureType" abstract="true"
   substitutionGroup="gml:_GML"/>
<!-- ========================================================== -->
<complexType name="AbstractFeatureType" abstract="true">
   <annotation>
      <documentation>
         An abstract feature provides a set of common properties,
         including id, name and description inherited from
         AbstractGMLType, plus boundedBy.  A concrete feature type
         must derive from this type and specify additional properties
         in an application schema.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractFeatureBaseType">
         <sequence>
            <element ref="gml:boundedBy" minOccurs="0"/>
            <!-- additional properties must be specified in an
                 application schema -->
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- ========================================================== -->
<complexType name="AbstractFeatureBaseType">
   <annotation>
      <documentation>
         A abstract feature base type, that shall include an
         identifying attribute ('id').
      </documentation>
   </annotation>
   <complexContent>
      <restriction base="gml:AbstractGMLType">
         <sequence>
            <group ref="gml:StandardObjectProperties"/>
         </sequence>
         <attribute ref="gml:id" use="required"/>
      </restriction>
   </complexContent>
</complexType>
<!-- ========================================================== -->
<!-- ========================================================== -->
```

```
<element name="boundedBy" type="gml:BoundingShapeType"/>
<!-- =========================================================== -->
<complexType name="BoundingShapeType">
   <annotation>
      <documentation>Bounding shape.</documentation>
   </annotation>
   <sequence>
      <element ref="gml:Envelope"/>
   </sequence>
</complexType>
<!-- =========================================================== -->
<!-- == Subset of gmlBase.xsd for this profile ================= -->
<!-- =========================================================== -->
<!-- ==================== Objects ============================== -->
<!-- =========================================================== -->
<!-- =============== Abstract "Object" is "anyType" ============ -->
<!-- == Global element at the head of the "Object" substitution= -->
<!-- == group                                                  = -->
<!-- =========================================================== -->
<element name="_Object" abstract="true">
   <annotation>
      <documentation>
         This abstract element is the head of a substitutionGroup
         hierararchy which may contain either simpleContent or
         complexContent elements.  It is used to assert the model
         position of "class" elements declared in other GML schemas.
      </documentation>
   </annotation>
</element>
<!-- ============================================================ -->
<!-- == Abstract "GMLobject" supertype ========================== -->
<element name="_GML" type="gml:AbstractGMLType" abstract="true"
   substitutionGroup="gml:_Object">
   <annotation>
      <documentation>
         Global element which acts as the head of a substitution group
         that may include any element which is a GML feature, object,
         geometry or complex value
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<group name="StandardObjectProperties">
   <annotation>
      <documentation>
         This content model group makes it easier to construct types
         that derive from AbstractGMLType and its descendents "by
         restriction".  A reference to the group saves having to
         enumerate the standard object properties.
      </documentation>
   </annotation>
   <sequence>
      <element ref="gml:description" minOccurs="0"/>
      <element ref="gml:name" minOccurs="0" maxOccurs="unbounded">
         <annotation>
            <documentation>
               Multiple names may be provided.  These will often be
               distinguished by being assigned by different authorities,
               as indicated by the value of the codeSpace attribute.
               In an instance document there will usually only be one
               name per authority.
            </documentation>
         </annotation>
      </element>
   </sequence>
</group>
<!-- =========================================================== -->
<complexType name="AbstractGMLType" abstract="true">
   <annotation>
      <documentation>
         All complexContent GML elements are directly or indirectly
         derived from this abstract supertype to establish a hierarchy
         of GML types that may be distinguished from other XML types
         by their ancestry.  Elements in this hierarchy must have an
         ID and are thus referenceable.
      </documentation>
   </annotation>
   <sequence>
      <group ref="gml:StandardObjectProperties"/>
   </sequence>
   <attribute ref="gml:id" use="optional"/>
</complexType>
<!-- =========================================================== -->
<!-- == Base Property Types ==================================== -->
<!-- =========================================================== -->
<complexType name="ReferenceType">
   <annotation>
      <documentation>
         A pattern or base for derived types used to specify complex
         types corresponding to a UML aggregation association. An
         instance of this type serves as a pointer to a remote Object.
      </documentation>
   </annotation>
   <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!-- =========================================================== -->
<!-- ==========================================================
 global attribute, attribute group and element declarations
 ============================================================  -->
<attribute name="id" type="ID">
   <annotation>
      <documentation>
         Database handle for the object.  It is of XML type ID,
         so is constrained to be unique in the XML document within
         which it occurs.  An external identifier for the object
         in the form of a URI may be constructed using standard XML
         and XPointer methods.  This is done by concatenating the
         URI for the document, a fragment separator, and the value
         of the id attribute.
      </documentation>
   </annotation>
</attribute>
<!-- =========================================================== -->
<attributeGroup name="AssociationAttributeGroup">
   <annotation>
      <documentation>
         Attribute group used to enable property elements to refer
         to their value remotely. It contains the simple link
         components from xlinks.xsd, with all members optional.
         These attributes can be attached to any element, thus
         allowing it to act as a pointer.
      </documentation>
   </annotation>
   <attributeGroup ref="xlink:simpleLink"/>
</attributeGroup>
<!-- =========================================================== -->
<element name="name" type="gml:CodeType">
   <annotation>
      <documentation>
         Label for the object, normally a descriptive name. An
         object may have several names, typically assigned by
         different authorities.  The authority for a name is
         indicated by the value of its (optional) codeSpace
         attribute.  The name may or may not be unique, as
         determined by the rules of the organization responsible
         for the codeSpace.
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<element name="description" type="string">
   <annotation>
```

```
         <documentation>
            Contains a simple text description of the object.
         </documentation>
         <documentation>
            Restricted to only allow a text string, as done in GML 3.2.
         </documentation>
      </annotation>
   </element>
   <!-- =========================================================== -->
   <!-- == Subset of basicTypes.xsd for this profile ============== -->
   <!-- =========================================================== -->
   <!-- =========================================================== -->
   <simpleType name="doubleList">
      <annotation>
         <documentation>
            XML List based on XML Schema double type.  An element
            of this type contains a space-separated list of double
            values
         </documentation>
      </annotation>
      <list itemType="double"/>
   </simpleType>
   <!-- =========================================================== -->
   <complexType name="CodeType">
      <annotation>
         <documentation>
            Name or code with an (optional) authority.  Text token.
            If the codeSpace attribute is present, then its value
            should identify a dictionary, thesaurus or authority
            for the term, such as the organisation who assigned
            the value, or the dictionary from which it is taken.
            A text string with an optional codeSpace attribute.
