Open GIS Consortium, Inc.

Date: 2003-01-29

Reference number of this OpenGIS® project document: **OGC 02-023r4**

Version: 3.00

Category: OpenGIS® Implementation Specification

Editors: Simon Cox , Paul Daisey, Ron Lake, Clemens Portele, Arliss Whiteside

OpenGIS® Geography Markup Language (GML) Implementation Specification

Copyright notice

Document type: OpenGIS® Implementation Specfciation
Document subtype: -
Document stage: Final
Document language: English

Contents

i. Preface

n/a

ii. Submitting organizations

CSIRO Australia

Galdos Systems

Interactive Instruments

BAE Systems

US Census

POSC

MapInfo

Oracle Corp

NTT Data

Laser-Scan Ltd.

iii. Contributors

CONTACT	COMPANY	ADDRESS	PHONE/FAX	EMAIL
John Bobbitt	POSC		+1 (713) 267-5174 +1 (713) 784-9219	bobbitt@posc.org
David Burggraf	Galdos Systems		+1-604-484-2750	dburggraf@galdosinc.com
Simon Cox	CSIRO	ARRC, PO Box 1130, Bentley, WA 6102, Australia	+61 8 6436 8639 +61 8 6436 8555	Simon.Cox@csiro.au
Paul Daisey	U.S. Census Bureau Geography Division		+1 (301) 763-4307 +1 (301)457-4710	pdaisey@geo.census.gov
John Herring	Oracle Corporation	1900 Oracle Way Reston, Virginia 20190 USA	+1(703)364-0689 +1(703)364-1364 fax	John.Herring@oracle.com
Sandra Johnson	MapInfo Corporation	1 Global View, Troy NY 12190 USA	+1-518-285-7275	Sandra_Johnson@mapinfo.com
Ron Lake	Galdos Systems	200 – 1155 W. Pender Street Vancouver, B.C. Canada V6E2-P4	+1-604-484-2750	rlake@galdosinc.com
Richard Martell	Galdos Systems		+1-604-484-2750	rmartell@galdosinc.com
Akifumi Nakai	NTT Data			nakaia@nttdata.co.jp
Clemens Portele	Interactive Instruments	Trierer Strasse 70-72 53115 Bonn	+49 (228) 9 14 10 73	portele@interactive-instruments.de
Milan Trninic	Galdos		+1-604-484-2750	mtrninic@galdosin

	Systems			c.com
Paul Watson	Laser-Scan Ltd	Science Park, Milton Rd Cambridge CB4 0FY United Kingdom	+44 (0) 1223 420414	paulw@lsl.co.uk
Arliss Whiteside	BAE Systems Mission Solutions		+1 858-592-1608	Arliss.Whiteside@ baesystems.com

iv. Revision history

Date	Release	Author	Paragraph modified	Description
08/19/02	02-023	GML 3.0 RWG	All	First version
09/06/02	02-023r1	GML 3.0 RWG	All	Document update
11/18/02	02-023r2	GML 3.0 RWG	All	Document update for review by OGC TC
12/02/02	02-023r3	GML 3.0 RWG	All	Additional update for review by OGC TC
12/18/02	02-023r4	GML 3.0 RWG	All	Final changes and nits, version submitted to the OGC TC

v. Changes to the OpenGIS® Abstract Specification

The OpenGIS® Abstract Specification requires the following changes to accommodate this OpenGIS® standard.

1. Topic 0. and ISO 19103: “velocity” measure should be “speed”. In standard physics/applied mathematics usage, velocity is a vector quantity, not a scalar.
2. Several topics: in GML we have found it convenient for an identified object to be able to carry several “names” – i.e. labels either assigned by external authorities or commonly used for an object. These may be distinguished from each other by noting the “codeSpace” for a name, which allows a consumer of the information to select the appropriate name for them to use. They are not distinguished from each other as “aliases” or “alternative names” since this concept is not considered

to be robust. A common mechanism has been used in GML, which might need to be reflected in the Abstract Spec.

vi. Recommended changes to the ISO Specifications

3. 19108 Temporal:
    a. A frame attribute is needed on TM_Position;
    b. in TM_OrdinalEra the begin and end properties should be associations with TM_Instant rather than direct DateTime, to support the description of contiguous intervals by referring to the begin or end values represented in a single place;
    c. in TM_CoordinateSystem the origin property should be an association with TM_Instant rather than direct DateTime;
    d. in TM_CoordinateSystem it is a requirement to be able to indicate the increment direction – for example, in geological time scales, it is conventional to measure backwards from the present.

Foreword

Attention is drawn to the possibility that some of the elements of OGC 02-023r4 may be the subject of patent rights. Open GIS Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

This third edition cancels and replaces the second edition (OGC 02-069). The second edition has been technically revised and extended.

This specification uses the Recommended XML encoding of Coordinate Reference System definitions prepared by the CRS SIG of OGC (document 02-036r6 or its most recent successor).

The annexes A, B and C are normative. All other annexes are informative.

This specification is based on a number of other specifications developed within or outside of OGC. See the chapter of normative references for a list.

Introduction

Geography Markup Language is an XML grammar written in XML Schema for the modelling, transport, and storage of geographic information.

The key concepts used by Geography Markup Language (GML) to model the world are drawn from the OGC Abstract Specification (available online: http://www.opengis.org/techno/abstract.htm).

GML provides a variety of kinds of objects for describing geography including features, coordinate reference systems, geometry, topology, time, units of measure and generalized values.

A geographic feature is "an abstraction of a real world phenomenon; it is a geographic feature if it is associated with a location relative to the Earth". So a digital representation of the real world can be thought of as a set of features. The state of a feature is defined by a set of properties, where each property can be thought of as a {name, type, value} triple.

The number of properties a feature may have, together with their names and types, are determined by its type definition. Geographic features with geometry are those with properties that may be geometry-valued. A feature collection is a collection of features that can itself be regarded as a feature; as a consequence a feature collection has a feature type and thus may have distinct properties of its own, in addition to the features it contains.

Geographic features in GML include coverages and observations as subtypes.

A coverage is a sub-type of feature that has a coverage function with a spatial domain and a value set range of homogeneous 2 to n dimensional tuples. A coverage can represent one feature or a collection of features "to model and make visible spatial relationships between, and the spatial distribution of, earth phenomena."

An observation models the act of observing, often with a camera, a person or some form of instrument ("an act of recognizing and noting a fact or occurrence often involving measurement with instruments"). An observation is considered to be a GML feature with a time at which the observation took place, and with a value for the observation.

A reference system provides a scale of measurement for assigning values "to a location, time or other descriptive quantity or quality".

A coordinate reference system consists of a set of coordinate system axes that is related to the earth through a datum that defines the size and shape of the earth. Geometries in GML indicate the coordinate reference system in which their measurements have been made. The “parent” geometry element of a geometric complex or geometric aggregate makes this indication for its constituent geometries.

A temporal reference system provides standard units for measuring time and describing temporal length or duration. Following ISO 8601, the Gregorian calendar with UTC is used in GML as the default temporal reference system.

A Units of Measure (UOM) dictionary provides definitions of numerical measures of physical quantities, such as length, temperature, and pressure, and of conversions between UOMs

OpenGIS® Geography Markup Language (GML) Implementation Specification – Version 3.0

1 Scope

The Geography Markup Language (GML) is an XML encoding for the modeling, transport and storage of geographic information including both the spatial and non-spatial properties of geographic features. This specification defines the XML Schema syntax, mechanisms, and conventions that:

- Provide an open, vendor-neutral framework for the definition of geospatial application schemas and objects;
- Allow profiles that support proper subsets of GML framework descriptive capabilities;
- Support the description of geospatial application schemas for specialized domains and information communities;
- Enable the creation and maintenance of linked geographic application schemas and datasets;
- Support the storage and transport of application schemas and data sets;
- Increase the ability of organizations to share geographic application schemas and the information they describe.

Implementers may decide to store geographic application schemas and information in GML, or they may decide to convert from some other storage format on demand and use GML only for schema and data transport.

> Note: The model is conformant with the parts of the Abstract Specification of OGC and the ISO 19100 series which are listed in the normative references. For other parts of this specifications no abstract specification was available.

2 Conformance

Conformance with this specification shall be checked using all the relevant tests specified in Annex A (normative). The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in ISO 19105: Geographic information — Conformance and Testing.

In order to conform to this OpenGIS® interface standard, a software implementation shall choose to implement:

a) Any one of the conformance levels specified in Annex B (normative).

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of OGC 02-023r2. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on OGC 02-023r2 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies.

Abstract Specification Topic 0: Overview, OGC document 99-100r1

Abstract Specification Topic 1: Feature Geometry, OGC document 01-101

Abstract Specification Topic 2: Spatial referencing by coordinates, OGC document 02-102

Abstract Specification Topic 5: The OpenGIS Feature, OGC document 99-105r2

Abstract Specification Topic 8: Relations between Features, OGC document 99-108r2

Abstract Specification Topic 10: Feature Collections, OGC document 99-110

ISO PDTS 19103, *Geographic Information – Conceptual Schema Language*

ISO DIS 19107, *Geographic Information – Spatial Schema*

ISO DIS 19108, *Geographic Information – Temporal Schema*

ISO DIS 19109, *Geographic Information – Rules for Application Schemas*

ISO DIS 19118, *Geographic Information – Encoding*

ISO DIS 19123, *Geographic Information – Coverages*

ISO DIS 19115, *Geographic Information – Metadata*

Recommended XML encoding of coordinate reference system definitions, OGC document 02-036r6 or its most recent successor.

RFC2396, Uniform Resource Identifiers (URI): Generic Syntax. (August 1998).

Available [Online]: ftp://www.ietf.org/rfc/rfc2396.txt

Available [Online]: http://www.faqs.org/rfcs/rfc2396.html

RFC2732, Format for Literal IPv6 Addresses in URLs. (December 1999).

Available [Online]: http://www.ietf.org/rfc/rfc2732.txt

The Schematron Assertion Language 1.5. Rick Jelliffe 2002-10-01

Available [Online]:
http://www.ascc.net/xml/resource/schematron/Schematron2000.html

XLink, XML Linking Language (XLink) Version 1.0. W3C Recommendation (27 June 2001).

Available [Online]: http://www.w3.org./TR/xlink/

XMLName, Namespaces in XML. W3C Recommendation (14 January 1999).

Available [Online]: http://www.w3.org/TR/1999/REC-xml-names-19990114/

XMLSchema-1, XML Schema Part 1: Structures. W3C Recommendation (2 May 2001).

Available [Online]: http://www.w3.org/TR/xmlschema-1/

XMLSchema-2, XML Schema Part 2: Datatypes. W3C Recommendation (2 May 2001).

Available [Online]: http://www.w3.org/TR/xmlschema-2/

Xpointer, XML Pointer Language (XPointer) Version 1.0. W3C Working Draft (16 August 2002).

Available [Online]: http://www.w3.org./TR/xptr/>

XML Base

Available [Online]: http://www.w3.org/TR/xmlbase/

XML, Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation 6 October 2000

Available [Online]: http://www.w3.org/TR/REC-xml

SVG, Scalable Vector Graphics (SVG) 1.0 Specification. W3C Recommendation (04 September 2001).

Available [Online]: http://www.w3.org./TR/SVG/

SMIL, Synchronized Multimedia Integration Language (SMIL 2.0). W3C Recommendation (07 August 2001)

Available [Online]: http://www.w3.org./TR/smil20/

4 Terms and definitions

4.1 application schema

conceptual schema for data required by one or more applications [ISO 19101]

4.2 GML application schema

A GML application schema is an XML Schema written according to the GML rules for application schemas (see Clause 7.) and which defines a vocabulary of geographic objects for a particular domain of discourse

4.3 association

Association is a structural relationship that describes a set of links, in which a link is a connection among objects; the semantic relationship between two or more classifiers that involves the connections among their instances (Booch, 1999)

semantic relationship between two or more classifiers that specifies connections among their instances [ISO 19103]

4.4 attribute

Attribute (UML) is a named property of a class that describes the range of values that instances of the property may hold.(Booch,1999)

Attribute (XML) is an information item in the XML Information Set [Infoset] .

NOTE: The syntax of an XML attribute is

Attribute::=   Name Eq AttValue

An attribute typically acts as an XML element modifier (e.g. <Road gml:id = “r1” /> here gml:id is an attribute.

In this document an attribute is an XML attribute unless otherwise specified

4.5 boundary

set that represents the limit of an entity [ISO 19107]

4.6 category

One of a set of classes in a classification scheme.

4.7 child

In XML, refers to an element c that is in the content of another element p, its parent, but is not in the content of any other element in the content of p.

4.8 codelist

Value domain including a code for each permissible value

4.9 codespace

Rule or authority for a code, name, term or category. Dictionary, authority, codelist or lexical pattern, etc.

4.10 complexType

In XML Schema refers to the one of the schema components for the definition of the content model of an XML element. See also simpleType

> *NOTE: An XML Schema complexType is a content model for an element that can have attributes and/or elements. SimpleType content models cannot have element content nor attributes.*

4.11 composite curve

sequence of **curves** such that each curve (except the first) starts at the end point of the previous curve in the **sequence** [ISO 19107]

> *NOTE A composite curve, as a set of direct positions, has all the properties of a curve.*

4.12 composite solid

connected **set** of **solids** adjoining one another along shared **boundary surfaces** [ISO 19107]

> *NOTE A composite solid, as a set of direct positions, has all the properties of a solid*

4.13 composite surface

connected **set** of **surfaces** adjoining one another along shared **boundary curves** [ISO 19107]

> *NOTE A composite surface, as a set of direct positions, has all the properties of a surface.*

4.14 closure

union of the **interior** and **boundary** of a **topological** or **geometric object** [ISO 19107]

4.15 coordinate

one of a **sequence** of numbers designating the position of a **point** in N-dimensional space [ISO 19107]

NOTE In a coordinate reference system, the numbers must be qualified by units

4.16 coordinate reference system

coordinate system that is related to the real world by a datum [ISO 19111]

4.17 coordinate system

set of (mathematical) rules for specifying how **coordinates** are to be assigned to **points** [ISO 19107]

4.18 coordinate tuple

Tuple composed of coordinates

4.19 count

Integer that measures the number of occurrences of some phenomena (e.g. photo count, species count)

4.20 coverage

Feature that incorporates an explicit mapping from a spatial-temporal domain to a range set. The range set values are often referred to as attribute values

4.21 curve

1-dimensional **geometric primitive**, representing the continuous image of a line

NOTE The boundary of a curve is the set of points at either end of the curve. If the curve is a cycle, the two ends are identical, and the curve (if topologically closed) is considered to not have a boundary. The first point is called the start point, and the last is the end point. Connectivity of the curve is guaranteed by the "continuous image of a line" clause. A topological theorem states that a continuous image of a connected set is connected.

4.22 data type

specification of a value domain with operations allowed on values in this domain [ISO 19103]

EXAMPLE Integer, Real, Boolean, String, Date (conversion of a data into a series of codes).

NOTE Data types include primitive predefined types and user-definable types.

4.23 datum

Zero or reference value/point/line/surface for a measurement scale.

parameter or set of parameters that may serve as a reference or basis for the calculation of other parameters [ISO 19107]

NOTE A datum defines the position of the origin, the scale, and the orientation of the axes of a coordinate system.

NOTE A datum may be a geodetic datum, a vertical datum or an engineering datum.

4.24 direct position

position described by a single set of coordinates within a coordinate reference system [ISO 19107]

4.25 domain

Set on which a mathematical **function** is defined (f:A→B, then A is the domain of the function f)

Domain as in domain of discourse refers to a subject or area of interest.

well-defined **set** [ISO 19107]

4.26 edge

1-dimensional topological primitive. Also called a topological element. [ISO 19107]

4.27 element

In XML, an element is an information item in the XML information Set. In an XML instance document, an element is everything bounded by a pair of XML tags (including the opening and closing tag)

NOTE: (From XML Information Set Specification) Definition: Each XML document contains one or more elements, the boundaries of which are either delimited by start-tags and end-tags, or, for empty elements, by an empty-element tag. Each element has a type, identified by name, sometimes called its "generic identifier" (GI), and may have a set of attribute specifications.] Each attribute specification has a name and a value

4.28 exterior

difference between the universe and the **closure** [ISO 19107]

4.29 face

2-dimensional **topological primitive** [ISO 19107]

NOTE: The geometric realization of a face is a surface. The boundary of a face is the set of directed edges within the same topological complex that are associated to the face via the boundary relations. These can be organized as rings

4.30 feature

abstraction of real world phenomena [ISO 19107]

NOTE: A feature may occur as a type or an instance. Feature type or feature instance should be used when only one is meant

4.31 feature relationship

association between **features** [ISO 19103]

4.32 function

A mathematical relationship between two sets. For any two sets A,B a function is a relation on A,B such that for each element in the domain there is only one element in the range

4.33 geodetic datum

datum describing the relationship of a coordinate system to the Earth [ISO 19111]

NOTE: In most cases, the geodetic datum includes an ellipsoid definition.

4.34 geometric object

spatial object representing a geometric set [ISO 19107]

4.35 geometric set

set of direct positions [ISO 19107]

4.36 geometry

Class of object that describes the location, shape or extent of a geographic feature. Various types of geometric classes (each of which are geometries) are described in the GML 3.0 geometry schemas.

4.37 geometry property

Property of a GML feature that describes some aspect of the geometry of the feature. The geometry property name is the role of the geometry in relation to the feature.

4.38 GML property

a child element of a GML object.

NOTE: It corresponds to feature attribute and feature association in ISO 19109. If a GML ***property of a feature*** *has an xlink:href attribute that references a feature, the* ***property*** *represents a feature association.*

4.39 graph of a function

Set of ordered pairs of the form (x,f(x)), x in some set A where f is the function and A is some set. This is the graph of the function f on the set A

4.40 grid

network composed of two or more sets of **curves** in which the members of each set intersect the members of the other sets in a systematic way [ISO 19123]

NOTE: The curves partition a space into grid cells.

4.41 iff

if and only if

4.42 interior

set **of all** direct positions **that are on a** geometric object **but which are not on its** boundary [ISO 19107]

4.43 line string

Piece-wise curve composed of straight-line segments

4.44 measure

A value described using a numeric amount with a scale or using a scalar reference system. When used as a noun, measure is a synonym for quantity.

4.45 measurand

Phenomenon or property that is subject to observation with a result expressed as a quantity or measure, i.e. on a numeric scale.

4.46 namespace

Collection of names, identified by an URI reference which are used in XML documents as element names and attribute names [XML]

4.47 node

1-d topological primitive or topological element

4.48 object

entity with a well defined boundary and identity that encapsulates state and behaviour [ISO 19107]

A GML object is an XML document element of a type derived from AbstractGMLType.