         </documentation>
      </annotation>
      <simpleContent>
         <extension base="string">
            <attribute name="codeSpace" type="anyURI" use="optional"/>
         </extension>
      </simpleContent>
   </complexType>
   <!-- =========================================================== -->
   <complexType name="MeasureType">
      <annotation>
         <documentation>
            Number with a scale. The value of uom (Units Of Measure)
            attribute is a reference to a Reference System for the
            amount, either a ratio or position scale.
         </documentation>
      </annotation>
      <simpleContent>
         <extension base="double">
            <attribute name="uom" type="anyURI" use="required"/>
         </extension>
      </simpleContent>
   </complexType>
   <!-- =========================================================== -->
   <simpleType name="NCNameList">
      <annotation>
         <documentation>
            A set of values, representing a list of token with the
            lexical value space of NCName. The tokens are seperated
            by whitespace.
         </documentation>
      </annotation>
      <list itemType="NCName"/>
   </simpleType>
   <!-- =========================================================== -->
</schema>
```

D.2 GML schema for compliance level SF-2

The following schema document, called *gmlsf2.xsd*, contains only those element declarations from GML that are required to validate a GML application schema that complies with compliance level SF-2 as defined in this profile. GML application schemas can ensure compliance with this profile by importing this schema document rather 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"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified"
   version="1.0.0">

   <annotation>
      <appinfo
source="urn:opengis:specification:gml:schemaxsd:gmlsf:1.0.0">gmlsf.xsd</appinfo>
      <documentation>
         GML 3.1.1 Simplified Features profile Level 2.
         Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.
      </documentation>
   </annotation>

   <!-- =========================================================== -->
   <!-- === includes and imports                                === -->
   <!-- =========================================================== -->
   <import namespace="http://www.w3.org/1999/xlink"
   schemaLocation="http://schemas.opengis.net/xlink/1.0.0/xlinks.xsd"/>

   <!-- =========================================================== -->
   <!-- === Subset of geometryAggregates.xsd for this profile ===== -->
   <!-- =========================================================== -->
   <!-- aggregate geometry objects -->
   <!-- =========================================================== -->
   <element name="_GeometricAggregate"
            type="gml:AbstractGeometricAggregateType"
            substitutionGroup="gml:_Geometry"
            abstract="true">
      <annotation>
         <documentation>
            The "_GeometricAggregate" element is the abstract head of
            the substitution group for all geometric aggregates.
         </documentation>
      </annotation>
   </element>
   <!-- =========================================================== -->
   <complexType name="AbstractGeometricAggregateType" abstract="true">
      <annotation>
         <documentation>
            This is the abstract root type of the geometric aggregates.
         </documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractGeometryType"/>
      </complexContent>
   </complexType>
   <!-- =========================================================== -->
   <complexType name="MultiGeometryPropertyType">
      <annotation>
         <documentation>
            A property that has a geometric aggregate as its value domain
            can either be an appropriate geometry element encapsulated in
            an element of this type or an XLink reference to a remote
            geometry element (where remote includes geometry elements
            located elsewhere in the same document). Either the reference
            or the contained element must be given, but neither both nor none.
         </documentation>
      </annotation>
      <sequence minOccurs="0">
         <element ref="gml:_GeometricAggregate"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup">
         <annotation>
            <documentation>
               This attribute group includes the XLink attributes (see
               xlinks.xsd). XLink is used in GML to reference remote
               resources (including those elsewhere in the same document).
               A simple link element can be constructed by including a
               specific set of XLink attributes. The XML Linking Language
               (XLink) is currently a Proposed Recommendation of the World
               Wide Web Consortium. XLink allows elements to be inserted
               into XML documents so as to create sophisticated links
               between resources; such links can be used to reference
               remote properties.
               A simple link element can be used to implement pointer
               functionality, and this functionality has been built
               into various GML 3 elements by including the
               gml:AssociationAttributeGroup.
            </documentation>
         </annotation>
      </attributeGroup>
   </complexType>
   <!-- =========================================================== -->
   <element name="MultiPoint"
            type="gml:MultiPointType"
            substitutionGroup="gml:_GeometricAggregate"/>
   <!-- =========================================================== -->
   <complexType name="MultiPointType">
      <annotation>
         <documentation>
            A MultiPoint is defined by one or more Points, referenced
            through pointMember elements.
         </documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
               <element ref="gml:pointMember"
                        minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- =========================================================== -->
   <complexType name="MultiPointPropertyType">
      <annotation>
         <documentation>
            A property that has a collection of points as its value
            domain can either be an appropriate geometry element
            encapsulated in an element of this type or an XLink
            reference to a remote geometry element (where remote
            includes geometry elements located elsewhere in the
            same document). Either the reference or the contained
            element must be given, but neither both nor none.
         </documentation>
      </annotation>
      <sequence minOccurs="0">
         <element ref="gml:MultiPoint"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup">
         <annotation>
            <documentation>
               This attribute group includes the XLink attributes
               (see xlinks.xsd). XLink is used in GML to reference
               remote resources (including those elsewhere in the
               same document). A simple link element can be constructed
               by including a specific set of XLink attributes. The XML
               Linking Language (XLink) is currently a Proposed Recommendation
               of the World Wide Web Consortium. XLink allows elements to be
               inserted into XML documents so as to create sophisticated
               links between resources; such links can be used to reference
               remote properties.
               A simple link element can be used to implement pointer
               functionality, and this functionality has been built into
               various GML 3 elements by including the
               gml:AssociationAttributeGroup.
            </documentation>
         </annotation>
      </attributeGroup>
   </complexType>
   <!-- =========================================================== -->
   <element name="MultiCurve"
            type="gml:MultiCurveType"
            substitutionGroup="gml:_GeometricAggregate"/>
   <!-- =========================================================== -->
   <complexType name="MultiCurveType">
      <annotation>
         <documentation>
            A MultiCurve is defined by one or more Curves, referenced
            through curveMember elements.
         </documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
               <element ref="gml:curveMember"
                        minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- =========================================================== -->
   <complexType name="MultiCurvePropertyType">
      <annotation>
         <documentation>
            A property that has a collection of curves as its value domain
            can either be an appropriate geometry element encapsulated in
            an element of this type or an XLink reference to a remote
            geometry element (where remote includes geometry elements
            located elsewhere in the same document). Either the reference
            or the contained element must be given, but neither both nor
            none.
         </documentation>
      </annotation>
      <sequence minOccurs="0">
         <element ref="gml:MultiCurve"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup">
         <annotation>
            <documentation>
               This attribute group includes the XLink attributes (see
               xlinks.xsd). XLink is used in GML to reference remote
               resources (including those elsewhere in the same document).
               A simple link element can be constructed by including a
               specific set of XLink attributes. The XML Linking Language
               (XLink) is currently a Proposed Recommendation of the World
               Wide Web Consortium. XLink allows elements to be inserted
               into XML documents so as to create sophisticated links between
               resources; such links can be used to reference remote properties.
               A simple link element can be used to implement pointer
               functionality, and this functionality has been built into
               various GML 3 elements by including the
               gml:AssociationAttributeGroup.
            </documentation>
         </annotation>
      </attributeGroup>
```

```
</complexType>
<!-- =========================================================== -->
<element name="MultiSurface"
         type="gml:MultiSurfaceType"
         substitutionGroup="gml:_GeometricAggregate"/>
<!-- =========================================================== -->
<complexType name="MultiSurfaceType">
   <annotation>
      <documentation>
         A MultiSurface is defined by one or more Surfaces,
         referenced through surfaceMember elements.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGeometricAggregateType">
         <sequence>
            <element ref="gml:surfaceMember"
                     minOccurs="0" maxOccurs="unbounded"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<complexType name="MultiSurfacePropertyType">
   <annotation>
      <documentation>
         A property that has a collection of surfaces as its value
         domain can either be an appropriate geometry element
         encapsulated in an element of this type or an XLink
         reference to a remote geometry element (where remote
         includes geometry elements located elsewhere in the
         same document). Either the reference or the contained
         element must be given, but neither both nor none.
      </documentation>
   </annotation>
   <sequence minOccurs="0">
      <element ref="gml:MultiSurface"/>
   </sequence>
   <attributeGroup ref="gml:AssociationAttributeGroup">
      <annotation>
         <documentation>
            This attribute group includes the XLink attributes (see
            xlinks.xsd). XLink is used in GML to reference remote
            resources (including those elsewhere in the same document).
            A simple link element can be constructed by including a
            specific set of XLink attributes. The XML Linking Language
            (XLink) is currently a Proposed Recommendation of the World
            Wide Web Consortium. XLink allows elements to be inserted
            into XML documents so as to create sophisticated links
            between resources; such links can be used to reference
            remote properties.
        A simple link element can be used to implement pointer
            functionality, and this functionality has been built
            into various GML 3 elements by including the
            gml:AssociationAttributeGroup.
         </documentation>
      </annotation>
   </attributeGroup>
</complexType>
<!-- =========================================================== -->
<element name="pointMember" type="gml:PointPropertyType">
   <annotation>
      <documentation>
         This property element either references a Point via the
         XLink-attributes or contains the Point element.
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<element name="surfaceMember" type="gml:SurfacePropertyType">
   <annotation>
```

```
      <documentation>
         This property element either references a surface via the
         XLink-attributes or contains the surface element. A surface
         element is any element which is substitutable for "_Surface".
      </documentation>
   </annotation>
</element>
<!-- ============================================================= -->
<!-- == Subset of geometryPrimitives.xsd for this profile ====== -->
<!-- ============================================================= -->
<element name="Curve" type="gml:CurveType" substitutionGroup="gml:_Curve"/>
<!-- ============================================================= -->
<complexType name="CurveType">
   <annotation>
      <documentation>
         Curve is a 1-dimensional primitive. Curves are continuous,
         connected, and have a measurable length in terms of the
         coordinate system.
         A curve is composed of one or more curve segments.
         The curve segments are connected to one another, with the
         end point of each segment except the last being the start
         point of the next segment in the segment list.
         The orientation of the curve is positive.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractCurveType">
         <sequence>
            <element ref="gml:segments">
               <annotation>
                  <documentation>
                     This element encapsulates the segments of the
                     curve.
                  </documentation>
               </annotation>
            </element>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- ============================================================= -->
<!-- curve segments (1-dimensional) -->
<!-- ============================================================= -->
<element name="_CurveSegment"
         type="gml:AbstractCurveSegmentType"
         abstract="true">
   <annotation>
      <documentation>
         The "_CurveSegment" element is the abstract head of the
         substituition group for all curve segment elements,
         i.e. continuous segments of the same interpolation
         mechanism.
      </documentation>
   </annotation>
</element>
<!-- ============================================================= -->
<complexType name="AbstractCurveSegmentType" abstract="true">
   <annotation>
      <documentation>
         Curve segment defines a homogeneous segment of a curve.
      </documentation>
   </annotation>
   <sequence/>
</complexType>
<!-- ============================================================= -->
<element name="segments" type="gml:CurveSegmentArrayPropertyType">
   <annotation>
      <documentation>
         This property element contains a list of curve segments.
         The order of the elements is significant and shall be
         preserved when processing the array.
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<complexType name="CurveSegmentArrayPropertyType">
   <annotation>
      <documentation>
         A container for an array of curve segments.
      </documentation>
   </annotation>
   <sequence>
      <element ref="gml:_CurveSegment" minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
</complexType>
<!-- =========================================================== -->
<element name="curveMember" type="gml:CurvePropertyType">
   <annotation>
      <documentation>
         This property element either references a curve via the
         XLink-attributes or contains the curve element. A curve
         element is any element which is substitutable for "_Curve".
      </documentation>
   </annotation>
</element>
<!-- ============================================================ -->
<!-- === global element in "_CurveSegment" substitution group === -->
<!-- ============================================================ -->
<element name="LineStringSegment"
         type="gml:LineStringSegmentType"
         substitutionGroup="gml:_CurveSegment"/>
<!-- ============================================================ -->
<complexType name="LineStringSegmentType">
   <annotation>
      <documentation>
         A LineStringSegment is a curve segment that is defined by
         two or more coordinate tuples, with linear interpolation
         between them.
         Note: LineStringSegment implements GM_LineString of ISO 19107.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractCurveSegmentType">
         <sequence>
            <element ref="gml:posList"/>
         </sequence>
         <attribute name="interpolation"
                    type="gml:CurveInterpolationType"
                    fixed="linear">
            <annotation>
               <documentation>
                  The attribute "interpolation" specifies
                  the curve interpolation mechanism used
                  for this segment. This mechanism uses
                  the control points and control parameters
                  to determine the position of this curve
                  segment. For a LineStringSegment the
                  interpolation is fixed as "linear".
               </documentation>
            </annotation>
         </attribute>
      </extension>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<!-- surface patches (2-dimensional) -->
<!-- =========================================================== -->
<element name="_SurfacePatch"
         type="gml:AbstractSurfacePatchType"
         abstract="true">
   <annotation>
      <documentation>
```

```
         The "_SurfacePatch" element is the abstract head of the
         substituition group for all surface pach elements describing
         a continuous portion of a surface.