4.49 observable

Phenomenon or property that is subject to observation

4.50 p-chain

Formal sum of p-cells or p-simplicies.

NOTE: Corresponds to TP_Expression in ISO 19107

4.51 point

0-dimensional **geometric primitive**, representing a position

NOTE: The boundary of a point is the empty set.

4.52 polygon

2-dimensional geometric primitive described by a single outer boundary and one or more inner boundaries

4.53 quantity

A value described using a numeric amount with a scale or using a scalar reference system. Quantity is a synonym for measure when the latter is used as a noun.

4.54 range

In the definition of a mathematical function (f:A→B), the set B is called the range of the function. In programming languages, a range is a kind of data type sub-typed from some numeric type as integers or floating point numbers and defined by a pair of numbers from the numeric super type. In XML Schema, one of the kinds of derived simpleTypes.

4.55 rectified grid

Grid whose **points** have **coordinates** relative to some **coordinate reference system**

4.56 schema

In general, a schema is an abstract representation of an **object's** characteristics and relationship to other **objects.** An XML schema represents the relationship between the attributes and elements of an XML object (for example, a document or a portion of a document)

4.57 semantic type

A category of objects that share some common characteristics and are thus given an identifying type name in a particular domain of discourse.

4.58 set

unordered collection of related items (**objects** or values) with no repetition [ISO 19107]

4.59 simpleType

In XML Schema, one of the means of specifying the content model of an XML element or attribute. See complexType.

> *NOTE: simpleTypes in XML Schema cannot contain attribute or element content. Simple Types have well-defined lexical representation that must be derived from XML Schema built-in simple types.*

4.60 spatial object

object used for representing a spatial characteristic of a feature [ISO 19107]

4.61 tag

In XML, the text bounded by angle brackets (e.g. <Road>).

> *NOTE: A tag with no forward slash (e.g. <Road>) is called a start tag (also opening tag), and one with a forward slash (e.g. </Road> is called an end tag (also closing tag).*

4.62 topological element

An element (XML) in a GML instance document that is a topological primitive in the sense of ISO 19107.

4.63 topological object

spatial object representing spatial characteristics that are invariant under continuous transformations [ISO 19107]

4.64 topology

Class objects that represent the topology of geographical features and which include as sub-classes the **topological elements**

Replace with 19107 definition – there is none in 19107

4.65 tuple

An ordered list of values

4.66 type

A class in a classification system

NOTE See also data type.

4.67 Uniform Resource Identifier (URI)

Simple and extensible means for identifying a resource; a short string or address; classified as a name, a locator, or both [RFC 2396]

NOTE: The general syntax is <scheme>::<scheme-specific-part>. The hierarchical syntax with a namespace is <scheme>://<authority><path>?<query>

4.68 value

Target of a GML property

NOTE: This is the same as the meaning of value in RDF (Resource Description Framework).

5 Conventions

5.1 Depecated parts of previous versions of GML

The verb "**deprecate**" provides notice that the referenced portion of the specification is being retained for backwards compatibility with earlier versions but may be removed from a future version of the specification without further notice.

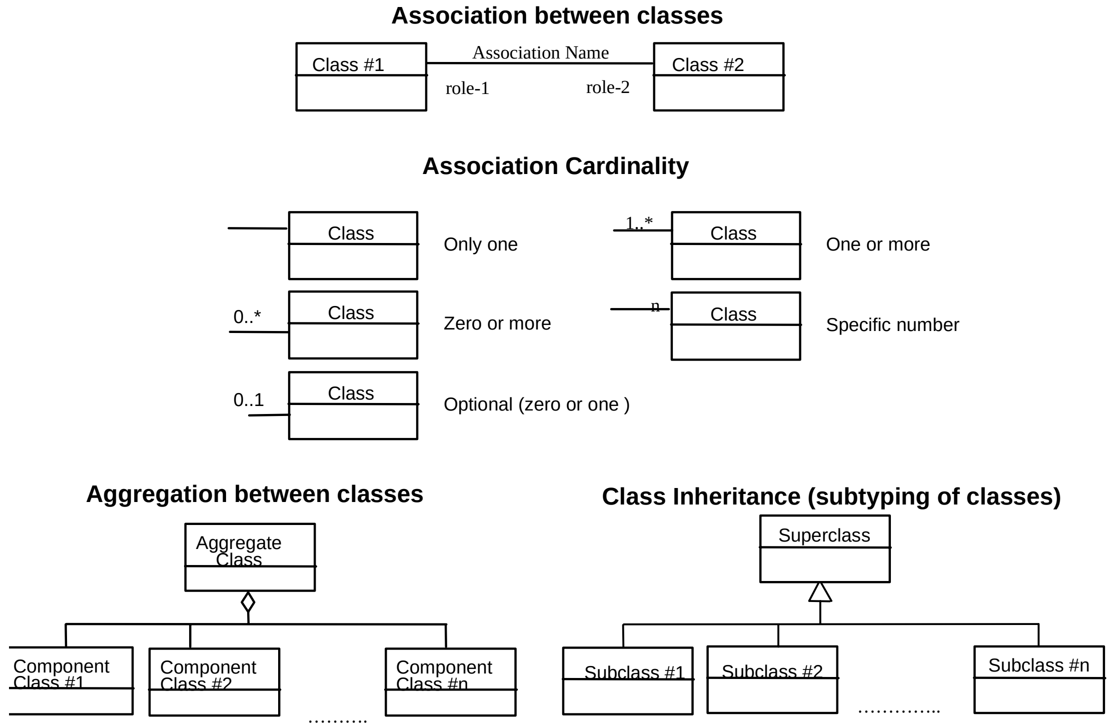
5.2 Symbols (and abbreviated terms)

The following symbols and abbreviated terms are used in this document:

DTD	Document Type Definition
EPSG	European Petroleum Survey Group
GIS	Geographic Information System
GML	Geography Markup Language
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
OGC	Open GIS Consortium
RDF	Resource Description Framework
RFC	Request for Comments
SMIL	Synchronized Multimedia Integration Language
SOAP	Simple Object Access Protocol
SVG	Scalable Vector Graphics
UML	Unified Modeling Language
URL	Uniform Resource Locator
WFS	Web Feature Service
XML	Extensible Markup Language
XSLT	eXtensible Stylesheet Language - Transformations
0D	Zero Dimensional
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional

5.3 UML Notation

Many diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram. The UML notations used in this standard are described in the diagram below.



OGC cleaners - Get non-truncated version from Arliss

Figure 5.3-1 — UML notation

In this standard, the following three stereotypes of UML classes are used:

a) <<DataType>> A descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A DataType is a class with no operations whose primary purpose is to hold the information.

b) <<CodeList>> is a flexible enumeration that uses string values for expressing a list of potential values.

c) <<Enumeration>> is a fixed list of valid identifiers of named literal values. Attributes of an enumerated type can only take values from this list.

d) <<Union>> is a list of attributes. The semantics is that only one of the attributes can be present at any time.

In this standard, the following standard data types are used:

a) CharacterString – A sequence of characters (in general this data type is mapped to "string" in XML Schema)

b) Integer – An integer number (in general this data type is mapped to "integer" in XML Schema)

c) Real – A floating point number (in general this data type is mapped to "double" in XML Schema)

d) Boolean – A value specifying TRUE or FALSE (in general this data type is mapped to "boolean" in XML Schema)

5.4 XML Schema

The normative parts of the specification use the W3C XML Schema language to describe the grammar of conformant GML data instances. XML Schema is a rich language with many capabilities and subtleties. While a reader who is unfamiliar with XML Schema may be able to follow the description in a general fashion, this specification is not intended to serve as an introduction to XML Schema. In order to have a full understanding of this specification it is necessary for the reader to have a reasonable knowledge of XML Schema.

6 Overview of GML Schemas

6.1 Review of GML version 2 – XML for Simple Features

The previous version (2.1.2) of GML was concerned with what the OGC calls simple features: features whose geometric properties are restricted to 'simple' geometries for which coordinates are defined in two dimensions and the delineation of a curve is subject to linear interpolation. These simple features in GML could be used to represent real-world phenomena. The task to which a digital representation will ultimately be put guides the classification of real world phenomena, which in turn determines the feature types that need to be defined.

For example, a city could be represented as a feature collection where the individual features represent such things as rivers, roads and colleges. Each of these feature types would have named, typed properties. The River feature type might have a property called name whose value must be of the type 'string'. It is common practice to refer to the typed property; thus the River feature type is said to have a string property called name. Similarly, the Road feature type might have a string property called classification and an integer property called number. Properties with simple types (e.g. integer, string, float, boolean) are collectively referred to as simple properties.

The features required to represent a city might have geometry-valued properties as well as simple properties. Just like other properties, geometric properties must be named. So the River feature type might have a geometric property called centerLineOf and the Road feature type might have a geometric property called linearGeometry. Just as it is common to have multiple simple properties defined on a single feature type, so too a feature type may have multiple geometric properties; the College feature type might have both a point property called location and a polygon property called campus.

6.2 GML version 3 – more than Simple Features, plus ISO conformance

This version (3.0) of GML addresses the following needs that were not addressed or adequately met by the previous version:

- represent geospatial phenomena in addition to simple 2D linear features, including features with complex, non-linear, 3D geometry, features with 2D topology, features with temporal properties, dynamic features, coverages, and observations;

- provide more explicit support for properties of features and other objects whose value is complex

- represent spatial and temporal reference systems, units of measure and standards information;

- use reference system, units and standards information in the representation of geospatial phenomena, observations, and values;

- represent default styles for feature and coverage visualization;

- conform with other standards, including
    - ISO DIS 19107 Geographic Information – Spatial Schema
    - ISO DIS 19108 Geographic Information – Temporal Schema
    - ISO DIS 19118 Geographic Information – Encoding
    - ISO DIS 19123 Geographic Information – Coverages

The expansion of GML to meet these needs is reflected in base schemas for GML version 3 that are over eight times as large as the base schemas for GML version 2.1.2. However, few applications will use all of the definitions added to GML version 3. Implementers may use a selective subset of the GML version 3 schemas sufficient to support the definitions in their application schemas. Methods for modular use of GML are discussed in clause 7.13; examples are provided in Annex E; a schema subsetting tool is provided in Annex F.

6.3 Backward compatibility

GML version 3.0 maintains backward compatibility for GML version 2.1.2 **instance documents** by preserving, but deprecating, some schema components that have been replaced by different constructs in the current version. GML version 2.1.2 application schemas will in most cases require only minor changes to upgrade to GML version 3.0, removing the use of such deprecated schema components. The deprecated GML schema components may be eliminated without notice from a future GML version.

6.4 GML 3 Application Schemas

Designers of GML Application schemas may extend or restrict the types defined in the GML base schemas to define appropriate types for an application domain. They may also use non-abstract elements and attributes from the GML base schemas as-is in an application schema if no changes are required. GML Application schemas may be constructed by hand using a text editor or specialized XML/Schema editor, in effect using XML/Schema as a conceptual schema language. GML Application schemas may also be constructed as part of a model driven process by automated translation to XML/Schema from conceptual models defined in other conceptual schema languages such as UML. In both cases, conformance with the requirements of this specification requires application schema use of all of the applicable GML base schema constructs, directly or by specialization, application schema validity according to the rules for XML/Schema, and application instance document data conformance to the application schema definitions verification by a validating XML parser. How the GML application schemas were produced is irrelevant for conformance to the requirements of this specification.

6.5 The GML 3 data model and how this is factored into schema documents

GML defines the various entities such as features, geometries, topologies etc. through a hierarchy of GML objects as shown in the UML diagram in Figure 6.5-1.

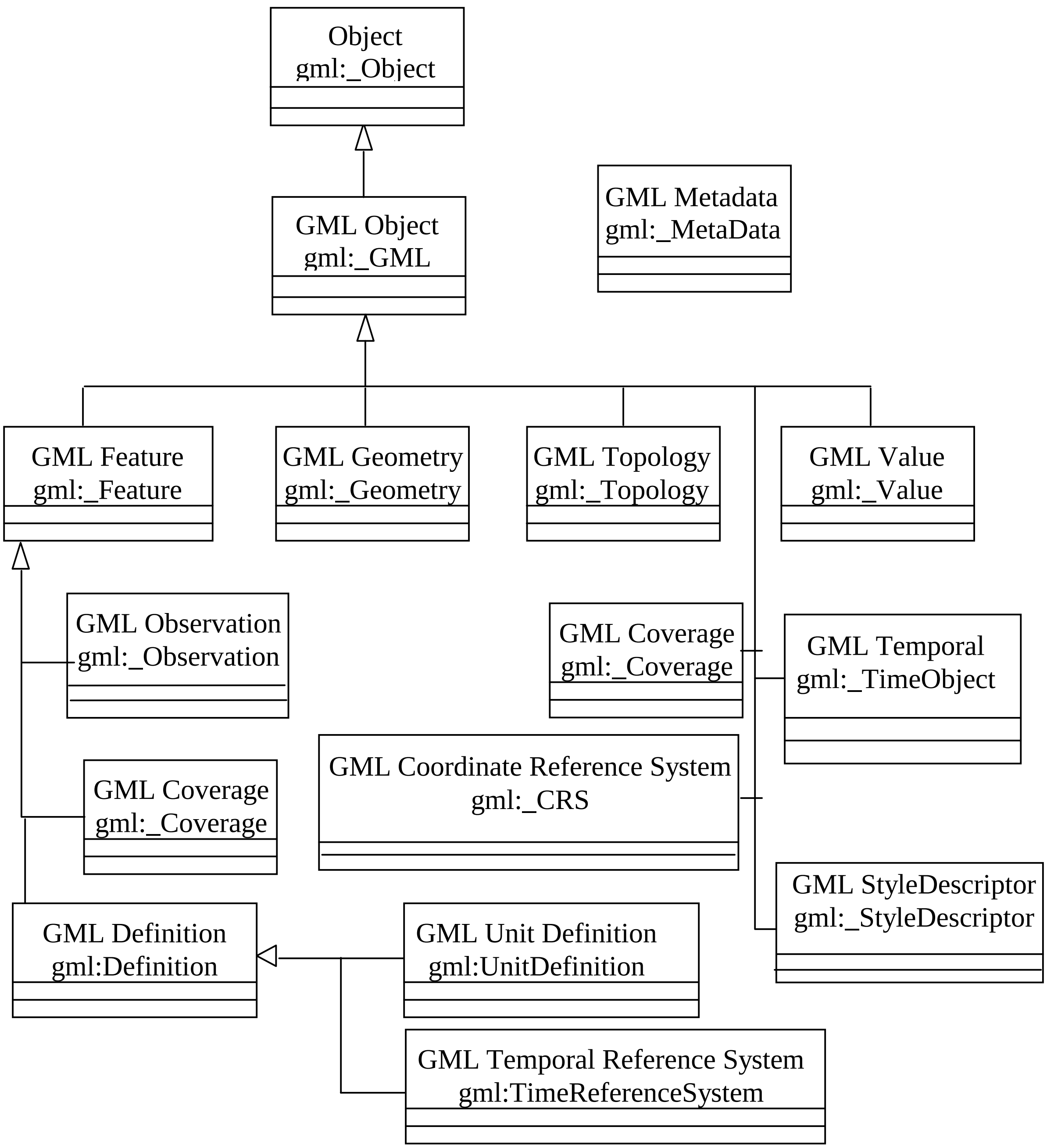


Figure 6.5-1 — GML Class Hierarchy

The items in Figure 6.5-1 with a preceding underscore (e.g. Feature) can be viewed as representative object instances of the class. The element gml:_Feature, for example, is to be interpreted as "any GML feature".

The normative schemas of GML are organized in this specification around these GML object types.

OGC cleaners please check integrity of this list and references.

Clause 7.2.2 describes the schema gmlBase.xsd. This defines the root object (gml;_Object), the root of the GML class hierarchy (gml:_GML) and the root for MetaData property packages (gml:_MetaData).

Clause 7.3.1 describes the Xlink schema. This schema is an OGC implementation of the XLink specification using XML Schema. It may be replaced in some future release by an equivalent schema from the W3C.

Clause 7.3.2 defines the GML representation of some basic data types that are used in different GML Schemas. Most of these types are simple types or simple content types.

Clause 7.4.1 describes the feature schema which defines gml:_Feature and some derived components.

Clause 7.5 describes the geometry schemas that define gml:_Geometry and some derived components.

Clause 7.6 describes the Coordinate Reference System schemas the define the sub-types of gml:_CoordinateReferenceSystem, and the elements and types required to construct specific Coordinate Reference Systems.

Clause 7.7 describes the schemas for Topology which define gml:_Topology and some derived components.

Clause 7.8 describes the schemas for temporal constructs and for dynamic features.

Clause 7.9 describes the schemas for definitions and dictionaries

Clause 7.10 describes the schemas for the construction of units of measure, measures and value objects.

Clause 7.11 describes the schemas for the description of Direction.

Clause 7.12 describes the schemas for Observations.

Clause 7.13.4 describes the schemas for grid geometries.

Clause 7.13.5 describes the schemas for coverages. This describes gml:_Coverage and some derived components.

Clause 7.14 describes the schemas for the graphical default styling of GML features, geometries and topologies.

6.6 GML schemas to import into your GML 3 Application Schema

Most applications will make use of only a subset of the schemas described in this specification. Schema dependencies are discussed in greater detail in Clause 7.14. As a starting point, consider the following;

1. If you are modelling geographic features you will need the feature.xsd.

2. If your features have properties which make use of units of measure you will need to import one of the basicTypes.xsd, measure.xsd and valueObjects.xsd schema. , but you will not need the units.xsd schema unless you are defining units of measure such as appear in a units of measure dictionary.
3. If you need only simple 1D or 2D geometries you need only the geometryBasic0d1d and geometryBasic2d schemas. You need the additional geometry schemas only if you require support for complex, 3D or non-linear geometry types. This is discussed in greater detail in Clause 6.5.
4. You only need topology.xsd if your features have topology properties.
5. You need the CoordinateReferenceSystem schemas only if you are constructing Coordinate Reference System dictionary entries and those for supporting components (e.g. Prime Meridians, Geodetic Datums etc.)
6. You need the temporal schemas only if you are concerned with time dependent feature properties or dynamic features.
7. You need the coverage schemas only if you are constructing coverages (e.g. remotely sensed images, aerial photographs, soil distribution, digital elevation models).
8. You need the observation schema only if you are concerned with modelling acts of observation such as taking photographs or making measurements. In the latter case you will also likely use the valueObjects.xsd and measure.xsd schemas.
9. You need direction schema only to describe direction constructs such as compass bearings. The direction schema is included by the observation.xsd schema in order to support directed observations.
10. You need the defaultStyle.xsd only if you are concerned with the description of graphical styles for features, geometries and topologies.
11. The Metadata element in gmlBase.xsd is used to define packages of metadata properties that can be attached to any resource including GML features, geometries, and topologies.