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<complexType name="AbstractSurfacePatchType" abstract="true">
   <annotation>
      <documentation>
         A surface patch defines a homogenuous portion of a surface.
      </documentation>
   </annotation>
   <sequence/>
</complexType>
<!-- =========================================================== -->
<element name="patches" type="gml:SurfacePatchArrayPropertyType">
   <annotation>
      <documentation>
          This property element contains a list of surface patches.
          The order of the elements is significant and shall be
          preserved when processing the array.
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<complexType name="SurfacePatchArrayPropertyType">
   <annotation>
      <documentation>
         A container for an array of surface patches.
      </documentation>
   </annotation>
   <sequence>
      <element ref="gml:_SurfacePatch"
               minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
</complexType>
<!-- =========================================================== -->
<element name="PolygonPatch"
         type="gml:PolygonPatchType"
         substitutionGroup="gml:_SurfacePatch"/>
<!-- =========================================================== -->
<complexType name="PolygonPatchType">
   <annotation>
      <documentation>
         A PolygonPatch is a surface patch that is defined by
         a set of boundary curves and an underlying surface to
         which these curves adhere. The curves are coplanar and
         the polygon uses planar interpolation in its interior.
         Implements GM_Polygon of ISO 19107.
      </documentation>
   </annotation>
   <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">
            <annotation>
               <documentation>
                  The attribute "interpolation" specifies the
                  interpolation mechanism used for this surface
                  patch. Currently only planar surface patches
                  are defined in GML 3, the attribute is fixed
                  to "planar", i.e. the interpolation method
                  shall return points on a single plane. The
                  boundary of the patch shall be contained within
                  that plane.
               </documentation>
            </annotation>
         </attribute>
      </extension>
   </complexContent>
</complexType>
<!-- ============================================================ -->
<element name="Surface"
              type="gml:SurfaceType"
              substitutionGroup="gml:_Surface"/>
<!-- ============================================================ -->
<complexType name="SurfaceType">
   <annotation>
      <documentation>
         A Surface is a 2-dimensional primitive and is composed
         of one or more surface patches. The surface patches are
         connected to one another.
         The orientation of the surface is positive ("up"). The
         orientation of a surface chooses an "up" direction
         through the choice of the upward normal, which, if the
         surface is not a cycle, is the side of the surface from
         which the exterior boundary appears counterclockwise.
         Reversal of the surface orientation reverses the curve
         orientation of each boundary component, and interchanges
         the conceptual "up" and "down" direction of the surface.
         If the surface is the boundary of a solid, the "up"
         direction is usually outward. For closed surfaces, which
         have no boundary, the up direction is that of the surface
         patches, which must be consistent with one another. Its
         included surface patches describe the interior structure
         of the Surface.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractSurfaceType">
         <sequence>
            <element ref="gml:patches">
               <annotation>
                  <documentation>
                     This element encapsulates the patches of the
                     surface.
                  </documentation>
               </annotation>
            </element>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- ============================================================ -->
<simpleType name="CurveInterpolationType">
   <annotation>
      <documentation>
         CurveInterpolationType is a list of codes that may be
         used to identify the interpolation mechanisms specified
         by an schema.
      </documentation>
   </annotation>
   <restriction base="string">
      <enumeration value="linear"/>
   </restriction>
</simpleType>
<simpleType name="SurfaceInterpolationType">
   <annotation>
      <documentation>
         SurfaceInterpolationType is a list of codes that may be
         used to identify the interpolation mechanisms specified
         by an application schema.
      </documentation>
   </annotation>
   <restriction base="string">
```

```
         <enumeration value="planar"/>
      </restriction>
   </simpleType>
   <!-- ============================================================ -->
   <!-- == Subset of geometryBasic2d.xsd for this profile ========= -->
   <!-- ============================================================ -->
   <!-- == primitive geometry objects (2-dimensional) ============= -->
   <!-- ============================================================ -->
   <element name="_Surface"
            type="gml:AbstractSurfaceType"
            substitutionGroup="gml:_GeometricPrimitive"
            abstract="true">
      <annotation>
         <documentation>
            The "_Surface" element is the abstract head of the
            substituition group for all (continuous) surface elements.
         </documentation>
      </annotation>
   </element>
   <!-- ============================================================ -->
   <complexType name="AbstractSurfaceType">
      <annotation>
         <documentation>
            An abstraction of a surface to support the different
            levels of complexity. A surface is always a continuous
            region of a plane.
         </documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractGeometricPrimitiveType"/>
      </complexContent>
   </complexType>
   <!-- ============================================================ -->
   <complexType name="SurfacePropertyType">
      <annotation>
         <documentation>
            A property that has a surface as its value domain can
            either be an appropriate geometry element encapsulated
            in an element of this type or an XLink reference to a
            remote geometry element (where remote includes geometry
            elements located elsewhere in the same document). Either
            the reference or the contained element must be given,
            but neither both nor none.
         </documentation>
      </annotation>
      <sequence minOccurs="0">
         <element ref="gml:_Surface"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup">
         <annotation>
            <documentation>
               This attribute group includes the XLink attributes (see
               xlinks.xsd). XLink is used in GML to reference remote
               resources (including those elsewhere in the same document).
               A simple link element can be constructed by including a
               specific set of XLink attributes. The XML Linking Language
               (XLink) is currently a Proposed Recommendation of the World
               Wide Web Consortium. XLink allows elements to be inserted
               into XML documents so as to create sophisticated links
               between resources; such links can be used to reference
               remote properties. A simple link element can be used to
               implement pointer functionality, and this functionality
               has been built into various GML 3 elements by including
               the gml:AssociationAttributeGroup.
            </documentation>
         </annotation>
      </attributeGroup>
   </complexType>
   <!-- ============================================================ -->
   <element name="Polygon"
            type="gml:PolygonType"
            substitutionGroup="gml:_Surface"/>
   <!-- =========================================================== -->
   <complexType name="PolygonType">
      <annotation>
         <documentation>
            A Polygon is a special surface that is defined by a single
            surface patch. The boundary of this patch is coplanar and
            the polygon uses planar interpolation in its interior. It
            is backwards compatible with the Polygon of GML 2.
         </documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractSurfaceType">
            <sequence>
               <element ref="gml:exterior" minOccurs="0"/>
               <element ref="gml:interior" minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
         </extension>
      </complexContent>
   </complexType>
   <!-- =========================================================== -->
   <!-- rings (closed curves for surface boundaries) -->
   <!-- =========================================================== -->
   <element name="exterior" type="gml:AbstractRingPropertyType">
      <annotation>
         <documentation>
            A boundary of a surface consists of a number of rings.