In many applications you will only need to import the feature.xsd as this transitively imports the simple geometry schemas and gmlBase.xsd. For a thorough discussion of schema dependencies and modularity see Clause 7.14.

7 Description of normative GML schemas

7.1 Introduction, namespaces, versioning

This clause describes the normative GML schema documents and explains their contents, structure and dependencies.

The components described in the xlinks.xsd schema document are in the http://www.w3.org/1999/xlink namespace, for which the prefix **xlink** is normally used.

All the other schema documents describe components in the http://www.opengis.net/gml namespace, for which the prefix **gml** is normally used.

Each schema document in GML carries a version attribute as defined in the XML Schema Recommendation. The format of the version attribute string is x.yy where x denotes the major version number and yy denotes a minor version number for that document. The current version is thus 3.00.

7.2 GML model and syntax

7.2.1 Overview of model and syntax

7.2.1.1 GML schema documents

GML schema documents are XML/Schemas that define XML types and XML elements to encode GML objects with identity, elements to encode GML properties of those objects, and XML attributes qualifying those properties.

A GML object is an XML element of a type derived directly or indirectly from AbstractGMLType. From this derivation, a GML object may have a gml:id attribute.

A GML property may not be derived from AbstractGMLType, may not have a gml:id attribute, or any other attribute of XML type ID.

An element is a GML property if and only if it is a child element of a GML object.

No GML object may appear as the immediate child of a GML object.

Consequently, no element may be both a GML object and a GML property.

> NOTE: In this version of GML, the use of additional XML attributes in a GML application schema is discouraged.

7.2.1.2 GML instance documents

GML uses an explicit syntax to instantiate the Feature model defined in the OGC abstract specification in an **XML instance document**. 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. The value of a property may be simple, or it may be a feature or other complex object. A property element may contain its value as content encoded inline, or reference its value with a simple XLink.

The result is a layered XML document, in which XML elements corresponding to features, objects or values occur interleaved with XML elements corresponding to the properties that relate them. The function of a feature, object or value in context can always be determined by inspecting the name of the property element which directly contains it, or which carries the reference to it.

7.2.1.3 Lexical conventions

There are several lexical conventions used in GML to assist in human comprehension of GML instances and schemas:

- objects are instantiated as XML elements with a conceptually meaningful name in UpperCamelCase
- properties are instantiated as XML elements whose name is in lowerCamelCase
- abstract elements have an underscore prepended to their _name
- the names of XML types are mainly in UpperCamelCase ending in the word “Type”
- abstract XML types have the word “Abstract” prepended.

7.2.2 gmlBase schema

The types and elements used to establish the GML model and syntax are described in the schema listed in Annex C. The schema is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:gmlBase:v3.0

7.2.2.1 Goals of base schema

The schema document **gmlBase.xsd** defines components that establish the GML Model and Syntax: in particular

- A root XML type from which XML types for all GML objects should be derived
- A pattern and components for GML properties
- Patterns for collections and arrays, and components for generic collections and arrays
- Components for associating metadata with GML objects
- Components for constructing definitions and dictionaries

7.2.2.2 gml:_GML, gml:_Object

The most basic components for constructing identifiable GML objects are described in the schema as follows:

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

```
<complexType name="AbstractGMLType" abstract="true">
  <sequence>
    <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
    <element ref="gml:description" minOccurs="0"/>
    <element ref="gml:name" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute ref="gml:id" use="optional"/>
</complexType>
```

The abstract element gml:_GML is "any GML object having identity". It acts as the head of an XML Schema substitution group, which may include any element which is a GML feature, or other object, with identity. This is used as a variable in content models in GML core and application schemas. It is effectively an abstract superclass for all GML objects.

The pairing of gml:_GML and gml:AbstractGMLType shows a basic pattern used in the GML schemas, whereby each GML object type is represented by a global element declaration, which has an associated XML Schema type definition. The name of an element representing a GML Object indicates the conceptual meaning of the object. Generic element names in GML include gml:_Object, gml:_GML, gml:_Feature, gml:_Value, gml:_Coverage, gml:_Topology and gml:_CRS. These other generic elements representing objects are defined elsewhere in this specification.

The child XML elements and XML attributes of a GML object are properties of that object. Thus an object represented by an gml:_GML element has four properties: metaDataProperty, description, name and id. These are described in clause 7.2.2.4.

An additional abstract utility element gml:_Object is declared as follows:

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

This element has no type defined, and is therefore implicitly an XML Schema "anyType". It is used as the head of an XML Schema substitution group which unifies complexContent and certain simpleContent elements used for datatypes in GML, including the gml:_GML substitution group.

> NOTE: gml:_Object is defined primarily to act as a variable in certain aggregate patterns where it is necessary to allow either elements in the gml:_GML substitution group, or certain complexContent or simpleContent elements to be valid in an instance.

A GML dataset (also called a data instance or data document) is a GML object or a collection of GML objects.

7.2.2.3 GML properties

7.2.2.3.1 Overview

The term “property” is used to refer to a GML property, which is any characteristic of a GML object. An element in a GML document or data stream is a GML property if and only if it is a child element of a GML object element. The meaning of each property is indicated by the name of the element that instantiates it.

GML Objects may have an unlimited number of properties in addition to those inherited from gml:AbstractGMLType. A property may be defined to have either simple or complex content. A property with simple content has an XML Schema simpleContent type, as in the case of the standard property elements gml:description and gml:name. A property with complex content has an XML Schema complexContent type, as in the case of the standard propery element gml:member.

Property elements may use two modes:

- by value: the value of the property is represented directly, as the content of the property element. This method is used by the standard property gml:name and may be used for gml:description (see clause 7.2.2.5).
- by reference: the value of the property is available elsewhere, and is identified by the value of an **xlink:href** attribute on the property element. This alternative method may be used for the standard property gml:description (see clause 7.2.2.5).

7.2.2.3.2 Standard pattern for property declarations

To support the encoding of properties that may have complex content, a basic *pattern* for property elements is provided in the schema as follows:

```
<element name="_association" type="gml:AssociationType" abstract="true"/>


<complexType name="AssociationType">
  <sequence>
    <element ref="gml:_Object" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

NOTE: The declaration of gml:_association and its accompanying type definition is provided for convenience, to act as a template or pattern for the construction of property elements in application schemas. There is no requirement for specific properties to use XML Schema type derivation from gml:AssocationType to create properties in a conformant GML application schema. This contrasts with the requirement that the content model for all identifiable objects must derive from gml:AbstractGMLType, and for all features from gml:AbstracatFeatureType

7.2.2.3.3 Use xlinks to attach property values by-reference

XLink components are the standard method to support hypertext referencing in XML. An XML Schema attribute group, gml:AssociationAttributeGroup, is provided to support the use of Xlinks as the method for indicating the value of a property by reference in a uniform manner. This is defined in the schema as follows:

In xlinks.xsd (see clause 7.3.1):

```
<attributeGroup name="simpleLink">
  <attribute name="type" type="string" fixed="simple" form="qualified"/>
  <attribute ref="xlink:href" use="optional"/>
  <attribute ref="xlink:role" use="optional"/>
  <attribute ref="xlink:arcrole" use="optional"/>
  <attribute ref="xlink:title" use="optional"/>
  <attribute ref="xlink:show" use="optional"/>
  <attribute ref="xlink:actuate" use="optional"/>
</attributeGroup>
```

in gmlBase.xsd:

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

The value of a GML property that carries an xlink:href attribute is the resource returned by traversing the link.

In addition to the simpleLink components, the additional attribute **remoteSchema**, is provided to indicate a schema which constrains the description of the remote resource referenced by the xlink. Note that all components in the attribute group are optional.

7.2.2.3.4 By value or by reference?

The gml:_Object element in the content model for properties is optional. In combination with the component cardinalities in AssociationAttributeGroup this means that an element of this type may have a content element (in the substitution group headed by gml:_Object) or xlink attributes. GML property elements which follow this pattern can be used to attach values either by value or by reference.

> NOTE: When used in the “by value” form, a GML property element corresponds with an implementation of a UML composition association. When used in the “by reference” form, a GML property element corresponds with an implementation of a UML aggregation association. The standard form, which supports either style, may be compared with the UML untyped association. The name of the property element corresponds to the rolename on the UML association, or to a UML attribute name. The name of the value element (gml:Point in the example here) corresponds to the class of the target of the association.

For example, a standard property of GML Features (described below in Clause 7.4) is "location". This may be used to indicate a location by-value as follows:

```
<gml:location>
  <gml:Point gml:id="point96" srsName="epsg:4326">
    <gml:coordinates>31:56:00S 115:50:00E</gml:coordinates>
  </gml:Point>
</gml:location>
```

which uses the gml:Point object as defined in the GML geometry schemas (described below in Clause 7.5). The same property element may be used to indicate a location by

```
<gml:location xlink:href="http://my.big.org/locations/point53"/>
```

reference as follows:

where http://my.big.org/location/point53 identifies a point supplied by the service indicated.

However, a property element following this pattern may have no content or attributes, or it may have both content and attributes, and still be XML Schema-valid. It is not possible to constrain the co-occurrence of content or attributes, so we cannot use W3C XML Schema to restrict a property to be exclusively by-value or by-reference.

The constraint may be described precisely using an auxiliary constraint language Schematron [schematron]. The abstract, global elements **_association** and **_strictAssociation** both use **AssociationType**, but the following schema fragments shows how an element declaration may include a Schematron constraint to limit the property to act in either by-value or by-reference mode, but not both.

```
    <appinfo>
      <sch:title>Schematron validation</sch:title>
      <sch:ns prefix="gml" uri="http://www.opengis.net/gml"/>
      <sch:ns prefix="xlink" uri="http://www.w3.org/1999/xlink"/>
      <sch:pattern name="Check either href or content not both">
        <sch:rule abstract="true" id="hrefOrContent">
          <sch:report test="@xlink:href and (*|text())">
            Property element may not carry both a reference to an object and contain an object.</sch:report>
          <sch:assert test="@xlink:href | (*|text())">
            Property element must either carry a reference to an object or contain an object.</sch:assert>
        </sch:rule>
      </sch:pattern>
    </appinfo>
```

```
  <element name="_strictAssociation" type="gml:AssociationType" abstract="true"
substitutionGroup="gml:_property">
    <annotation>
      <appinfo>
        <sch:pattern name="refAndContent co-occurence prohibited">
          <sch:rule context="gml:_strictAssociation">
```



```
        <sch:extends rule="hrefOrContent"/>
      </sch:rule>
    </sch:pattern>
  </appinfo>
  <documentation>must carry a reference to an object or contain an object but not both</documentation>
 </annotation>
</element>
```

NOTE: Some XML validators will process the Schematron constraints automatically. Otherwise, the Schematron code can be treated merely as a formal description of the required constraint. It is included here primarily as an illustration of how this might be used for specific purposes by application schema developers.

To support the encoding of properties whose value is provided remotely by-reference (i.e. implementing the UML aggregation association), the following components are provided:

```
<element name="_reference" type="gml:ReferenceType" abstract="true"/>

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

The element gml:_reference is abstract, and thus may be used as the head of a subtitution group of more specific elements providing a value by reference.

7.2.2.4 Text values – gml:stringOrRefType

gml:StringOrRefType is provided to contain extended text values. It is defined in the schema document as follows:

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

It is an implementation of the GML property model in a simpleContent type.

NOTE: this is possible because an empty string is schema-valid.

This type is available wherever there is a need for a "text" type property. It is of string type, so the text can be included inline, but the value can also be referenced remotely via xlinks from the AssociationAttributeGroup. If the remote reference is present, then the value obtained by traversing the link should be used, and the string content of the element can be used for an annotation.

7.2.2.5 Standard properties of GML Objects: gml:metaDataProperty, gml:description, gml:name, gml:id

XML Schema types for all concrete GML objects derive from gml:AbstractGMLType. This means that all GML objects inherit certain standard properties that are included in the content model of gml:AbstractGMLType. In detail these are declared as follows:

```
<element name="metaDataProperty" type="gml:MetaDataPropertyType"/>
```

This property contains or refers to a metadata package that contains metadata properties. More detail is provided below in clause 7.2.2.8.

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

The value of this property is a text description of the object. **description** uses the GML **StringOrRefType** defined in clause 7.2.2.4, so may contain a simple text string content, or carry a reference to an external description.

```
<element name="name" type="gml:CodeType"/>
```

This property provides an identifier for the object, commonly a descriptive name.

An object may have several names, typically assigned by different authorities. gml:name uses the GML **CodeType** content model. 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. In common usage there will be one name per authority, so a processing application may select the name from its preferred codeSpace.

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

This property supports provision of a database handle for the object. Its use is optional but recommended. 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 methods [URI]. This is done by concatenating the URI for the document, a fragment separator “#”, and the value of the attribute of XML type ID.

7.2.2.6 gml:member – a utility property

A concrete property element named “member” with complex content is declared in gmlBase:

```
<element name="member" type="gml:AssociationType" substitutionGroup="gml:_property"/>
```


This is primarily to support the construction of object collections. In an instance

```
<gml:member xlink:href="http://my.org/thingie#i456"
gml:remoteSchema="http://my.org/schemas/thingie.xsd"/>
```

document this element may be used in a data instance as follows:

```
<gml:member>
  <my:Thingie  gml:id="i456" … />
</gml:member>
```

where the value is given by reference, or

where the value is encoded inline. Since gml:_Object is an abstract element, in this real instance it is replaced by a concrete element in its substitution group. The following element definition in the http://my.org/schemas/thingie.xsd application schema where my:ThingyType is defined supports this replacement:

```
<element name="Thingie" type="my:ThingyType" substitutionGroup="gml:_Object"/>
```

7.2.2.7 Bags and arrays

For a property that can have multiple values all with the same relationship to the containing object, the cardinality of the association is extended as follows:

```
<complexType name="ArrayAssociationType">
  <sequence>
    <element ref="gml:_Object" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>

<element name="members" type="gml:ArrayAssociationType"/>
```

This property element may appear in a data instance as follows:

```
<gml:members>
  <my:Thingie  gml:id="i456" … />
  <my:Thingie  gml:id="i457" … />
  <my:Thingie  gml:id="i458" … />
</gml:members>
```

Two concrete object elements are provided for generic arrays and bags. The schema constructions are as follows:

```
<complexType name="ArrayType">
  <complexContent>
```

```
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:members" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="Array" type="gml:ArrayType" substitutionGroup="gml:_GML"/>
```

intended to be used for a collection whose member objects are of homogeneous type and where their order is significant, and

```
<complexType name="BagType">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:member" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:members" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="Bag" type="gml:BagType" substitutionGroup="gml:_GML"/>
```

for general collections with no implication about the type, order or uniqueness of the member objects.

7.2.2.8 Metadata: gml:metaDataProperty, gml:_MetaData, gml:GenericMetaData

The metaDataProperty follows the property pattern and is used to contain or refer to metadata for GML objects. It is defined in the schema as follows:

```
<complexType name="MetaDataPropertyType">
  <sequence>
    <element ref="gml:_MetaData" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
  <attribute name="about" type="anyURI" use="optional"/>
</complexType>

<element name="metaDataProperty" type="gml:MetaDataPropertyType"
substitutionGroup="gml:_property"/>
```

The optional “about” attribute carries a URI which points to an element or range of elements, or other resource to which this metadata refers.

The value of the metaDataProperty is an abstract element gml:_MetaData that acts as a placeholder for “any package of metadata properties”, defined as follows:

```
<complexType name="AbstractMetaDataType" abstract="true" mixed="true">
  <attribute ref="gml:id" use="optional"/>
</complexType>

<element name="_MetaData" type="gml:AbstractMetaDataType" abstract="true"
substitutionGroup="gml:_Object"/>
```

It is used as follows:

1. The user defines or re-uses an existing metadata schema. The structure of this metadata schema is more or less arbitrary but should satisfy the following pattern:

   - The metadata schema must declare a single root element. This element must be substitutable for gml:_MetaData – i.e. it is derived from AbstractMetaDataType
   - The content model of this root element must be an XML Schema ComplexType that derives by extension from gml:AbstractMetaDataType.
   - The content model of this root element must consist of an XML model group of elements each of which is a metadata property (e.g. age, accuracy, creation date etc.). of the object to which the gml:metaDataProperty is attached.
   - If the user wishes to more strongly restrict the value of the gml:metaDataProperty, the application schema designer must define a new property that is an XML Schema restriction of gml:metaDataProperty.

2. The metaDataProperty in the data instance points to or includes the values of the metadata properties defined by the schema in 1. above.

3. The metaData property is used in one of two ways:

   - The metadata schema must declare a single root element. This element must be substitutable for gml:_MetaData.
   - Attached to a GML object and without the “gml:about” attribute it provides metadata for the GML object to which it is attached.
   - Attached to a GML object collection and with the gml:about attribute it provides metadata for the GML objects within the object collection referenced by the gml:about attribute, the value of which is an XPointer expression.

For convenience, a generic concrete MetaData element is provided:

```
<complexType name="GenericMetaDataType" mixed="true">
  <complexContent mixed="true">
    <extension base="gml:AbstractMetaDataType">
      <sequence>
        <any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
```

```
   </complexContent>
  </complexType>

  <element name="GenericMetaData" type="gml:GenericMetaDataType"
substitutionGroup="gml:_MetaData"/>
```

This element is intended to act as a container for metadata encoded using XML but defined in external schemas, in cases where it is not possible to assign the concrete components to the GML _MetaData substitution group.

7.3 Supporting components,

7.3.1 Xlinks – Object Associations and Remote Properties

The normative xlink specification is available from W3C [xlink]. A schema document xlinks.xsd is provided as part of GML, pending the availability of a normative W3C XML Schema implementation.

Xlink components are used in GML to implement associations between objects by reference. GML property elements may carry xlink attributes, which support the encoding of an association relationship by reference, the name of the property element denoting the target role in the association. The most important xlink component is:

href	Identifier of the resource which is the target of the association, given as a URI

The appearance of an xlink:href on a GML property indicates that the value of the property is to be found by traversing the link, that is the value is pointed to by the value of the xlink:href attribute. Following the terminology of Xlink, GML properties with xlink:href attributes are sometimes referred to as remote properties.

The other xlink components are used to indicate additional semantics of the relationship. The most useful of these are

role	description of the nature of the target resource, given as a URI
arcrole	description of the role or purpose of the target resource in relation to the present resource, given as a URI
title	description of the association or the target resource, given as text.

For complete definitions of these and other xlink components, including their use in extended xlink association maps, refer to the xlink specification [xlink].

In the GML core schemas (those defined in Clause 7. of this specification), simple xlinks are used exclusively to denote object, feature or geometry associations and to denote remotely referenced property values.