            In the normal 2D case, one of these rings is distinguished
            as being the exterior boundary. In a general manifold this
            is not always possible, in which case all boundaries shall
            be listed as interior boundaries, and the exterior will be
            empty.
         </documentation>
      </annotation>
   </element>
   <!-- =========================================================== -->
   <element name="interior" type="gml:AbstractRingPropertyType">
      <annotation>
         <documentation>
            A boundary of a surface consists of a number of rings. The
            "interior" rings seperate the surface / surface patch from
            the area enclosed by the rings.
         </documentation>
      </annotation>
   </element>
   <!-- =========================================================== -->
   <complexType name="AbstractRingPropertyType">
      <annotation>
         <documentation>
            Encapsulates a ring to represent the surface boundary property
            of a surface.
         </documentation>
      </annotation>
      <sequence>
         <element ref="gml:LinearRing"/>
      </sequence>
   </complexType>
   <!-- =========================================================== -->
   <element name="LinearRing" type="gml:LinearRingType"
      substitutionGroup="gml:_Geometry"/>
   <!-- =========================================================== -->
   <complexType name="LinearRingType">
      <annotation>
         <documentation>
            A LinearRing is defined by four or more coordinate tuples,
            with linear interpolation between them; the first and last
            coordinates must be coincident.
         </documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractGeometryType">
```

```
         <sequence>
            <element ref="gml:posList">
               <annotation>
                  <documentation>
                     The "posList" element provides a compact way to
                     specifiy the coordinates of the control points,
                     if all control points are in the same coordinate
                     reference systems and belong to this ring only.
                     The number of direct positions in the list must
                     be at least four.
                  </documentation>
               </annotation>
            </element>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- ============================================================ -->
<!-- == Subset of geometryBasic0d1d.xsd for this profile ======= -->
<!-- ============================================================ -->
<!-- ============================================================ -->
<!-- == abstract supertype for geometry objects ================ -->
<!-- ============================================================ -->
<element name="_Geometry" type="gml:AbstractGeometryType"
   substitutionGroup="gml:_GML" abstract="true">
   <annotation>
      <documentation>
         The "_Geometry" element is the abstract head of the
         substituition group for all geometry elements of GML 3.
         This includes pre-defined and user-defined geometry elements.
         Any geometry element must be a direct or indirect
         extension/restriction  of AbstractGeometryType
         and must be directly or indirectly in the substitution
         group of "_Geometry".
      </documentation>
   </annotation>
</element>
<!-- ============================================================ -->
<complexType name="GeometryPropertyType">
   <annotation>
      <documentation>
         A geometric property can either be any geometry element
         encapsulated in an element of this type or an XLink
         reference to a remote geometry element (where remote
         includes geometry elements located elsewhere in the same
         document). Note that either the reference or the contained
         element must be given, but not both or none.
      </documentation>
   </annotation>
   <sequence minOccurs="0">
      <element ref="gml:_Geometry"/>
   </sequence>
   <attributeGroup ref="gml:AssociationAttributeGroup">
      <annotation>
         <documentation>
            This attribute group includes the XLink attributes (see
            xlinks.xsd). XLink is used in GML to reference remote
            resources (including those elsewhere in the same document).
            A simple link element can be constructed by including a
            specific set of XLink attributes. The XML Linking Language
            (XLink) is currently a Proposed Recommendation of the World
            Wide Web Consortium. XLink allows elements to be inserted
            into XML documents so as to create sophisticated links
            between resources; such links can be used to reference
            remote properties. A simple link element can be used to
            implement pointer functionality, and this functionality
            has been built into various GML 3 elements by including
            the gml:AssociationAttributeGroup.
         </documentation>
      </annotation>
   </attributeGroup>
```

```
</complexType>
<!-- ============================================================== -->
<complexType name="AbstractGeometryType" abstract="true">
   <annotation>
      <documentation>
         All geometry elements are derived directly or indirectly
         from this abstract supertype. A geometry element may
         have an identifying attribute ("gml:id"), a name (attribute
         "name") and a description (attribute "description"). It may
         be associated with a spatial reference system (attribute
         "srsName"). The following rules shall be adhered: - Every
         geometry type shall derive from this abstract type. - Every
         geometry element (i.e. an element of a geometry type) shall
         be directly or indirectly in the substitution group of _Geometry.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGMLType">
          <attribute name="srsName" type="anyURI" use="optional">
              <annotation>
                  <documentation>
         In general this reference points to a CRS instance of
         gml:CoordinateReferenceSystemType (see
         coordinateReferenceSystems.xsd). For well known
         references it is not required that the CRS description
         exists at the location the URI points to. If no srsName
         attribute is given, the CRS must be specified as part of
         the larger context this geometry element is part of, e.g.
         a geometric element like point, curve, etc. It is expected
         that this attribute will be specified at the direct position
         level only in rare cases.
                  </documentation>
              </annotation>
          </attribute>
      </extension>
   </complexContent>
</complexType>
<!-- ============================================================== -->
<element name="_GeometricPrimitive" type="gml:AbstractGeometricPrimitiveType"
   abstract="true" substitutionGroup="gml:_Geometry">
   <annotation>
      <documentation>
         The "_GeometricPrimitive" element is the abstract head of
         the substituition group for all (pre- and user-defined)
         geometric primitives.
      </documentation>
   </annotation>
</element>
<!-- ============================================================== -->
<complexType name="AbstractGeometricPrimitiveType" abstract="true">
   <annotation>
      <documentation>
         This is the abstract root type of the geometric primitives.
         A geometric primitive is a geometric object that is not
         decomposed further into other primitives in the system.
         All primitives are oriented in the direction implied by
         the sequence of their coordinate tuples.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGeometryType"/>
   </complexContent>
</complexType>
<!-- ============================================================== -->
<!-- primitive geometry objects (0-dimensional) -->
<!-- ============================================================== -->
<element name="Point"
         type="gml:PointType"
         substitutionGroup="gml:_GeometricPrimitive"/>
<!-- ============================================================== -->
<complexType name="PointType">
```

```
   <annotation>
      <documentation>
         A Point is defined by a single coordinate tuple.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGeometricPrimitiveType">
         <sequence>
            <element ref="gml:pos"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- ========================================================== -->
<complexType name="PointPropertyType">
   <annotation>
      <documentation>
         A property that has a point as its value domain can either
         be an appropriate geometry element encapsulated in an element
         of this type or an XLink reference to a remote geometry
         element (where remote includes geometry elements located
         elsewhere in the same document). Either the reference or the
         contained element must be given, but neither both nor none.