7.3.2 BasicTypes schema

Some simpleContent types are used in a number of places within the GML schema. For convenience, these are gathered together in the schema document basicTypes.xsd. The basic types and elements are described in the schema listed in Annex C. The basicTypes schema is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:basicTypes:v3.0

7.3.2.1 gml:NullType, gml: Null element

NullType defines a content model that allows an absent value to be indicated with some explanation.

```
<simpleType name="NullEnumeration">
  <restriction base="string">
    <enumeration value="inapplicable"/>
    <enumeration value="missing"/>
    <enumeration value="template"/>
    <enumeration value="unknown"/>
    <enumeration value="withheld"/>
  </restriction>
</simpleType>
<simpleType name="NullType">
  <union memberTypes="gml:NullEnumeration anyURI"/>
</simpleType>
```

NullType is a union type consisting of the following enumerated values:

innapplicable - the object does not have a value

missing - the correct value is not readily available to the sender of this data. Furthermore, a correct value may not exist.

template - the value will be available later

unknown - the correct value is not known to, and not computable by, the sender of this data. However, a correct value probably exists.

withheld - the value is not divulged

and

a URI – which should refer to a resource which describes the reason for the value not being available.

A particular community may choose to assign more detailed semantics to the standard values provided. Alternatively, the URI method enables a specific or more complete explanation for the absence of a value to be provided elsewhere.

A utility Null element is declared using this type.

```
<element name="Null" type="gml:NullType"/>
```

This element might appear in data instance documents as follows:

```
<gml:Null>withheld</gml:Null>

<gml:Null>http://my.big.org/explanations/theDogAteIt</gml:Null>
```

The first example uses one of the built-in values for Null. The second example contains a reference to an explanation available elsewhere, identified by a URI.

7.3.2.2 Simple Content Types which may carryNull Values

A set of simple types provide extensions to the XML Schema Boolean, string, double and integer by allowing either the XML Schema built-in simple type or the gml:NullType. They are constructed as follows:

```
<simpleType name="NameOrNull">
  <union memberTypes="gml:NullEnumeration Name anyURI"/>
</simpleType>
```

The types following this pattern are:

booleanOrNull stringOrNull NameOrNull doubleOrNull integerOrNull	Union of the respective XML Schema built-in simple type and the GML **Nulltype**. An element which uses one of these types may have content which is either boolean/string/Name/double/integer or a value from the Nulltype

7.3.2.3 Simple List Types based on XML Schema Built-in Types

A set of simple types for lists of simple values are constructed according to the following patterns:

```
<simpleType name="booleanList">
  <list itemType="boolean"/>
</simpleType>


<simpleType name="doubleOrNullList">
  <list itemType="gml:doubleOrNull"/>
</simpleType>
```

The types following this pattern are:

booleanList doubleList NameList integerList booleanOrNullList NameOrNullList doubleOrNullList	List of values of the respective XML Schema built-in simple types, or of the Union types listed above. The *OrNullList types support null values interspersed within a list.

integerOrNull List	

An element which uses one of these types will contain a whitespace-separated list of members of the relevant type (see http://www.w3.org/TR/xmlschema-2/#atomic-vs-list for more details of the XML list structure).

> NOTE: None of list types defined here use an XML Schema string as an item. The reason for this is that a string may include embedded spaces, linefeeds, etc (http://www.w3.org/TR/xmlschema-2/#string). Since whitespace acts as the item separator in a list instance, there would be ambiguity in identifying items that potentially contain whitespace. On the other hand, an instance of the XML Schema Name type may not contain whitespace (http://www.w3.org/TR/2000/WD-xml-2e-20000814#NT-Name), so this may be used safely in a list context. The corollary of this is that if a term may contain whitespace, then such a term may not occur in a list instance.

7.3.2.4 gml:SignType

This is a utility type with values "+" (plus) and "-" (minus).

```
<simpleType name="SignType">
  <restriction base="string">
    <enumeration value="-"/>
    <enumeration value="+"/>
  </restriction>
</simpleType>
```

Elements or attributes of this type are used in various places, e.g. to indicate the direction of topological objects with "+" for forwards, or "-" for backwards.

7.3.2.5 gml:CodeType

This is a generalized type to be used for a term, keyword or name. It adds a XML attribute codeSpace to a term, where the value of the (optional) codeSpace should indicate a dictionary, thesaurus, classification scheme, authority, or pattern for the term:

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

An element of this type may appear as in the following example:

```
<gml:name codeSpace = "http://www.ukusa.gov/placenames">St Paul</gml:name>
```


St Paul is asserted to be a meaningful name according to www.ukusa.gov/placenames. Note that in all cases the rules for the values, including such things as uniqueness constraints, are set by the authority responsible for the codeSpace.

7.3.2.6 gml:CodeListType and gml:CodeOrNullListType

These two types provide for lists of terms. The schema definitions are as follows:

```
<complexType name="CodeListType">
  <simpleContent>
    <extension base="gml:NameList">
      <attribute name="codeSpace" type="anyURI" use="optional"/>
    </extension>
  </simpleContent>
</complexType>

<complexType name="CodeOrNullListType">
  <simpleContent>
    <extension base="gml:NameOrNullList">
      <attribute name="codeSpace" type="anyURI" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

The values in an instance element of CodeListType should all be valid according to the rules of the dictionary, classification scheme, or authority identified by the value of its codeSpace attribute (e.g. list of localities, soil types, rock types, animal species etc.), for example:

```
<species codeSpace="http://my.big.org/florelegium">dryandra banksia hardenbergia lavendar</species>
```

where the listed items are from http://my.big.org/florelegium which is a (hypothetical) list of flowers.

An instance element of CodeOrNullListType may also include embedded values from NullType. It is intended to be used in situations where a term or classification is expected, but the value may be absent for some reason.

7.3.2.7 gml:Measure Type

A Measure Type is an amount encoded as a double, together with a units of measure indicated by a uom attribute. The value of uom (Units Of Measure) attribute is a reference to a Reference System for the amount, either a ratio or interval scale.

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

An element of this type might appear in a data instance document as follows :

```
<height uom="http://www.equestrian.org/units/hands">14</height>
<height uom="http://www.iso.org/iso/en/…/units/m">1.4224</height>
```

where the resource identified by the value of the uom attribute defines the unit of measure. GML components for this are defined in clause 7.10.2.

7.3.2.8 gml:MeasureListType, gml:MeasureOrNullListType

These two types provide for lists of quantities. The schema definitions are as follows:

```
<complexType name="MeasureListType">
  <simpleContent>
    <extension base="gml:doubleList">
      <attribute name="uom" type="anyURI" use="required"/>
    </extension>
  </simpleContent>
</complexType>

<complexType name="MeasureOrNullListType">
  <simpleContent>
    <extension base="gml:doubleOrNullList">
      <attribute name="uom" type="anyURI" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

The values in an instance element of MeasureListType are on a uniform scale, for example:

```
<heights uom=" http://www.iso.org/iso/en/…/units/m">1.76 1.85 1.56 1.98</heights>
```

An instance element of MeasureOrNullListType may also include embedded values from NullType. It is intended to be used in situations where a quantity is expected, but the value may be absent for some reason, for example:

```
<weights uom=" http://www.iso.org/iso/en/…/units/kg">67.0 73.4 withheld 85.1</weights>
```

7.3.2.9 gml:Coordinates Type

CoordinatesType is a text string, intended to be used to record an array of tuples or coordinates. While it is not possible to enforce the internal structure of the string through schema validation, some optional attributes are provided to support a description of the internal structure:


```
<complexType name="CoordinatesType">
  <simpleContent>
    <extension base="string">
      <attribute name="decimal" type="string" default="."/>
      <attribute name="cs" type="string" default=","/>
      <attribute name="ts" type="string" default="&#x20;"/>
    </extension>
  </simpleContent>
</complexType>
```

which are intended to be used as follows:

decimal	symbol used for a decimal point (default=”.” a stop or period)
cs	symbol used to separate components within a tuple or coordinate string (default=”,” a comma)
ts	symbol used to separate tuples or coordinate strings (default=” “ a space)

In a data instance document, an element of this type might appear as follows:

```
<gml:coordinates
decimal=“.“
cs=“,"
ts=“ ">1.413,4.524 1.429,5.516 1.432,7.235</gml:coordinates>
```

Since it is based on the XML Schema string type, CoordinatesType can be used in the construction of tables of tuples or arrays of tuples, including ones that contain mixed text and numeric values. For example:

```
<my:tupleList>bettong,357.,2.3  skink,140.,0.75  wombat,770.,17.5</my:tupleList>
```

7.3.2.10 Examples of some properties using the simpleContent types defined here

This Clause illustrates how the GML types described above can be used in for GML properties. We illustrate these first by a set of simple examples:

Example 1. MeasureType

The application schema (Feature Application Schema) contains the declaration:

```
<element name = “height”  type = “gml:MeasureType”/>
```

and the corresponding instance element is:

```
<abc:height uom = “#m”>50</abc:height>
```

Example 2. CountType

The application schema (Feature Application Schema) contains the declaration:

<element name = “noSpecies” type = “gml:CountType”/>

and the corresponding instance element is:

<abc:noSpecies>50000</abc: noSpecies>

Example 3: CodeType

The application schema (Feature Application Schema) contains the declaration

<element name = “soilType” type = “gml:CodeType”/>

and the corresponding instance element is:

<abc:soilType codeSpace = “http://www.ukusa.gov/soils.xml”>podzolic</abc: soilType>

In this example, the value of the codeSpace attribute, if present, is a reference to a Reference System for the value, a dictionary, nominal classification schema or code list.

7.4 Feature Model

A GML feature is a meaningful object in the selected domain of discourse such as a Road, River, Person, Vehicle or Administrative Boundary. This follows the general definition of a feature given in ISO 19109 and the OGC Abstract Specification Topic 5.

7.4.1 Feature schema

The feature schema, feature.xsd, provides a framework for the creation of GML features and feature collections. These are described in the schema listed in Annex C.

The feature schema includes a basic geometry schema (geometryBasic2d.xsd). The gmlBase schema is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:feature:v3.0

7.4.1.1 gml:AbstractFeatureType, gml:_Feature

The basic feature model is given by the gml:AbstractFeatureType, defined in the schema as follows:

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

```
    <attribute name="fid" type="string">
     <annotation>
      <appinfo>deprecated</appinfo>
      <documentation>deprecated in GML version 3.0</documentation>
     </annotation>
    </attribute>
   </extension>
  </complexContent>
 </complexType>
```

The content model for gml:AbstractFeatureType adds two specific properties suitable for geographic features to the content model defined in gml:AbstractGMLType. The value of the **gml:boundedBy** property describes an envelope that encloses the entire feature instance, and is primarily useful for supporting rapid searching for features that occur in a particular location. The value of the **gml:location** property describes the extent, position or relative location of the feature.

7.4.1.2 gml:_Feature

The element gml:_Feature is declared as follows:

```
 <element name="_Feature" type="gml:AbstractFeatureType" abstract="true"
substitutionGroup="gml:_GML"/>
```

This abstract element serves as the head of a substitution group which may contain any elements whose content model is derived from gml:AbstractFeatureType. This may be used as a variable in the contruction of content models.

gml:_Feature can be thought of as “anything that is a GML feature” and can be used to define variables or templates in which the value of a GML property is “any feature”. This occurs in particular in a GML Feature Collection (see 7.4.8) where the <gml:featureMember> and <gml:featureMembers> properties contain one or multiple copies of gml:_Feature respectively.

However, an application schema writer may prevent an element with a content model derived directly or indirectly from gml:AbstractFeatureType from being included in any general gml:FeatureCollection by excluding it from direct or indirect membership in the gml:_Feature substitutionGroup. For example:

```
<element name=”WalledGarden” type=”abc:WalledGardenType”/>
```

GML features also have a (optional) fid attribute. This is for backwards compatibility with GML 2.x and is deprecated in GML 3.

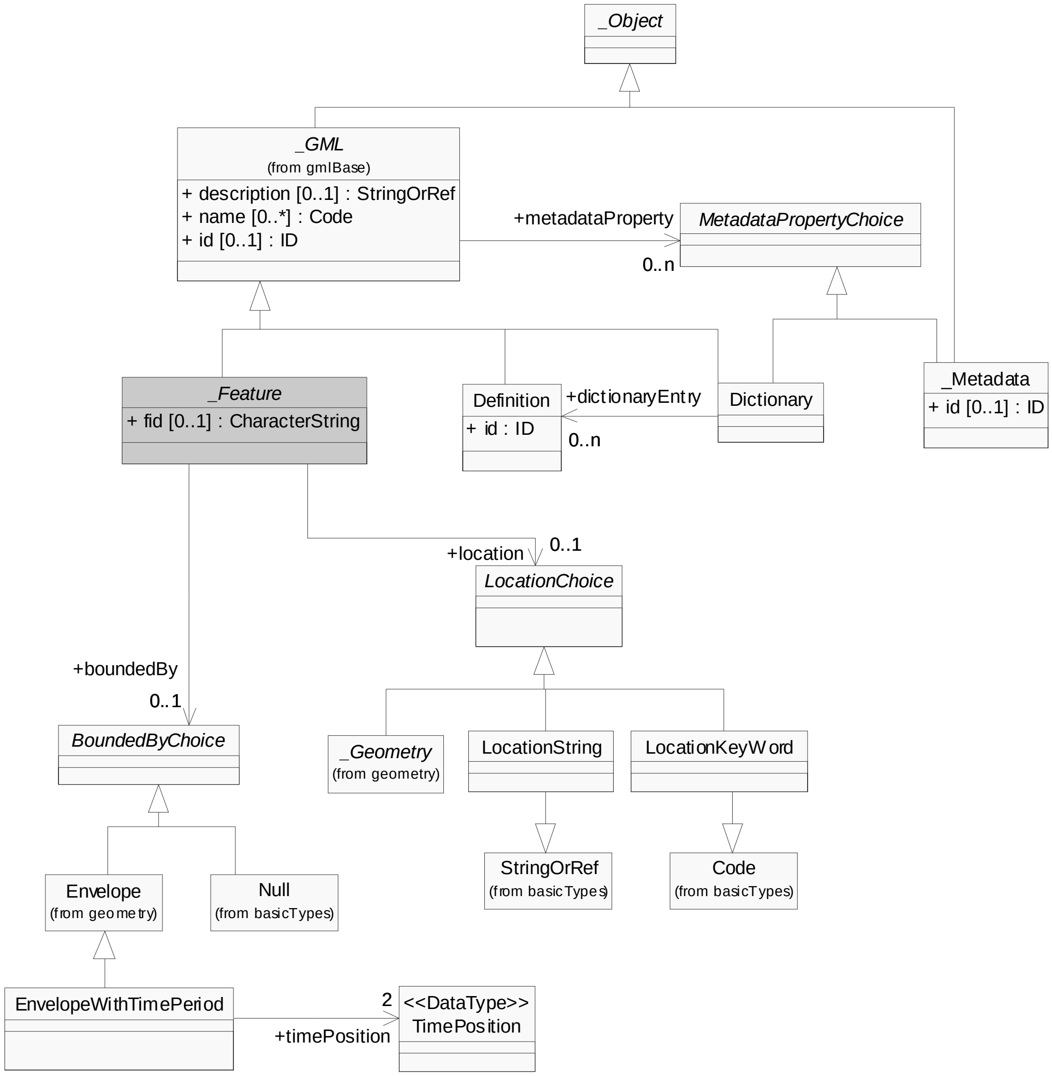


Figure 7.4-1 UML representation of gml:_feature

The feature model described in the feature.xsd schema may be represented by the UML class diagram in Figure 7.4-1.

All specific feature types defined in application schemas must be derived from gml:AbstractFeatureType, and thus all GML features have optional gml:boundedBy and gml:location properties, as well as the standard gml:metaDataProperty, gml:description and gml:name properties inherited from gml:AbstractGMLType. AbstractFeatureType also inherits gml:id from gml:AbstractGMLType and this is the preferred means of supporting database identifiers in GML 3.0.

General purpose software designed to process arbitrary GML data must be able to traverse the XML Schema inheritance tree in order to determine whether or not a given element in the data stream is a GML feature.

A GML feature has a set of properties, where the specific set of properties defines the feature type. A specific feature type is defined by its list of properties [ISO 19110]. In GML properties may be simple, using XML Schema simpleContent types, or properties may be declared using the gml:AssociationType pattern described in Clause 7.3.2.

In the application schema defining a feature there must be a global element declared whose name is the semantic type of the feature in the domain of discourse.

```
<element name="<<featureName>>"  type = "<<contentModel >>" />
```

If that feature element may be used in a general gml:FeatureCollection, or in a feature collection defined by a different application schema, then its global element declaration must be made a member of the gml:_Feature substitution group:

```
<element name="<<featureName>>"  type = "<<contentModel >>" substitutionGroup="gml:_Feature" />
```

The content model of the feature may be a named (or anonymous) complex type.

7.4.1.3 gml:boundedBy

The value of this property defines the bounds of an envelope that encloses the entire feature. Its content model is as follows:

```
<element name="boundedBy" type="gml:BoundingShapeType"/>
<complexType name="BoundingShapeType">
  <sequence>
    <group ref="gml:boundingShape"/>
  </sequence>
</complexType>
```

The gml:boundingShape is defined by:

```
<group name="boundingShape">
  <choice>
    <element ref="gml:Envelope"/>
    <element ref="gml:Null"/>
  </choice>
</group>
```

For envelopes that include the temporal extent of a feature (collection), an extension of gml:Envelope (see 7.5.1.5) is defined as:

```
<element name="EnvelopeWithTimePeriod" type="gml:EnvelopeWithTimePeriodType"
  substitutionGroup="gml:Envelope"/>

<complexType name="EnvelopeWithTimePeriodType">
  <complexContent>
    <extension base="gml:EnvelopeType">
      <sequence>
        <element ref="gml:timePosition" minOccurs="2" maxOccurs="2"/>
      </sequence>
      <attribute name="frame" type="anyURI" use="optional" default="#ISO-8601"/>
    </extension>
  </complexContent>
</complexType>
```

This adds two gml:timePosition properties which describe the extent of a time-envelope. In this way a boundedBy property may indicate the time-extent of a feature in a way that is symmetrical with the way that the spatial extent is described.

Note that since gml:EnvelopeWithTimePeriod is assigned to the substitution group headed by gml:Envelope, it may be used whenever gml:Envelope is valid.

7.4.1.4 gml:location

The location property describes the generalized location of the feature. The value of this property can be any of the following types as specified in the gml:locator.

```
<element name="location" type="gml:LocationPropertyType" substitutionGroup="gml:_property"/>

  <complexType name="LocationPropertyType">
  <sequence >
    <group ref="gml:locator"  minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

where the gml:locator is defined as:

```
<group name="locator">
  <choice>
    <element ref="gml:_Geometry"/>
    <element ref="gml:LocationString"/>
    <element ref="gml:LocationKeyWord"/>
  </choice>
</group>
```

Note that the value of the location property is a complex type but is not an arbitrary GML object.The location of a GML feature can be a geometry, a location string or a location keyword.

A location string is a text string which should describe the location. It is declared as follows:

```
<element name="LocationString" type="gml:StringOrRefType"/>
```

The location keyword is described by:

```
<element name="LocationKeyWord" type="gml:CodeType"/>
```

The following examples illustrate the different ways that the location property may appear in a data instance.

Location given as a gml Geometry in a particular spatial reference system:

```
<gml:location>
  <gml:Point gml:id="point96" srsName="epsg:4326">
    <gml:coordinates>31:56:00S 115:50:00E</gml:coordinates>
  </gml:Point>
</gml:location>
```

Location given using a name from a controlled source:

```
<gml:location>
  <gml:LocationKeyWord
codeSpace="http://www.anzlic.org.au/icsm/cgna/index.html">Leederville</gml:LocationKeyWord>
</gml:location>
```

Location given using a text string:

```
<gml:location>
  <gml:LocationString>Nigel Foster's town of residence</gml: LocationString>
</gml:location>
```

Location given by another service:

```
<gml:location xlink:href="http://my.big.org/locations/point53"/>
```

7.4.1.5 gml:priorityLocation

A property gml:priorityLocation is provided for GML Application Schema developers that wish to provide prioritized locations for their features. A gml:priorityLocation as the following content model:

```
<element name="priorityLocation" type="gml:PriorityLocationPropertyType"
        substitutionGroup="gml:location"/>
```

```
<complexType name="PriorityLocationPropertyType">
  <complexContent>
    <extension base="gml:LocationPropertyType">
      <attribute name="priority" type="string" use="optional"/>
    </extension>
  </complexContent>
</complexType>
```

Note that this simply adds a priority string to the base gml:location property. This can be used to assign levels of importance to the different locations (e.g. location by location string, geometry etc.).

7.4.1.6 gml:BoundedFeatureType

A simple restriction of gml:AbstractFeatureType makes the optional boundedBy property mandatory. gml:BoundedFeatureType is defined as follows:

```
<complexType name="BoundedFeatureType" abstract="true">
  <annotation>
    <documentation> Makes boundedBy mandatory - used to build Feature Collections </documentation>
  </annotation>
  <complexContent>
    <restriction base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:description" minOccurs="0"/>
        <element ref="gml:name" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:boundedBy"/>
        <element ref="gml:location" minOccurs="0"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

NOTE: In XML Schema a content model derived by restriction must lead to a content model which is valid according to the base type – i.e. it is a valid subset of the parent. In the schema document this is expressed by copying down all the XML elements that are required for the content of the derived type, with their occurrence counts. In this instance the adjustment made in the derived type is that an element that was optional in the parent (gml:boundedBy had minOccurs=”0”) is made mandatory (no occurrence constraints implies exactly one). Any element from the parent that is omitted (which can only be those with minOccurs=”0” in the parent definition) is not inherited by the derived type. In contrast, it is *not* necessary to copy down the XML attributes, only those which are altered by restriction need be mentioned. By default the derived type has the same XML attributes and occurrence constraints as the parent. See XML Schema Part 1 (Structures) for more detail.

7.4.1.7 Feature Association Properties: gml:featureMember, gml:featureProperty, gml:featureMembers

A particular class of properties define associations between features. These use the gml:AssociationType pattern as follows:

```
<complexType name="FeaturePropertyType">
  <sequence>
    <element ref="gml:_Feature" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

The concrete elements gml:featureMember and gml:featureProperty use this content model, and are declared as follows:

```
<element name="featureMember" type="gml:FeaturePropertyType"/>
```

```
<element name="featureProperty" type="gml:FeaturePropertyType"/>
```

At times it is useful to define a property containing an array of other features. This is done using a feature array property type as defined by the following content model:

```
 <complexType name="FeatureArrayPropertyType">
  <sequence>
    <element ref="gml:_Feature" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

The concrete elements gml:featureMembers uses this content model, and is declared as follows:

```
<element name="featureMembers" type="gml:FeatureArrayPropertyType"/>
```

7.4.1.8 Feature Collection

A GML Feature Collection is a collection of GML feature instances that can behave as a GML feature. All GML feature collections must be derived by extension or restriction from gml:AbstractFeatureCollectionType, which is defined as follows:

```
<complexType name="AbstractFeatureCollectionType" abstract="true">
  <complexContent>
    <extension base="gml:BoundedFeatureType">
      <sequence>
```



```
      <element ref="gml:featureMember" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="gml:featureMembers" minOccurs="0"/>
    </sequence>
   </extension>
  </complexContent>
 </complexType>
```

The gml:featureMember property (but not the gml:featureMembers property) follows the association pattern and can thus refer to a “remote” feature by means of the xlink:ref attribute.

The compositing property gml:featureMembers encloses a set of members of the Feature Collection regardless of their semantic type as features. gml:featureMember encloses or references a single feature instance. gml:featureMember and gml:featureMembers properties can appear on the same Feature Collection, but there can be only one gml:featureMembers property.

GML Feature Collections are themselves valid GML features and can have gml:location and other properties as defined in their GML Application Schema (see chapter 8.).

gml:AbstractFeatureCollectionType is based on gml:BoundedFeatureType. This means that the gml:boundedBy property (see Clause 7.2.1.1) is mandatory on all GML FeatureCollections. This is to ensure that spatial searching is supported without imposing the need to inspect every member of a collection. This is true even if the Feature Collection is itself contained within (i.e. is the value of a property of) another feature.

The schema also provides a concrete feature collection:

```
<element name="FeatureCollection" type="gml:FeatureCollectionType" substitutionGroup="gml:_Feature"/>
 <complexType name="FeatureCollectionType">
  <complexContent>
   <extension base="gml:AbstractFeatureCollectionType"/>
  </complexContent>
 </complexType>
```

Users of the concrete gml:FeatureCollection should note that it allows any valid GML feature as a member.

The content model of a GML Feature Collection must be derived from gml:AbstractFeatureCollectionType. This in turn derives from gml:AbstractFeatureType. Hence feature collections are features, and are in general substitutable for gml:_Feature.

7.4.1.9 Named Geometry Properties

The feature.xsd schema provides a set of aliases for the GML geometry properties. These are:

```
<element name="centerOf" type="gml:PointPropertyType" substitutionGroup="gml:pointProperty"/>
<element name="position" type="gml:PointPropertyType" substitutionGroup="gml:pointProperty"/>
<element name="extentOf" type="gml:SurfacePropertyType" substitutionGroup="gml:surfaceProperty"/>
```

```
<element name="edgeOf" type="gml:CurvePropertyType" substitutionGroup="gml:curveProperty"/>
<element name="centerLineOf" type="gml:CurvePropertyType" substitutionGroup="gml:curveProperty"/>
```

These property elements provide common role names for the geometry of geographic features. The specific semantics of these role names (e.g. centerOf) is not defined in GML, and it is expected that additional clarification would be provided in <annotations> within the using GML application schema.

7.5 Geometry

7.5.1 General Concepts

7.5.1.1 Overview

The geometry model of GML follows Topic 1 of the Abstract Specification of OGC which is identical to ISO DIS 19107.

> NOTE: The underlying concepts of the types and elements of the GML geometry model are discussed in this document in more detail.

This Clause describes the geometry model used by GML in the XML Schema documents:

- geometryBasic0d1d.xsd
- geometryBasic2d.xsd
- geometryPrimitives.xsd
- geometryAggregates.xsd
- geometryComplexes.xsd

The basic GML types and elements are described in the schemas listed in Annex C. The geometry schemas are identified by the following location-independent name (using URN syntax):

```
urn:opengis:specification:gml:schema-xsd:geometryBasic0d1d:v3.0
urn:opengis:specification:gml:schema-xsd:geometryBasic2d:v3.0
urn:opengis:specification:gml:schema-xsd:geometryPrimitives:v3.0
urn:opengis:specification:gml:schema-xsd:geometryAggregates:v3.0
urn:opengis:specification:gml:schema-xsd:geometryComplexes:v3.0
```

All types, elements and attributes in these XML Schema documents are covered by this Clause.

> NOTE: The GML geometry model is complex. To make the schemas more accessible to readers with or without prior experience with GML 2, the different types and elements are grouped in several XML Schema documents (in GML 2 there was only one schema, geometry.xsd).

The types and elements required for backwards compatibility with GML 2 are part of geometryBasic0d1d.xsd, geometryBasic2d.xsd and geometryAggregates.xsd. geometryPrimitives.xsd and geometryComplex.xsd are comprised of entirely new types and elements.

Class diagrams of an UML Model are added for illustration. These diagrams are informative, not normative. Figure 7.5-1 shows an overview of the type hierarchy of the GML geometry types.

NOTE: The diagrams represent an implementation model derived from the conceptual UML model specified in Topic 1 of OGC's Abstract Specification (ISO DIS 19107).

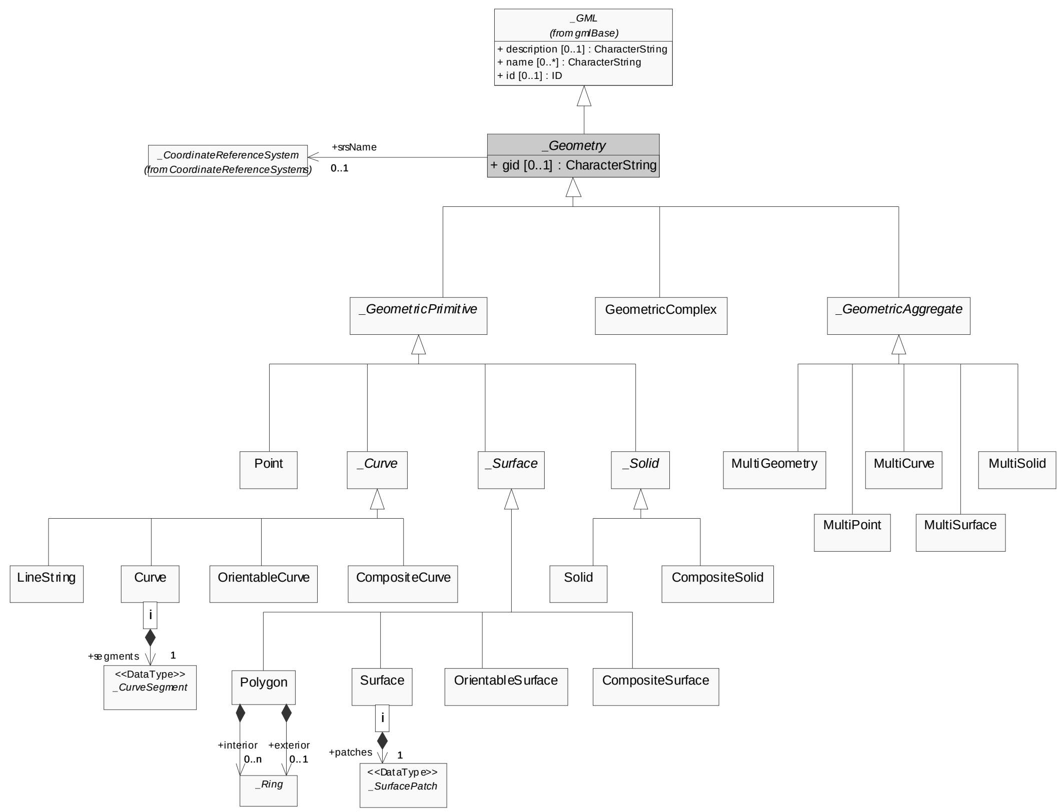


Figure 7.5-1 – Type Hierarchy of the GML geometry types

Any geometry element that inherits the semantics of AbstractGeometryType can be viewed as a set of direct positions.

All of the classes in the inheritance tree of AbstractGeometryType inherit an optional association to a coordinate reference system. All direct positions shall directly or indirectly be associated with a coordinate reference system. When geometry elements are aggregated in another geometry element (such as a MultiGeometry or

GeometricComplex), which already has a coordinate reference system specified, then these elements are assumed to be in that same coordinate reference system unless otherwise specified.

The geometry model distinguishes geometric primitives, aggregates and complexes.

Geometric primitives (i.e. instances of a subtype of AbstractGeometricPrimitiveType) will be open, that is, they will not contain their boundary points; curves will not contain their end points, surfaces will not contain their boundary curves, and solids will not contain their bounding surfaces.

Geometric aggregates (i.e. instances of a subtype of AbstractGeometricAggregateType) are arbitrary aggregations of geometry elements. They are not assumed to have any additional internal structure and are used to "collect" pieces of geometry of a specified type. Application schemas may use aggregates for features that use multiple geometric objects in their representations.

Geometric complexes (i.e. instances of GeometricComplexType) are closed collections of geometric primitives, i.e. they will contain their boundaries.

The composite geometries (CompositeCurve, CompositeSurface and CompositeSolid) can be viewed as primitives and as complexes. See Clause 7.6 and ISO DIS 19107 for more details on the nature of composite geometries.

This clause consists of the following parts:

▪ General concepts (this subclause) ▪ Coordinate Geometry ▪ Simple Geometric Primitives (0- and 1-dimensional)	geometryBasic0d1d.xsd
▪ Simple Geometric Primitives (2-dimensional)	geometryBasic2d.xsd
▪ More Geometric Primitives (1-, 2- and 3-dimensional)	geometryPrimitives.xsd
▪ Geometric Complex and geometric composites	geometryComplexes.xsd
▪ Geometric Aggregates	geometryAggregates.xsd
▪ Geometric Properties	n/a
▪ User-defined Geometry Types and Geometry Property Types	n/a

7.5.1.2 Abstract Geometry

Complex Type AbstractGeometryType

```
<complexType name="AbstractGeometryType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <attribute name="gid" type="string" use="optional" />
```



```
            <attribute name="srsName" type="anyURI" use="optional" />
        </extension>
    </complexContent>
</complexType>
```

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 to:

- 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.

The attribute “gid” is included for backward compatibility with GML 2 and is deprecated with GML 3. This identifier is superceded by "gml:id" inherited from AbstractGMLType. The attribute "gid" should not be used anymore and may be deleted in future versions of GML without further notice.

In general the attribute “srsName” 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 (Note: These "WKCRS"-ids still have to be specified).

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 aggregate.

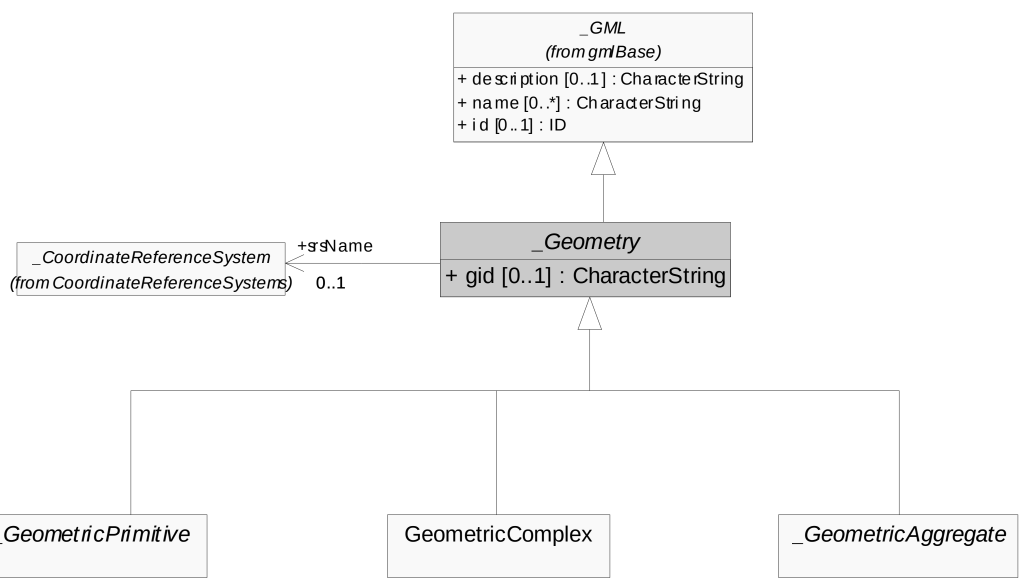


Figure 7.5-2 – AbstractGeometryType

Element Geometry

```
<element name="_Geometry" type="gml:AbstractGeometryType" abstract="true" substitutionGroup="gml:_GML" />
```

The "_Geometry" element is the abstract head of the substitution 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".