      </documentation>
   </annotation>
   <sequence minOccurs="0">
      <element ref="gml:Point"/>
   </sequence>
   <attributeGroup ref="gml:AssociationAttributeGroup">
      <annotation>
         <documentation>
            This attribute group includes the XLink attributes (see
            xlinks.xsd). XLink is used in GML to reference remote
            resources (including those elsewhere in the same document).
            A simple link element can be constructed by including a
            specific  set of XLink attributes. The XML Linking Language
            (XLink) is currently a Proposed Recommendation of the World
            Wide Web Consortium. XLink allows elements to be inserted
            into XML documents so as to create sophisticated links
            between resources; such links can be  used to reference
            remote properties. A simple link element can be used to
            implement pointer functionality, and this functionality
            has been built into various GML 3 elements by including
            the gml:AssociationAttributeGroup.
         </documentation>
      </annotation>
   </attributeGroup>
</complexType>
 <!-- ========================================================== -->
<!-- primitive geometry objects (1-dimensional) -->
<!-- ========================================================== -->
<element name="_Curve" type="gml:AbstractCurveType"
   substitutionGroup="gml:_GeometricPrimitive" abstract="true">
   <annotation>
      <documentation>
         The "_Curve" element is the abstract head of the
         substituition group for all (continuous) curve elements.
      </documentation>
   </annotation>
</element>
<!-- ========================================================== -->
<complexType name="AbstractCurveType" abstract="true">
   <annotation>
      <documentation>
         An abstraction of a curve to support the different levels
         of complexity. The curve can always be viewed as a geometric
         primitive, i.e. is continuous.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractGeometricPrimitiveType"/>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<complexType name="CurvePropertyType">
   <annotation>
      <documentation>
         A property that has a curve as its value domain can either
         be an appropriate geometry element encapsulated in an
         element of this type or an XLink reference to a remote
         geometry element (where remote includes geometry elements
         located elsewhere in the same document). Either the reference
         or the contained element must be given, but neither both nor none.
      </documentation>
   </annotation>
   <sequence minOccurs="0">
      <element ref="gml:_Curve"/>
   </sequence>
   <attributeGroup ref="gml:AssociationAttributeGroup">
      <annotation>
         <documentation>
            This attribute group includes the XLink attributes (see
            xlinks.xsd). XLink is used in GML to reference remote
            resources (including those elsewhere in the same document).
            A simple link element can be constructed by including a
            specific set of XLink attributes.

            The XML Linking Language (XLink) is currently a Proposed
            Recommendation of the World Wide Web Consortium. XLink
            allows elements to be inserted into XML documents so as
            to create sophisticated links between resources; such
            links can be used to reference remote properties. A simple
            link element can be used to implement pointer functionality,
            and this functionality has been built into various GML 3
            elements by including the gml:AssociationAttributeGroup.
         </documentation>
      </annotation>
   </attributeGroup>
</complexType>
<!-- =========================================================== -->
<element name="LineString"
         type="gml:LineStringType"
         substitutionGroup="gml:_Curve"/>
<!-- =========================================================== -->
<complexType name="LineStringType">
   <annotation>
      <documentation>
         A LineString is a special curve that consists of a
         single segment with linear interpolation. It is defined
         by two or more coordinate tuples, with linear interpolation
         between them. It is backwards compatible with the LineString
         of GML 2.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractCurveType">
         <sequence>
            <element ref="gml:posList"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<!-- positions -->
<!-- =========================================================== -->
<element name="pos" type="gml:DirectPositionType"/>
<!-- =========================================================== -->
<complexType name="DirectPositionType">
   <annotation>
      <documentation>
         DirectPosition instances hold the coordinates for one position
         in the coordinate reference system (CRS) referenced in a
         larger element. In this case, the CRS shall be assumed to be
         the value referenced in the containing object's CRS.
      </documentation>
   </annotation>
   <simpleContent>
      <extension base="gml:doubleList"/>
   </simpleContent>
</complexType>
<!-- =========================================================== -->
<element name="posList" type="gml:DirectPositionListType"/>
<!-- =========================================================== -->
<complexType name="DirectPositionListType">
   <annotation>
      <documentation>
         DirectPositionList instances hold the coordinates for a
         sequence of direct positions within the same coordinate
         reference system (CRS).
      </documentation>
   </annotation>
   <simpleContent>
      <extension base="gml:doubleList"/>
   </simpleContent>
</complexType>
<!-- =========================================================== -->
<!-- Envelope -->
<!-- =========================================================== -->
<element name="Envelope" type="gml:EnvelopeType"/>
<!-- =========================================================== -->
<complexType name="EnvelopeType">
   <annotation>
      <documentation>
         Envelope defines an extent using a pair of positions
         defining opposite corners in arbitrary dimensions. The
         first direct position is the "lower corner" (a coordinate
         position consisting of all the minimal ordinates for each
         dimension for all points within the envelope), the second
         one the "upper corner" (a coordinate position consisting
         of all the maximal ordinates for each dimension for all
         points within the envelope).
      </documentation>
   </annotation>
   <sequence>
      <element name="lowerCorner" type="gml:DirectPositionType"/>
      <element name="upperCorner" type="gml:DirectPositionType"/>
   </sequence>
   <attribute name="srsName" type="anyURI" use="required">
     <annotation>
          <documentation>
         In general this reference points to a CRS instance of
         gml:CoordinateReferenceSystemType (see
         coordinateReferenceSystems.xsd). For well known
         references it is not required that the CRS description
         exists at the location the URI points to.
          </documentation>
     </annotation>
   </attribute>
</complexType>
<!-- =========================================================== -->
<!-- == Subset of feature.xsd for this profile ================= -->
<!-- =========================================================== -->
<!-- =========================================================== -->
<element name="_Feature" type="gml:AbstractFeatureType" abstract="true"
   substitutionGroup="gml:_GML"/>
<!-- =========================================================== -->
<complexType name="AbstractFeatureType" abstract="true">
   <annotation>
      <documentation>
         An abstract feature provides a set of common properties,
         including id, name and description inherited from
         AbstractGMLType, plus boundedBy.  A concrete feature type
         must derive from this type and specify additional properties
         in an application schema.
      </documentation>
   </annotation>
   <complexContent>
      <extension base="gml:AbstractFeatureBaseType">
         <sequence>
            <element ref="gml:boundedBy" minOccurs="0"/>
            <!-- additional properties must be specified in an
                 application schema -->
         </sequence>
      </extension>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<complexType name="AbstractFeatureBaseType">
   <annotation>
      <documentation>
         A abstract feature base type, that shall include an
         identifying attribute ('id').
      </documentation>
   </annotation>
   <complexContent>
      <restriction base="gml:AbstractGMLType">
         <sequence>
            <group ref="gml:StandardObjectProperties"/>
         </sequence>
         <attribute ref="gml:id" use="required"/>
      </restriction>
   </complexContent>
</complexType>
<!-- =========================================================== -->
<!-- =========================================================== -->
<element name="boundedBy" type="gml:BoundingShapeType"/>
<!-- =========================================================== -->
<complexType name="BoundingShapeType">
   <annotation>
      <documentation>Bounding shape.</documentation>
   </annotation>
   <sequence>
      <element ref="gml:Envelope"/>
   </sequence>
</complexType>
<!-- =========================================================== -->
<!-- == Subset of gmlBase.xsd for this profile ================= -->
<!-- =========================================================== -->
<!-- ==================== Objects ============================== -->
<!-- =========================================================== -->
<!-- =============== Abstract "Object" is "anyType" ============ -->
<!-- == Global element at the head of the "Object" substitution= -->
<!-- == group                                                  = -->
<!-- =========================================================== -->
<element name="_Object" abstract="true">
   <annotation>
      <documentation>
         This abstract element is the head of a substitutionGroup
         hierararchy which may contain either simpleContent or
         complexContent elements.  It is used to assert the model
         position of "class" elements declared in other GML schemas.
      </documentation>
   </annotation>
</element>
<!-- ============================================================ -->
<!-- == Abstract "GMLobject" supertype ========================== -->
<element name="_GML" type="gml:AbstractGMLType" abstract="true"
   substitutionGroup="gml:_Object">
   <annotation>
      <documentation>
         Global element which acts as the head of a substitution group
         that may include any element which is a GML feature, object,
         geometry or complex value
      </documentation>
   </annotation>
</element>
<!-- ============================================================ -->
<group name="StandardObjectProperties">
   <annotation>
      <documentation>
         This content model group makes it easier to construct types
         that derive from AbstractGMLType and its descendents "by
         restriction".  A reference to the group saves having to
         enumerate the standard object properties.
      </documentation>
   </annotation>
   <sequence>
      <element ref="gml:description" minOccurs="0"/>
      <element ref="gml:name" minOccurs="0" maxOccurs="unbounded">
         <annotation>
            <documentation>
               Multiple names may be provided.  These will often be
               distinguished by being assigned by different authorities,
               as indicated by the value of the codeSpace attribute.
               In an instance document there will usually only be one
               name per authority.
            </documentation>
         </annotation>
      </element>
   </sequence>
</group>
<!-- ============================================================ -->
<complexType name="AbstractGMLType" abstract="true">
   <annotation>
      <documentation>
         All complexContent GML elements are directly or indirectly
         derived from this abstract supertype to establish a hierarchy
         of GML types that may be distinguished from other XML types
         by their ancestry.  Elements in this hierarchy must have an
         ID and are thus referenceable.
      </documentation>
   </annotation>
   <sequence>
      <group ref="gml:StandardObjectProperties"/>
   </sequence>
   <attribute ref="gml:id" use="optional"/>
</complexType>
<!-- ============================================================ -->
<!-- == Base Property Types ===================================== -->
<!-- ============================================================ -->
<complexType name="ReferenceType">
   <annotation>
      <documentation>
         A pattern or base for derived types used to specify complex
         types corresponding to a UML aggregation association. An
         instance of this type serves as a pointer to a remote Object.
      </documentation>
   </annotation>
   <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!-- ============================================================ -->
<!-- ===========================================================
 global attribute, attribute group and element declarations
 ============================================================  -->
<attribute name="id" type="ID">
   <annotation>
      <documentation>
         Database handle for the object.  It is of XML type ID,
         so is constrained to be unique in the XML document within
         which it occurs.  An external identifier for the object
         in the form of a URI may be constructed using standard XML
         and XPointer methods.  This is done by concatenating the
         URI for the document, a fragment separator, and the value
         of the id attribute.
      </documentation>
   </annotation>
</attribute>
<!-- =========================================================== -->
<attributeGroup name="AssociationAttributeGroup">
   <annotation>
      <documentation>
         Attribute group used to enable property elements to refer
         to their value remotely. It contains the simple link
         components from xlinks.xsd, with all members optional.
         These attributes can be attached to any element, thus
         allowing it to act as a pointer.
      </documentation>
   </annotation>
   <attributeGroup ref="xlink:simpleLink"/>
</attributeGroup>
<!-- =========================================================== -->
<element name="name" type="gml:CodeType">
   <annotation>
      <documentation>
         Label for the object, normally a descriptive name. An
         object may have several names, typically assigned by
         different authorities.  The authority for a name is
         indicated by the value of its (optional) codeSpace
         attribute.  The name may or may not be unique, as
         determined by the rules of the organization responsible
         for the codeSpace.
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<element name="description" type="string">
   <annotation>
      <documentation>
         Contains a simple text description of the object.
      </documentation>
      <documentation>
         Restricted to only allow a text string, as done in GML 3.2.
      </documentation>
   </annotation>
</element>
<!-- =========================================================== -->
<!-- == Subset of basicTypes.xsd for this profile ============== -->
<!-- =========================================================== -->
<!-- =========================================================== -->
<simpleType name="doubleList">
   <annotation>
      <documentation>
         XML List based on XML Schema double type.  An element
         of this type contains a space-separated list of double
         values
      </documentation>
   </annotation>
   <list itemType="double"/>
</simpleType>
<!-- =========================================================== -->
<complexType name="CodeType">
   <annotation>
      <documentation>
         Name or code with an (optional) authority.  Text token.
         If the codeSpace attribute is present, then its value
         should identify a dictionary, thesaurus or authority
         for the term, such as the organisation who assigned
         the value, or the dictionary from which it is taken.
         A text string with an optional codeSpace attribute.
      </documentation>
   </annotation>
   <simpleContent>
      <extension base="string">
         <attribute name="codeSpace" type="anyURI" use="optional"/>
      </extension>
   </simpleContent>
   </complexType>
   <!-- =========================================================== -->
   <complexType name="MeasureType">
      <annotation>
         <documentation>
            Number with a scale. The value of uom (Units Of Measure)
            attribute is a reference to a Reference System for the
            amount, either a ratio or position scale.
         </documentation>
      </annotation>
      <simpleContent>
         <extension base="double">
            <attribute name="uom" type="anyURI" use="required"/>
         </extension>
      </simpleContent>
   </complexType>
   <!-- =========================================================== -->
   <simpleType name="NCNameList">
      <annotation>
         <documentation>
            A set of values, representing a list of token with the
            lexical value space of NCName. The tokens are seperated
            by whitespace.