Complex Type GeometryPropertyType

```
<complexType name="GeometryPropertyType">
    <sequence>
        <element ref="gml:_Geometry"  minOccurs="0" />
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```

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 or another document). Note that either the reference or the contained element must be given, but not both or none. See clause 7.3.

If a feature has a property, which takes a geometry element as its value, this is called a geometry property. A generic type for such a geometry property is GeometryPropertyType which follows the general rules described in clauses 7.3 and 7.5.7.

Complex Type GeometryArrayPropertyType

```
<complexType name="GeometryArrayPropertyType">
    <sequence>
        <element ref="gml:_Geometry"  minOccurs="0" maxOccurs="unbounded" />
    </sequence>
</complexType>
```

A container for an array of geometry elements. The elements are always contained in the array property, referencing geometry elements or arrays of geometry elements via XLinks is not supported.

If a feature has a property which takes an array of geometry elements as its value, this is called a geometry array property. A generic type for such a geometry property is GeometryArrayPropertyType which follows the general rules described in clause 7.3.4.

In general the elements in an array have to be homogenous, containing elements that are all of the same type. For example, all elements in a GeometryArrayPropertyType are of the type AbstractGeometryType (including types derived from this abstract base type) as long as the element is directly or indirectly substitutable for _Geometry.

7.5.1.3 Coordinate Geometry

Complex Type DirectPositionType

```
<complexType name="DirectPositionType">
    <simpleContent>
        <extension base="gml:doubleList">
            <attribute name="srsName" type="anyURI" use="optional" />
            <attribute name="dimension" type="positiveInteger" use="optional" />
        </extension>
    </simpleContent>
</complexType>
```

DirectPosition instances hold the coordinates for a position within some coordinate reference system (CRS). Since DirectPositions, as data types, will often be included in larger objects (such as geometry elements) that have references to CRS, the "srsName" attribute will in general be missing, if this particular DirectPosition is included in a larger element with such a reference to a CRS. In this case, the CRS is implicitly assumed to take on the value of the containing object's CRS.

In general the attribute “srsName” 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 (Note: These "WKCRS"-ids still have to be specified).

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 the attribute will be specified at the direct position level only in rare cases.

The attribute "dimension" is the length of the coordinate sequence (the number of entries in the list). This is determined by the coordinate reference system.

Element pos

```
<element name="pos" type="gml:DirectPositionType" />
```

Element pointRep

```
<element name="pointRep" type="gml:PointPropertyType" />
```

Element coordinates

```
<element name="coordinates" type="gml:CoordinatesType" />
```

Complex Type CoordType

```
<complexType name="CoordType">
    <sequence>
        <element name="X" type="decimal" />
        <element name="Y" type="decimal" minOccurs="0" />
        <element name="Z" type="decimal" minOccurs="0" />
    </sequence>
</complexType>
```

Represents a coordinate tuple in one, two, or three dimensions. Deprecated with GML 3.0 and replaced by DirectPositionType.

Element coord

```
<element name="coord" type="gml:CoordType" />
```

Deprecated with GML 3.0 and included for backwards compatibility with GML 2. Use the "pos" element instead.

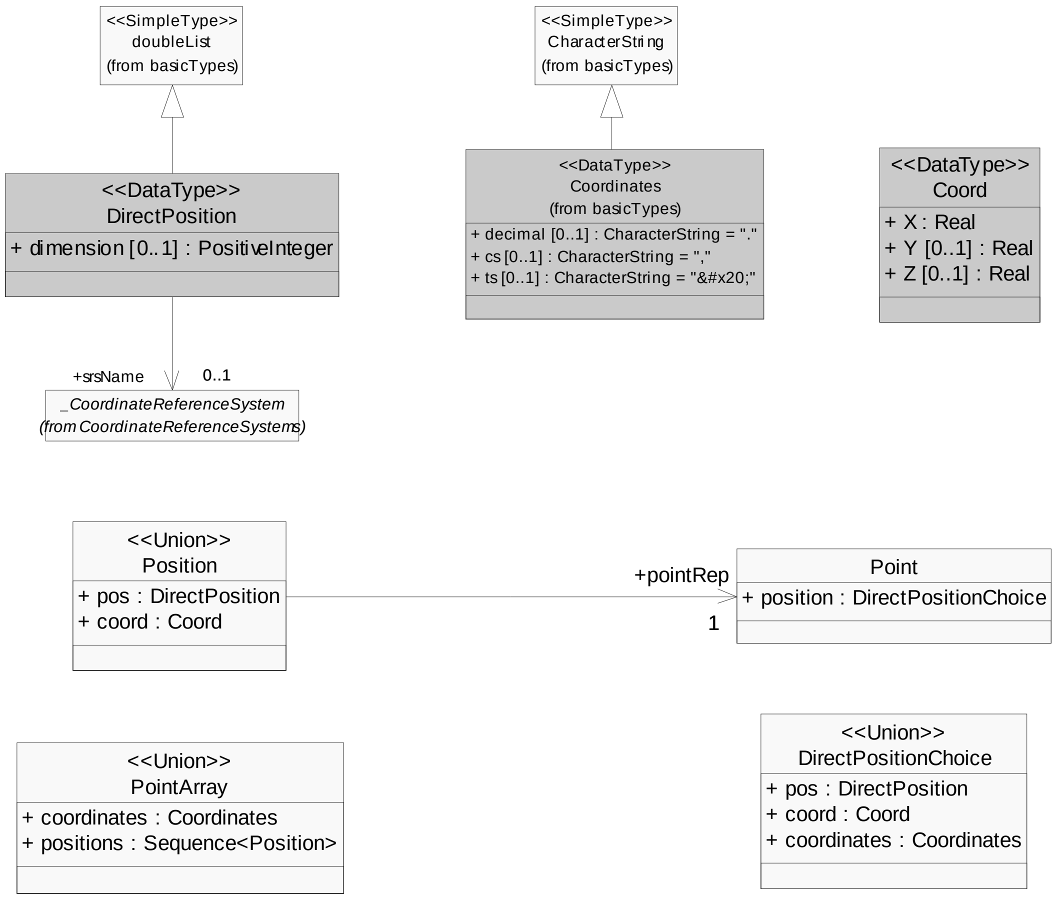


Figure 7.5-3 – Coordinate Geometry

7.5.1.4 Vector

Complex Type VectorType

```
<complexType name="VectorType">
  <simpleContent>
    <restriction base="gml:DirectPositionType"/>
  </simpleContent>
</complexType>
```

Element vector

```
<element name="vector" type="gml:VectorType" />
```

A Vector is an ordered set of numbers called coordinates that represent a position in a coordinate reference system (CRS). For some application the components of the position may be adjusted to yield a unit vector.

Note: This definition allows VectorType to be used elsewhere when appropriate – e.g. for offsetVector in grids.xsd, and vector to be used directly when appropriate – e.g. in DirectionVector in direction.xsd.

7.5.1.5 Envelope

Complex Type EnvelopeType

```
<complexType name="EnvelopeType">
    <complexContent>
        <extension base="gml:AbstractGeometryType">
            <sequence>
                <choice>
                    <element ref="gml:coord" minOccurs="2" maxOccurs="2"/>
                    <element ref="gml:pos" minOccurs="2" maxOccurs="2"/>
                    <element ref="gml:coordinates"/>
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

Element envelope

```
<element name="Envelope" type="gml:EnvelopeType" substitutionGroup="gml:_Geometry"/>
```

Envelope defines an extent using a pair of positions defining opposite corners in arbitrary dimensions.

Note: Envelope is often referred to as a minimum bounding box or rectangle. Regardless of dimension, an envelope can be represented without ambiguity as two direct positions (coordinate points).

7.5.2 Simple Geometric Primitives (0- and 1-dimensional)

Complex Type AbstractGeometricPrimitiveType

```
<complexType name="AbstractGeometricPrimitiveType" abstract="true">
    <complexContent>
        <extension base="gml:AbstractGeometryType" />
    </complexContent>
</complexType>
```

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.

Element _GeometricPrimitive

```
<element name="_GeometricPrimitive" type="gml:AbstractGeometricPrimitiveType" abstract="true"
    substitutionGroup="gml:_Geometry" />
```

The "_GeometricPrimitive" element is the abstract head of the substitution group for all (pre- and user-defined) geometric primitives.

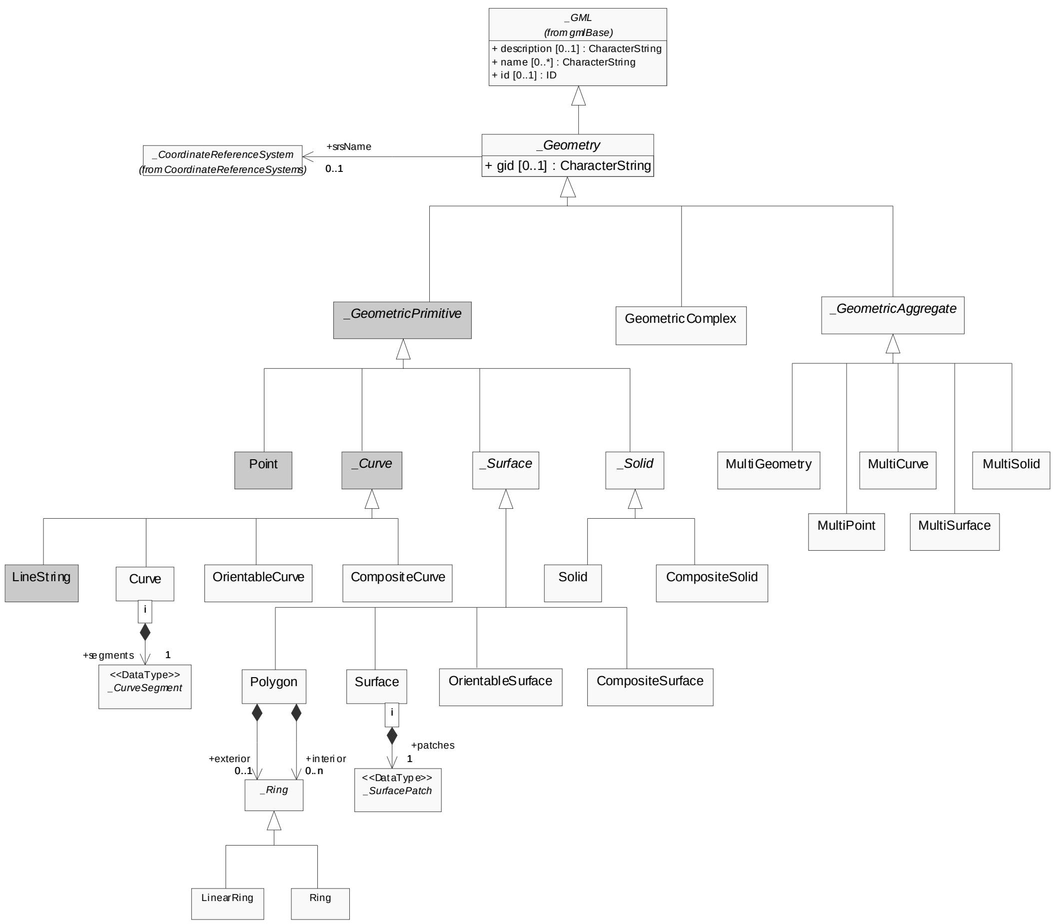


Figure 7.5-4 – Simple Geometric Primitives (0- and 1-dimensional)

Complex Type GeometricPrimitivePropertyType

```
<complexType name="GeometricPrimitivePropertyType">
    <sequence >
```



```
        <element ref="gml:_GeometricPrimitive"  minOccurs="0" />
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```

A property that has a geometric primitive 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.

Complex Type PointType

```
<complexType name="PointType">
    <complexContent>
        <extension base="gml:AbstractGeometricPrimitiveType">
            <sequence>
                <choice>
                    <element ref="gml:pos" />
                    <element ref="gml:coordinates" />
                    <element ref="gml:coord" />
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

A Point is defined by a single coordinate tuple.

GML supports two different ways to specify the direct position of a point.

- The "pos" element is of type DirectPositionType.
- The "coordinates" element is of type CoordinatesType. The number of direct positions in the coordinates list must be one.

The use of the element “coord” is deprecated with GML version 3.0. Use "pos" instead. The "coord" element is included for backwards compatibility with GML 2.

Element Point

```
<element name="Point" type="gml:PointType" substitutionGroup="gml:_GeometricPrimitive" />
```

Complex Type PointPropertyType

```
<complexType name="PointPropertyType">
```



```
    <sequence >
      <element ref="gml:Point" minOccurs="0" />
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
  </complexType>
```

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.

Element pointProperty

```
  <element name="pointProperty" type="gml:PointPropertyType" />
```

This property element either references a point via the XLink-attributes or contains the point element. pointProperty is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for Point.

Complex Type PointArrayPropertyType

```
  <complexType name="PointArrayPropertyType">
    <sequence>
      <element ref="gml:Point" minOccurs="0" maxOccurs="unbounded" />
    </sequence>
  </complexType>
```

A container for an array of points. The elements are always contained in the array property, referencing geometry elements or arrays of geometry elements via XLinks is not supported.

Element pointArrayProperty

```
  <element name="pointArrayProperty" type="gml:PointArrayPropertyType" />
```

This property element contains a list of point elements. pointArrayProperty is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for a list of Points.

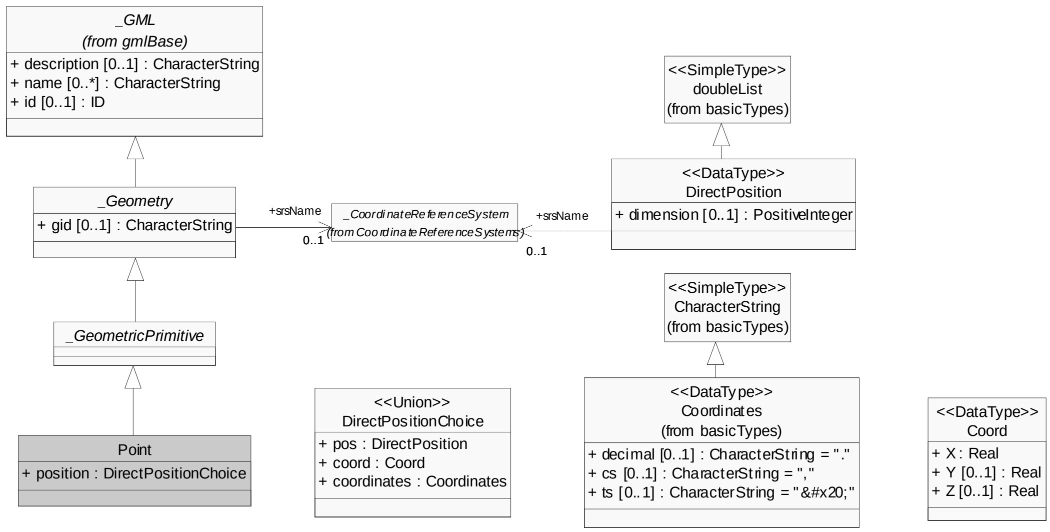


Figure 7.5-5 – Point

Complex Type AbstractCurveType

```
<complexType name="AbstractCurveType" abstract="true">
    <complexContent>
        <extension base="gml:AbstractGeometricPrimitiveType" />
    </complexContent>
</complexType>
```

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.

Element _Curve

```
<element name="_Curve" type="gml:AbstractCurveType" abstract="true"
    substitutionGroup="gml:_GeometricPrimitive" />
```

The "_Curve" element is the abstract head of the substitution group for all (continuous) curve elements.

Complex Type CurvePropertyType

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

```
</complexType>
```

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.

Element curveProperty

```
<element name="curveProperty" type="gml:CurvePropertyType" />
```

This property element either references a curve via the XLink-attributes or contains the curve element. curveProperty is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for _Curve.

Complex Type CurveArrayPropertyType

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

A container for an array of curves. The elements are always contained in the array property, referencing geometry elements or arrays of geometry elements via XLinks is not supported.

Element curveArrayProperty

```
<element name="curveArrayProperty" type="gml:CurveArrayPropertyType" />
```

This property element contains a list of curve elements. curveArrayProperty is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for a list of _Curves.

Complex Type LineStringType

```
<complexType name="LineStringType">
    <complexContent>
        <extension base="gml:AbstractCurveType">
            <sequence>
                <choice>
                    <choice minOccurs="2" maxOccurs="unbounded">
```

```
                        <element ref="gml:pos" />
                        <element ref="gml:pointRep" />
                        <element ref="gml:coord" />
                    </choice>
                    <element ref="gml:coordinates" />
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

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. GM_LineString of ISO 19107 is implemented by LineStringSegment.

GML supports two different ways to specify the control points of a line string:

- A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this curve, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this curve (reuse of existing points).
- The "coordinates" element allows for a compact way to specify the coordinates of the control points, if all control points are in the same coordinate reference systems and belong to this curve only. The number of direct positions in the coordinates list must be at least two and include the start and end point. Note that the start and end points of one line string are often identical with those of other line strings.

The use of the element “coord” is deprecated with GML version 3.0. Use "pos" instead. The "coord" element is included for backwards compatibility with GML 2.

Element LineString

```
<element name="LineString" type="gml:LineStringType" substitutionGroup="gml:_Curve" />
```

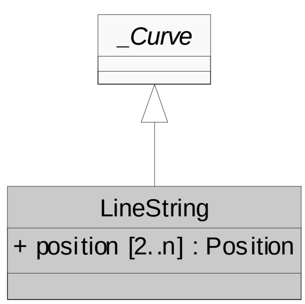


Figure 7.5-6 – LineString

Complex Type LineStringPropertyType

```
<complexType name="LineStringPropertyType">
  <sequence>
    <element ref="gml:LineString"  minOccurs="0" />
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```

This type is deprecated with GML 3 and shall not be used. It is included for backwards compatibility with GML 2. Use CurvePropertyType instead.

A property that has a line string 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.

Element lineStringProperty

```
<element name="lineStringProperty" type="gml:LineStringPropertyType" />
```

Deprecated with GML 3.0 and included only for backwards compatibility with GML 2.0. Use "curveProperty" instead.

This property element either references a line string via the XLink-attributes or contains the line string element.

7.5.3 Simple Geometric Primitives (2-dimensional)

Complex Type AbstractSurfaceType

```
<complexType name="AbstractSurfaceType">
  <complexContent>
    <extension base="gml:AbstractGeometricPrimitiveType" />
  </complexContent>
</complexType>
```

An abstraction of a surface to support the different levels of complexity. A surface is always a continuous region of a plane.

Element _Surface

```
<element name="_Surface" type="gml:AbstractSurfaceType" abstract="true"
```



```
substitutionGroup="gml:_GeometricPrimitive" />
```

The "_Surface" element is the abstract head of the substitution group for all (continuous) surface elements.

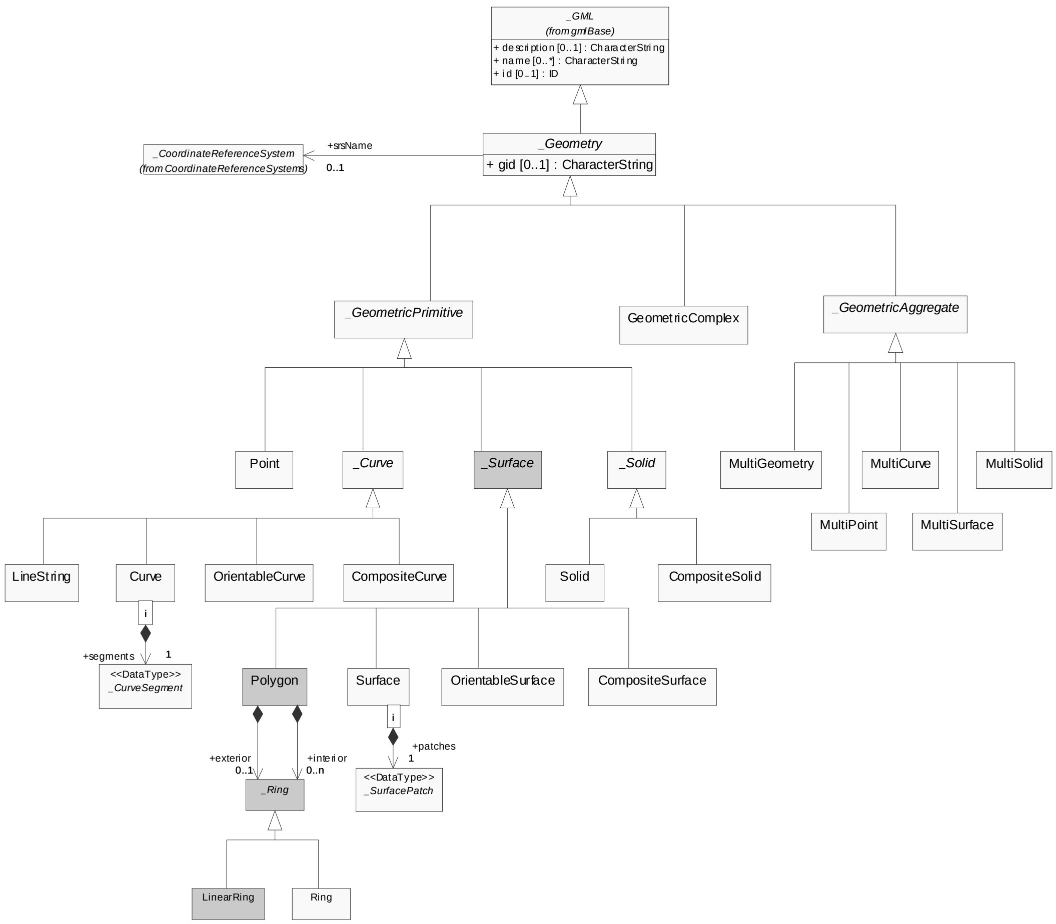


Figure 7.5-7 – Simple Geometric Primitives (2-dimensional)

Complex Type SurfacePropertyType

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

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.