         </documentation>
      </annotation>
      <list itemType="NCName"/>
   </simpleType>
   <!-- =========================================================== -->
</schema>
```

Annex E
(normative)

Schema for compliance level element (Level SF-0, SF-1, SF-2)

The following schema document, called *gmlsfLevels.xsd*, declares the elements **gmlsf:ComplianceLevel** and **gmlsf:GMLProfileSchema**. The first element is used in an appinfo annotation, in a conformant GML application schema, to declare the compliance level to which the schema complies. Using this element, a consumer of the schema may determine its compliance level without having to interpret the entire schema.

The **gmlsf:GMLProfileSchema** element allows to specify the GML Profile schema describing associated with the application schema. See Annex D.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
   targetNamespace="http://www.opengis.net/gmlsf"
   xmlns:gmlsf="http://www.opengis.net/gmlsf"
   xmlns="http://www.w3.org/2001/XMLSchema"
   elementFormDefault="qualified">
   <annotation>
      <appinfo
source="urn:opengis:specification:gml:schema-xsd:gmlsf:1.0.0">gmlsfLevels.xsd</appinfo>
      <documentation>
         Compliance levels schema for Simple features GML 3.1.1 Profile
         Copyright (c)2006 Open Geospatial Consortium, Inc. All Rights Reserved.
      </documentation>
   </annotation>
   <!-- ============================================================= -->
   <element name="ComplianceLevel">
      <annotation>
         <documentation>
            Level 0 = no complex-valued properties and minOccurs,maxOccurs
                      have a value domain of 0 or 1
            Level 1 = complex-valued properties with no restriction on
                      minOccurs and maxOccurs
            Level 2 = no restrictions on type of non-spatial scalar properties
                      but must only support spatial properties declared in
                      clause 8
         </documentation>
      </annotation>
      <simpleType>
         <restriction base="integer">
            <enumeration value="0"/>
            <enumeration value="1"/>
            <enumeration value="2"/>
         </restriction>
      </simpleType>
   </element>
   <!-- ============================================================= -->
   <element name="GMLProfileSchema" type="anyURI">
      <annotation>
         <documentation>
            This URI references the profile schema to which a GML
            application schema conforms.
         </documentation>
      </annotation>
   </element>
```

```
</schema>
```

Annex F
(informative)

Revision history

Date	Release	Description
22-May-08	1.0	Kevin Stegemoller - corrected example schemas
4-APR-06	1.0	Carl Reed/Panagiotis (Peter) A. Vretanos to get document ready for release as an adopted OpenGIS Implementation Specification
07-MAR-2006	0.0.34	D. Arctur corrected table references and clause cross-references. P. Vretanos: Added recommended reading from GML specification.
06-MAR-2006	0.0.33	A. Whiteside changes identifying several typos resulting in incorrect element declarations as well as a number of editorial changes fixing some unclear wording, etc…
16-FEB-2006	0.0.32	Clemens Portele editorial edits integrated into the specification. Added stronger **bolded** language to clause 8.3.1 to indicate that when OGC comes up with a firm decision on the issue of validating against gml.xsd –vs- gmlsf.xsd then this document will be ammeded accordingly.
13-FEB-2006	0.0.31	Added statement to clause 8.2.1 to address the OGC decision from November TC that says "going forward all first and second level revision changes for all OGC XML schemes (except profiles) require a new namespace".
12-FEB-2006	0.0.30	Apply review comments from A. Whiteside and S. Keens.
28-JAN-2006	0.0.29	Editorial reorganization that separates the requirements for each compliance level into their own clauses. This should make it easier for developers to figure out precisely what the requirements are for each compliance level. There has been no change to the technical content of the document.
09-NOV-2005	0.0.28	Revised clause 2. Renamed conformance levels to SF-0, SF-1, SF-2. Numbered all normative references and renumbered informative references (Bibliography). Fixed reference in document to normative and informative references. Updates clause describing the importation of the gml schemas as per the consensus reached on 05-065r3. Added support for feature collection to SF-0 and SF-1. Updated conformance tests accordingly.
08-NOV-2005	0.0.27	Changed all references to "class" to "level". Changed all references to sfgml to be gmlsf. Labelled each subclause and ANNEX in Table 2 with the level it applies to.
30-OCT-2005	0.0.25	Update capabilities matrix to indicate that remote values are supported through gml:ReferenceType but not gml:AssocationAttributeGroup.
29-OCT-2005	0.0.24	Rename specification back to GML simple features profile, mandate use of Member rather and Members for aggregate geometries, update clause 8.4.3 to ensure that all types are listed, a separate GML schema for level 0 or 1 and 2, re-import schemas and examples into document
25-OCT-2005	0.0.23	Put conformance level description in correct clause. Indicate which conformance tests shall be tested for each conformance level. Fix typos identified by Arctur.
18-OCT-2005	0.0.22	Fix broken table and section references. Add Hydro model example in ANNEX C.
17-OCT-2005	0.0.21	Rename document to be "GML basic features profile". Add three conformance levels. Add new version of gmlsf.xsd schema based on twiki comments.
11-OCT-2005	0.0.20	T.B.D. Multiple review comments from RWG.

Date	Release	Description
07-JUL-2005	0.0.19	Editorial comment from Carl Reed.
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.gmlsf.xsd
23-APR-2005	0.0.16	Review comments from David Burggraf. Added profile schema gmlsf.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 John Davidson 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	Edric Keighan review comments
17-Jan-2003	0.0.1	Initial version

Bibliography

The following references are informative, not normative.

[1] OGC 99-049, OpenGIS® Simple Features Specification for SQL, Version 1.1

[2] OGC 04-024, OpenGIS® Web Map Service Implementation Specification version 1.3, 2004

[3] OGC 02-004, Patterns in GML, 2002

[4] OGC 01-025, Simple GML: Geography Markup Language (GML) 2.0 Implementation Profile, 2001

[5] OGC 05-010, URNs of definitions in ogc namespace, 2005

[6] 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

[7] ISO/IEC JTC 1/SC 32/WG 4, SQL Multimedia and Application Packages (SQL/MM) – Part 3: Spatial, Mark Ashworth, March 1999

[8] ISO 19125-1: Geographic information – Simple feature access - Part 1: Common architecture

[9] ISO 19125-2: Geographic information – Simple feature access - Part 2: SQL option

[10] ISO/DIS 19136: Geographic information - Geography Markup Language

[11] OGC 05-037, OpenGIS® Geography Markup Language Implementation Specification version 3.2

[12] OGC 04-094, OpenGIS® Web Feature Service Implementation Specification version 1.1, 2004.