Element surfaceProperty

```
<element name="surfaceProperty" type="gml:SurfacePropertyType" />
```

This property element either references a surface via the XLink-attributes or contains the surface element. surfaceProperty is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for _Surface.

Complex Type SurfaceArrayPropertyType

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

A container for an array of surfaces. The elements are always contained in the array property, referencing geometry elements or arrays of geometry elements via XLinks is not supported.

Element surfaceArrayProperty

```
<element name="surfaceArrayProperty" type="gml:SurfaceArrayPropertyType" />
```

This property element contains a list of surface elements. surfaceArrayProperty is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for a list of _Surfaces.

Complex Type PolygonType

```
<complexType name="PolygonType">
    <complexContent>
        <extension base="gml:AbstractSurfaceType">
            <sequence>
                <element ref="gml:exterior" minOccurs="0" />
                <element ref="gml:interior" minOccurs="0" maxOccurs="unbounded" />
            </sequence>
        </extension>
    </complexContent>
```

```
</complexType>
```

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. GM_Polygon of ISO 19107 is implemented by PolygonPatch.

The elements “exterior” and “interior” describe the surface boundary of the polygon and are specified below.

Element Polygon

```
<element name="Polygon" type="gml:PolygonType" substitutionGroup="gml:_Surface" />
```

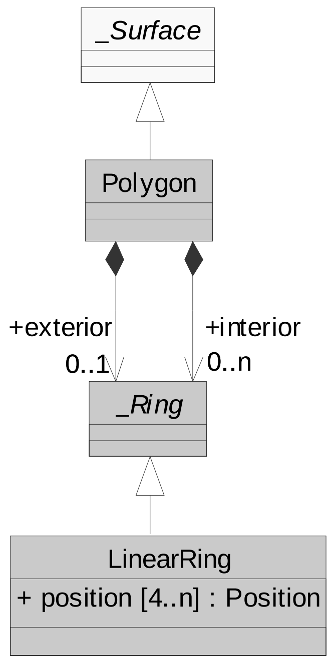


Figure 7.5-8 – Polygon

Complex Type AbstractRingType

```
<complexType name="AbstractRingType" abstract="true">
    <complexContent>
        <extension base="gml:AbstractGeometryType" />
    </complexContent>
</complexType>
```

An abstraction of a ring to support surface boundaries of different complexity.

Element _Ring

```
<element name="_Ring" type="gml:AbstractRingType" abstract="true" />
```

The "_Ring" element is the abstract head of the substituition group for all closed boundaries of a surface patch.

Complex Type AbstractRingPropertyType

```
<complexType name="AbstractRingPropertyType">
    <sequence>
        <element ref="gml:_Ring" />
    </sequence>
</complexType>
```

Encapsulates a ring to represent the surface boundary property of a surface.

Element exterior

```
<element name="exterior" type="gml:AbstractRingPropertyType" />
```

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.

Element interior

```
<element name="interior" type="gml:AbstractRingPropertyType" />
```

A boundary of a surface consists of a number of rings. The "interior" rings separate the surface / surface patch from the area enclosed by the rings.

Element outerBoundaryIs

```
<element name="outerBoundaryIs" type="gml:AbstractRingPropertyType" substitutionGroup="gml:exterior" />
```

Deprecated with GML 3.0, included only for backwards compatibility with GML 2. Use "exterior" instead.

Element innerBoundaryIs

```
<element name="innerBoundaryIs" type="gml:AbstractRingPropertyType" substitutionGroup="gml:interior" />
```

Deprecated with GML 3.0, included only for backwards compatibility with GML 2. Use "interior" instead.

Complex Type LinearRingType

```
<complexType name="LinearRingType">
```

```
    <complexContent>
        <extension base="gml:AbstractRingType">
            <sequence>
                <choice>
                    <choice minOccurs="4" maxOccurs="unbounded">
                        <element ref="gml:pos" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:coordinates" />
                    <element ref="gml:coord" minOccurs="4" maxOccurs="unbounded" />
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

A LinearRing is defined by four or more coordinate tuples, with linear interpolation between them; the first and last coordinates must be coincident.

GML supports two different ways to specify the control points of a linear ring:

- A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this ring, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this ring (reuse of existing points).
- The "coordinates" element allows for a compact way to specify 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 coordinate list must be at least four.

The use of the element “coord” is deprecated with GML version 3.0 and included for backwards compatibility with GML 2. Use "pos" elements instead.

Element LinearRing

```
<element name="LinearRing" type="gml:LinearRingType" substitutionGroup="gml:_Ring" />
```

Complex Type LinearRingPropertyType

```
<complexType name="LinearRingPropertyType">
    <choice>
        <element ref="gml:LinearRing" />
    </choice>
</complexType>
```

Encapsulates a ring to represent properties in features or geometry collections.

Complex Type PolygonPropertyType

```
<complexType name="PolygonPropertyType">
    <sequence>
        <element ref="gml:Polygon"  minOccurs="0" />
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```

This type is deprecated with GML 3 and shall not be used. It is included for backwards compatibility with GML 2. Use SurfacePropertyType instead.

A property that has a polygon 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.

Element polygonProperty

```
<element name="polygonProperty" type="gml:PolygonPropertyType" />
```

Deprecated with GML 3.0 and included only for backwards compatibility with GML 2.0. Use "surfaceProperty" instead.

This property element either references a polygon via the XLink-attributes or contains the polygon element.

7.5.4 More Geometric Primitives (1-, 2- and 3-dimensional)

7.5.4.1 Overview

Beside the “simple” geometric primitives specified in the previous clause, this clause specifies additional, more extensive primitives to describe real world situations which require a more expressive geometry model.

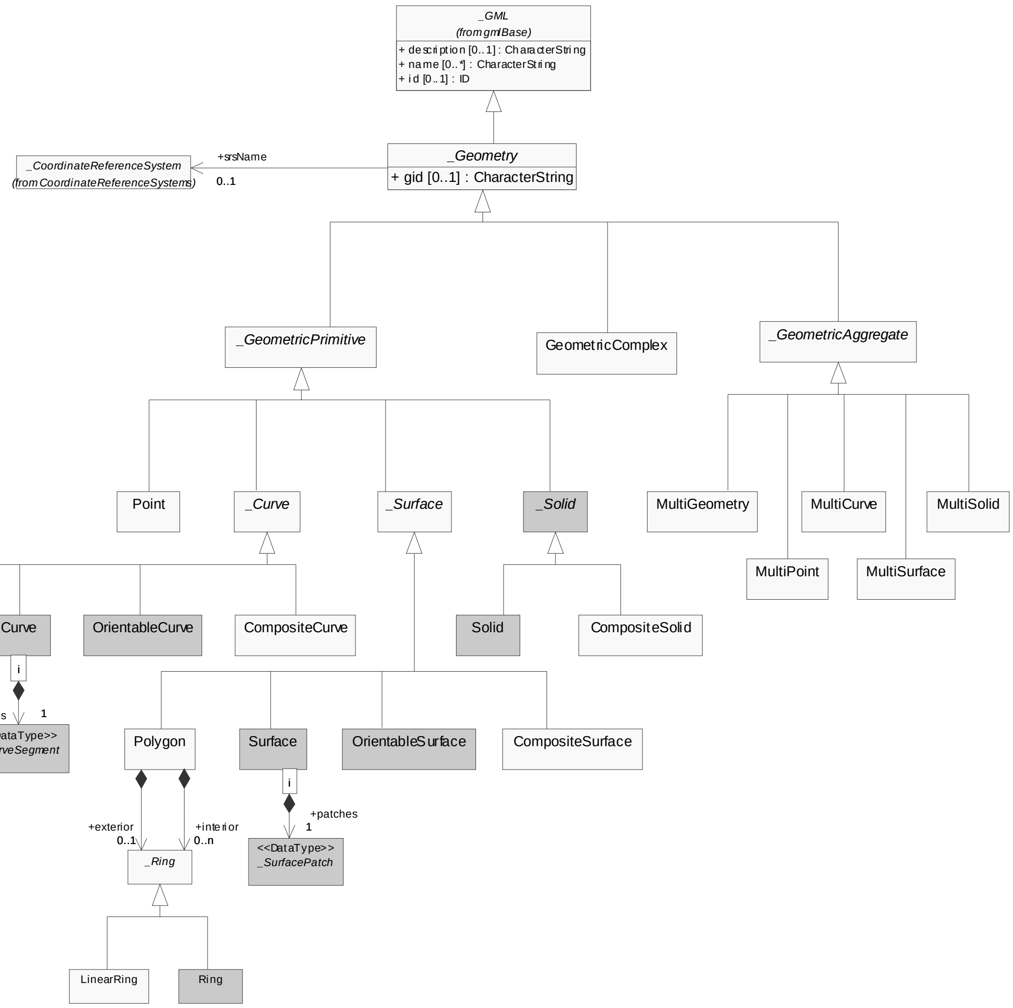


Figure 7.5-9 – Additional Geometric Primitives (1-, 2- and 3-dimensional)

7.5.4.2 Additional 1-dimensional geometric primitives

<u>*Complex Type* CurveType</u>

```
<complexType name="CurveType">
    <complexContent>
        <extension base="gml:AbstractCurveType">
            <sequence>
                <element ref="gml:segments" />
            </sequence>
        </extension>
    </complexContent>
```



```
</complexType>
```

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. Each curve segment within a curve may be defined using a different interpolation method. 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.

The element “segments” encapsulates the segments of the curve.

Element Curve

```
<element name="Curve" type="gml:CurveType" substitutionGroup="gml:_Curve" />
```

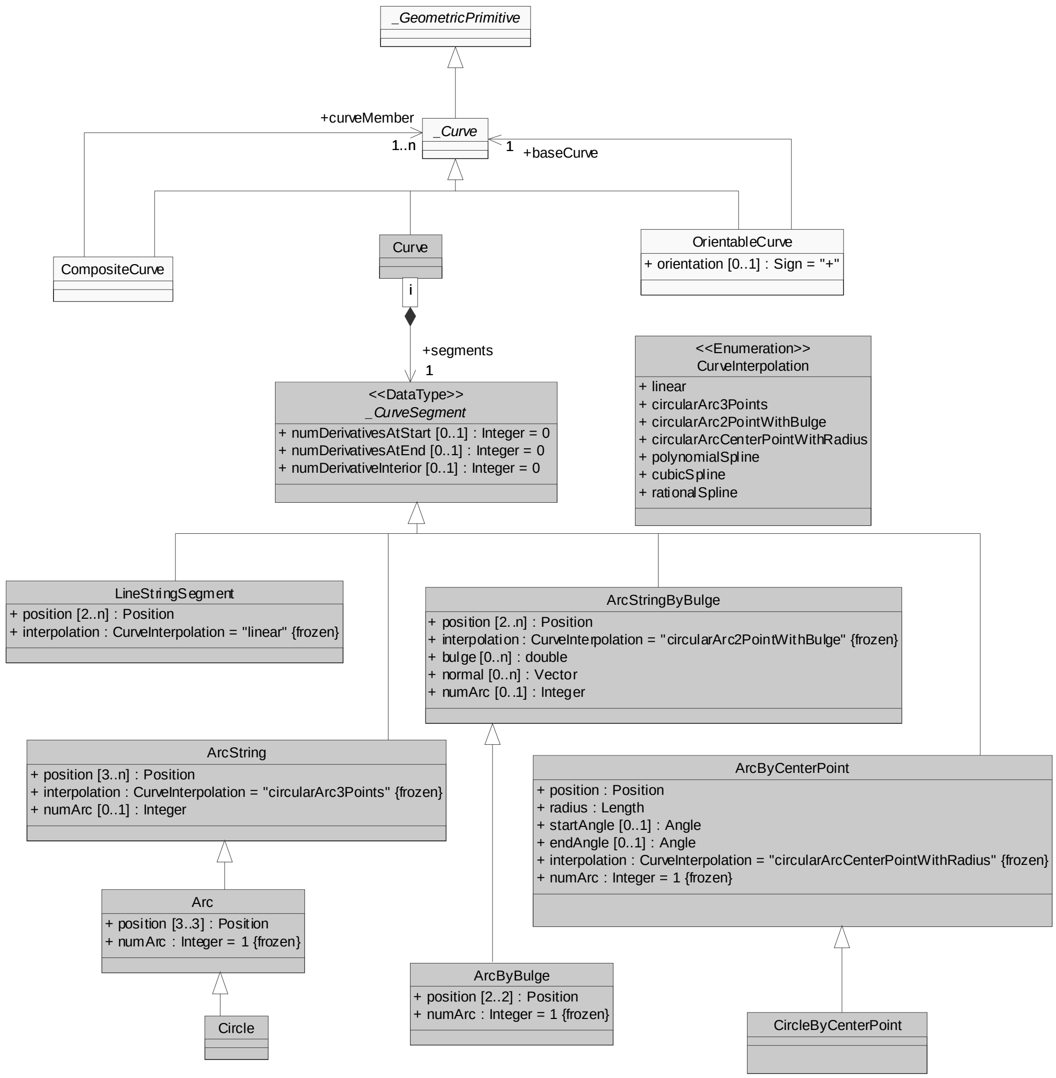


Figure 7.5-10 – Curve and Curve Segments (1)

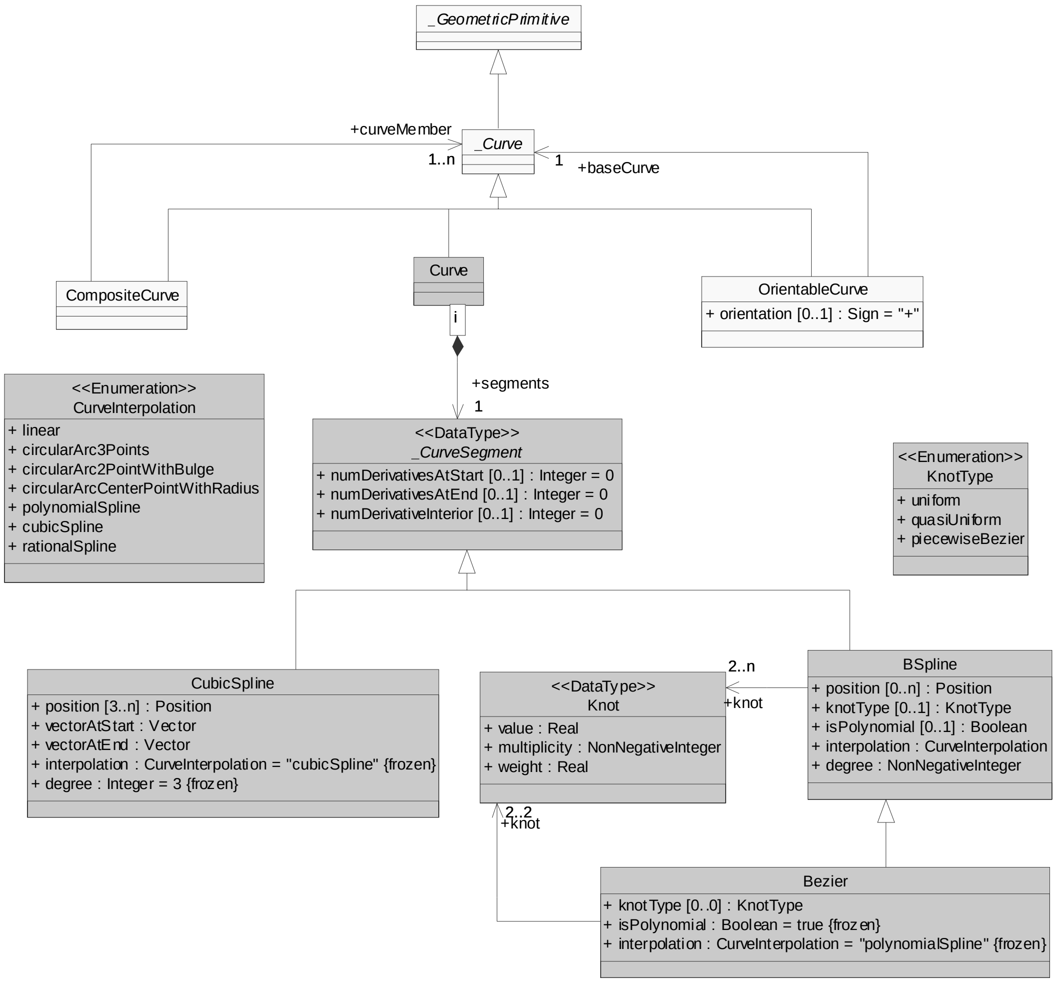


Figure 7.5-11 – Curve and Curve Segments (2)

Complex Type AbstractCurveSegmentType

```
<complexType name="AbstractCurveSegmentType" abstract="true">
    <attribute name="numDerivativesAtStart" type="integer" use="optional" default="0" />
    <attribute name="numDerivativesAtEnd" type="integer" use="optional" default="0" />
    <attribute name="numDerivativeInterior" type="integer" use="optional" default="0" />
</complexType>
```

Curve segment defines a homogeneous segment of a curve.

The attribute "numDerivativesAtStart" specifies the type of continuity between this curve segment and its predecessor. If this is the first curve segment in the curve, one of these values, as appropriate, is ignored. The default value of "0" means simple continuity, which is a mandatory minimum level of continuity. This level is referred to as "C 0 " in mathematical texts. A value of 1 means that the function and its first derivative are continuous at the appropriate end point: "C 1 " continuity. A value of "n" for any integer means the function and its first n derivatives are continuous: "C n " continuity.

The attribute "numDerivativesAtEnd" specifies the type of continuity between this curve segment and its successor. If this is the last curve segment in the curve, one of these values, as appropriate, is ignored. The default value of "0" means simple continuity, which is a mandatory minimum level of continuity. This level is referred to as "C 0 " in mathematical texts. A value of 1 means that the function and its first derivative are continuous at the appropriate end point: "C 1 " continuity. A value of "n" for any integer means the function and its first n derivatives are continuous: "C n " continuity.

The attribute "numDerivativesInterior" specifies the type of continuity that is guaranteed interior to the curve. The default value of "0" means simple continuity, which is a mandatory minimum level of continuity. This level is referred to as "C 0 " in mathematical texts. A value of 1 means that the function and its first derivative are continuous at the appropriate end point: "C 1 " continuity. A value of "n" for any integer means the function and its first n derivatives are continuous: "C n " continuity.

> NOTE: Use of these attributes is only appropriate when the basic curve definition is an underdetermined system. For example, line string segments cannot support continuity above C 0 , since there is no spare control parameter to adjust the incoming angle at the end points of the segment. Spline functions on the other hand often have extra degrees of freedom on end segments that allow them to adjust the values of the derivatives to support C 1 or higher continuity.

Element _CurveSegment

```
<element name="_CurveSegment" type="gml:AbstractCurveSegmentType" abstract="true" />
```

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.

Complex Type CurveSegmentArrayPropertyType

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

A container for an array of curve segments.

Element segments


```
<element name="segments" type="gml:CurveSegmentArrayPropertyType" />
```

This property element contains a list of curve segments. The order of the elements is significant and shall be preserved when processing the array.

Simple Type CurveInterpolationType

```
<simpleType name="CurveInterpolationType">
    <restriction base="string">
        <enumeration value="linear" />
        <enumeration value="geodesic" />
        <enumeration value="circularArc3Points" />
        <enumeration value="circularArc2PointWithBulge" />
        <enumeration value="circularArcCenterPointWithRadius" />
        <enumeration value="elliptical" />
        <enumeration value="clothoid" />
        <enumeration value="conic" />
        <enumeration value="polynomialSpline" />
        <enumeration value="cubicSpline" />
        <enumeration value="rationalSpline" />
    </restriction>
</simpleType>
```

CurveInterpolationType is a list of codes that may be used to identify the interpolation mechanisms specified by an application schema.

Complex Type LineStringSegmentType

```
<complexType name="LineStringSegmentType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>
                <choice>
                    <choice minOccurs="2" maxOccurs="unbounded">
                        <element ref="gml:pos" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:coordinates" />
                </choice>
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="linear" />
        </extension>
```

```
    </complexContent>
</complexType>
```

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.

GML supports two different ways to specify the control points of a curve segment.

- A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this curve segment, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this curve segment (reuse of existing points).
- The "coordinates" element allows for a compact way to specify the coordinates of the control points, if all control points are in the same coordinate reference systems and belong to this curve segment only. The number of direct positions in the coordinate list must be at least two and include the start and end point.

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".

Element LineStringSegment

```
<element name="LineStringSegment" type="gml:LineStringSegmentType"
    substitutionGroup="gml:_CurveSegment" />
```

Complex Type ArcStringType

```
<complexType name="ArcStringType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>
                <choice>
                    <choice minOccurs="3" maxOccurs="unbounded">
                        <element ref="gml:pos" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:coordinates" />
                </choice>
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="circularArc3Points" />
```



```
      <attribute name="numArc" type="integer" use="optional" />
    </extension>
  </complexContent>
</complexType>
```

An ArcString is a curve segment that uses three-point circular arc interpolation.

GML supports two different ways to specify the control points of a curve segment.

- A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this curve segment, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this curve segment (reuse of existing points).
- The "coordinates" element allows for a compact way to specify the coordinates of the control points, if all control points are in the same coordinate reference systems and belong to this curve segment only. The number of direct positions in the coordinate list must be at least three and include the start and end point.

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 an ArcString the interpolation is fixed as "circularArc3Points".

The number of arcs in the arc string can be explicitly stated in the attribute “numArc”. The number of control points in the arc string must be 2 * numArc + 1.

Element ArcString

```
<element name="ArcString" type="gml:ArcStringType" substitutionGroup="gml:_CurveSegment" />
```

Complex Type ArcType

```
<complexType name="ArcType">
  <complexContent>
    <restriction base="gml:ArcStringType">
      <sequence>
        <choice>
          <choice minOccurs="3" maxOccurs="3">
            <element ref="gml:pos" />
            <element ref="gml:pointRep" />
          </choice>
          <element ref="gml:coordinates" />
        </choice>
      </sequence>
```

```
            <attribute name="numArc" type="integer" use="optional" fixed="1" />
        </restriction>
    </complexContent>
</complexType>
```

An Arc is an arc string with only one arc unit, i.e. three control points.

GML supports two different ways to specify the control points of a curve segment.

- A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this curve segment, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this curve segment (reuse of existing points).
- The "coordinates" element allows for a compact way to specify the coordinates of the control points, if all control points are in the same coordinate reference systems and belong to this curve segment only. The number of direct positions in the coordinate list must be three and include the start and end point.

An arc is an arc string consisting of a single arc, the attribute “numArc” is fixed to "1".

Element Arc

```
<element name="Arc" type="gml:ArcType" substitutionGroup="gml:ArcString" />
```

Complex Type CircleType

```
<complexType name="CircleType">
    <complexContent>
        <extension base="gml:ArcType" />
    </complexContent>
</complexType>
```

A Circle is an arc whose first and last control points coincide to form a full circle. The "start" and "end" bearing are equal and shall be the bearing for the first controlPoint listed. This still requires at least 3 distinct non-co-linear points to be unambiguously defined. The arc is simply extended until the first point is encountered.

Element Circle

```
<element name="Circle" type="gml:CircleType" substitutionGroup="gml:Arc" />
```

Complex Type ArcStringByBulgeType

```
<complexType name="ArcStringByBulgeType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
```

```
<sequence>
    <choice>
        <choice minOccurs="2" maxOccurs="unbounded">
            <element ref="gml:pos" />
            <element ref="gml:pointRep" />
        </choice>
        <element ref="gml:coordinates" />
    </choice>
    <element name="bulge" type="gml:double" maxOccurs="unbounded"/>
    <element name="normal" type="gml:VectorType" maxOccurs="unbounded" />
</sequence>
<attribute name="interpolation" type="gml:CurveInterpolationType"
    fixed="circularArc2PointWithBulge" />
<attribute name="numArc" type="integer" use="optional" />
</extension>
</complexContent>
</complexType>
```

This variant of the arc computes the mid points of the arcs instead of storing the coordinates directly. The control point sequence consists of the start and end points of each arc plus the bulge.

GML supports two different ways to specify the control points of a curve segment.

- A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this curve segment, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this curve segment (reuse of existing points).
- The "coordinates" element allows for a compact way to specify the coordinates of the control points, if all control points are in the same coordinate reference systems and belong to this curve segment only. The number of direct positions in the coordinate list must be at least two and include the start and end point.

The element “bulge” controls the offset of each arc's midpoint. The "bulge" is the real number multiplier for the normal that determines the offset direction of the midpoint of each arc. The length of the bulge sequence is exactly 1 less than the length of the control point array, since a bulge is needed for each pair of adjacent points in the control point array. The bulge is not given by a distance, since it is simply a multiplier for the normal.

The midpoint of the resulting arc is given by: midPoint = ((startPoint + endPoint)/2.0) + bulge*normal.

The element "normal" is a vector normal (perpendicular) to the chord of the arc, the line joining the first and last point of the arc. In a 2D coordinate system, there are only two possible directions for the normal, and it is often given as a signed real, indicating its length, with a positive sign indicating a left turn angle from the chord line, and a negative sign indicating a right turn from the chord. In 3D, the normal determines the plane of the arc, along with the start and endPoint of the arc.

The normal is usually a unit vector, but this is not absolutely necessary. If the normal is a zero vector, the geometric object becomes equivalent to the straight line between the two end points. The length of the normal sequence is exactly the same as for the bulge sequence, 1 less than the control point sequence length.

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 an ArcStringByBulge the interpolation is fixed as "circularArc2PointWithBulge".

The number of arcs in the arc string can be explicitly stated in the attribute “numArc”. The number of control points in the arc string must be numArc + 1.

Element ArcStringByBulge

```
<element name="ArcStringByBulge" type="gml:ArcStringByBulgeType" substitutionGroup="gml:_CurveSegment" />
```

Complex Type ArcByBulgeType

```
<complexType name="ArcByBulgeType">
    <complexContent>
        <restriction base="gml:ArcStringByBulgeType">
            <sequence>
                <choice>
                    <choice minOccurs="2" maxOccurs="2">
                        <element ref="gml:pos" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:coordinates" />
                </choice>
                <element name="bulge" type="gml:double" />
                <element name="normal" type="gml:VectorType"/>
            </sequence>
            <attribute name="numArc" type="integer" use="optional" fixed="1" />
        </restriction>
    </complexContent>
</complexType>
```

An ArcByBulge is an arc string with only one arc unit, i.e. two control points and one bulge.

GML supports two different ways to specify the control points of a curve segment.

- A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this curve segment, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this curve segment (reuse of existing points).
- The "coordinates" element allows for a compact way to specify the coordinates of the control points, if all control points are in the same coordinate reference systems and belong to this curve segment only. The number of direct positions in the coordinate list must be two and include the start and end point.

The element “bulge” controls the offset of each arc's midpoint. The "bulge" is the real number multiplier for the normal that determines the offset direction of the midpoint of each arc. The length of the bulge sequence is exactly 1 less than the length of the control point array, since a bulge is needed for each pair of adjacent points in the control point array. The bulge is not given by a distance, since it is simply a multiplier for the normal.

The midpoint of the resulting arc is given by: midPoint = ((startPoint + endPoint)/2.0) + bulge*normal.

For an Arc there is exactly one bulge.

The element "normal" is a vector normal (perpendicular) to the chord of the arc, the line joining the first and last point of the arc. In a 2D coordinate system, there are only two possible directions for the normal, and it is often given as a signed real, indicating its length, with a positive sign indicating a left turn angle from the chord line, and a negative sign indicating a right turn from the chord. In 3D, the normal determines the plane of the arc, along with the start and endPoint of the arc.

The normal is usually a unit vector, but this is not absolutely necessary. If the normal is a zero vector, the geometric object becomes equivalent to the straight line between the two end points. The length of the normal sequence is exactly the same as for the bulge sequence, 1 less than the control point sequence length.

For an Arc there is exactly one normal vector.

An arc is an arc string consisting of a single arc, the attribute “numArc” is fixed to "1".

Element ArcByBulge

```
<element name="ArcByBulge" type="gml:ArcByBulgeType" substitutionGroup="gml:ArcStringByBulge" />
```

Complex Type ArcByCenterPointType

```
<complexType name="ArcByCenterPointType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>
```

```
        <choice>
          <choice>
            <element ref="gml:pos" />
            <element ref="gml:pointRep" />
          </choice>
          <element ref="gml:coordinates" />
        </choice>
        <element name="radius" type="gml:LengthType" />
        <element name="startAngle" type="gml:AngleType" minOccurs="0" />
        <element name="endAngle" type="gml:AngleType" minOccurs="0" />
      </sequence>
      <attribute name="interpolation" type="gml:CurveInterpolationType"
        fixed="circularArcCenterPointWithRadius" />
      <attribute name="numArc" type="integer" use="required" fixed="1" />
    </extension>
  </complexContent>
</complexType>
```

This variant of the arc requires that the points on the arc have to be computed instead of storing the coordinates directly. The control point is the center point of the arc plus the radius and the bearing at start and end. This representation can be used only in 2D.

GML supports two different ways to specify the control points of a curve segment.

- A "pos" (DirectPositionType) or "pointRep" (PointPropertyType) element. The "pos" element contains a center point that is only part of this curve segment, a "pointRep" element contains a point that may be referenced from other geometry elements or reference another point defined outside of this curve segment (reuse of existing points).
- The "coordinates" element can be used to specify the coordinates of the center point, too. The number of direct positions in the coordinate list must be one.

The element “radius” specifies the radius of the arc.

The element “startAngle” specifies the bearing of the arc at the start.

The element “endAngle” specifies the bearing of the arc at the end.

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 an ArcByCenterPoint the interpolation is fixed as "circularArcCenterPointWithRadius".

Since this type describes always a single arc, the attribute “numArc” is fixed to "1".

Element ArcByCenterPoint

```
<element name="ArcByCenterPoint" type="gml:ArcByCenterPointType" substitutionGroup="gml:_CurveSegment" />
```

Complex Type CircleByCenterPointType

```
<complexType name="CircleByCenterPointType">
    <complexContent>
        <extension base="gml:ArcByCenterPointType" />
    </complexContent>
</complexType>
```

A CircleByCenterPoint is an ArcByCenterPoint with identical start and end angle to form a full circle. Again, this representation can be used only in 2D.

Element CircleByCenterPoint

```
<element name="CircleByCenterPoint" type="gml:CircleByCenterPointType"
    substitutionGroup="gml:ArcByCenterPoint" />
```

Complex Type CubicSplineType

```
<complexType name="CubicSplineType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
            <sequence>
                <choice>
                    <choice minOccurs="3" maxOccurs="unbounded">
                        <element ref="gml:pos" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:coordinates" />
                </choice>
                <element name="vectorAtStart" type="gml:VectorType" />
                <element name="vectorAtEnd" type="gml:VectorType" />
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="cubicSpline" />
            <attribute name="degree" type="integer" fixed="3" />
        </extension>
    </complexContent>
</complexType>
```

Cubic splines are similar to line strings in that they are a sequence of segments each with its own defining function. A cubic spline uses the control points and a set of derivative parameters to define a piecewise 3rd degree polynomial interpolation. Unlike line-strings, the parameterization by arc length is not necessarily still a polynomial.

The function describing the curve must be C2, that is, have a continuous 1st and 2nd derivative at all points, and pass through the controlPoints in the order given. Between the control points, the curve segment is defined by a cubic polynomial. At each control point, the polynomial changes in such a manner that the 1st and 2nd derivative vectors are the same from either side. The control parameters record must contain vectorAtStart, and vectorAtEnd which are the unit tangent vectors at controlPoint[1] and controlPoint[n] where n = controlPoint.count.

NOTE: Only the direction of the vectors is relevant, not their length.

GML supports two different ways to specify the control points of a curve segment.

- A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this curve segment, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this curve segment (reuse of existing points).
- The "coordinates" element allows for a compact way to specify the coordinates of the control points, if all control points are in the same coordinate reference systems and belong to this curve segment only. The number of direct positions in the coordinate list must be at least three and include the start and end point.

The element “vectorAtStart” is the unit tangent vector at the start point of the spline.

The element “vectorAtEnd” is the unit tangent vector at the end point of the spline.

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 CubicSpline the interpolation is fixed as "cubicSpline".

The attribute "degree" shall be the degree of the polynomial used for interpolation in this spline. Therefore the degree for a cubic spline is fixed to "3".

Element CubicSpline

```
<element name="CubicSpline" type="gml:CubicSplineType" substitutionGroup="gml:_CurveSegment" />
```

Complex Type KnotType

```
<complexType name="KnotType">
    <sequence>
        <element name="value" type="double" />
        <element name="multiplicity" type="nonNegativeInteger" />
        <element name="weight" type="double" />
    </sequence>
```



```
</complexType>
```

A knot is a breakpoint on a piecewise spline curve.

The property element "value" is the value of the parameter at the knot of the spline. The sequence of knots shall be a non-decreasing sequence. That is, each knot's value in the sequence shall be equal to or greater than the previous knot's value. The use of equal consecutive knots is normally handled using the multiplicity.

The property element "multiplicity" is the multiplicity of this knot used in the definition of the spline (with the same weight).

The property element "weight" is the value of the averaging weight used for this knot of the spline.

Complex Type KnotPropertyType

```
<complexType name="KnotPropertyType">
    <sequence>
        <element name="Knot" type="gml:KnotType" />
    </sequence>
</complexType>
```

Encapsulates a knot to use it in a geometric type.

Simple Type KnotTypesType

```
<simpleType name="KnotTypesType">
    <restriction base="string">
        <enumeration value="uniform" />
        <enumeration value="quasiUniform" />
        <enumeration value="piecewiseBezier" />
    </restriction>
</simpleType>
```

Defines allowed values for the knots` type. Uniform knots implies that all knots are of multiplicity 1 and they differ by a positive constant from the preceding knot. Knots are quasi-uniform if they are of multiplicity (degree + 1) at the ends, of multiplicity 1 elsewhere, and they differ by a positive constant from the preceding knot.

Complex Type BSplineType

```
<complexType name="BSplineType">
    <complexContent>
        <extension base="gml:AbstractCurveSegmentType">
```

```
            <sequence>
                <choice>
                    <choice minOccurs="0" maxOccurs="unbounded">
                        <element ref="gml:pos" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:coordinates" />
                </choice>
                <element name="degree" type="nonNegativeInteger" />
                <element name="knot" type="gml:KnotPropertyType" minOccurs="2" maxOccurs="unbounded" />
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType" default="polynomialSpline" />
            <attribute name="isPolynomial" type="boolean" use="optional" />
            <attribute name="knotType" type="gml:KnotTypesType" use="optional" />
        </extension>
    </complexContent>
</complexType>
```

A B-Spline is a piecewise parametric polynomial or rational curve described in terms of control points and basis functions. Knots are breakpoints on the curve that connect its pieces.

They are given as a non-decreasing sequence of real numbers. If the weights in the knots are equal then it is a polynomial spline. The degree is the algebraic degree of the basis functions.

GML supports two different ways to specify the control points of a curve segment.

- A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this curve segment, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this curve segment (reuse of existing points).
- The "coordinates" element allows for a compact way to specify the coordinates of the control points, if all control points are in the same coordinate reference systems and belong to this curve segment only.

The property element "degree" shall be the degree of the polynomial used for interpolation in this spline.

The property element "knot" shall be the sequence of distinct knots used to define the spline basis functions.

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 BSpline the interpolation can be either "polynomialSpline" or "rationalSpline", default is "polynomialSpline".

The attribute “isPolynomial” is set to “true” if this is a polynomial spline.

The attribute "knotType" gives the type of knot distribution used in defining this spline. This is for information only and is set according to the different construction-functions.

Element BSpline

```
<element name="BSpline" type="gml:BSplineType" substitutionGroup="gml:_CurveSegment" />
```

Complex Type BezierType

```
<complexType name="BezierType">
    <complexContent>
        <restriction base="gml:BSplineType">
            <sequence>
                <choice>
                    <choice minOccurs="0" maxOccurs="unbounded">
                        <element ref="gml:pos" />
                        <element ref="gml:pointRep" />
                    </choice>
                    <element ref="gml:coordinates" />
                </choice>
                <element name="degree" type="nonNegativeInteger" />
                <element name="knot" type="gml:KnotPropertyType" minOccurs="2" maxOccurs="2" />
            </sequence>
            <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="polynomialSpline" />
            <attribute name="isPolynomial" type="boolean" fixed="true" />
            <attribute name="knotType" type="gml:KnotTypesType" use="prohibited" />
        </restriction>
    </complexContent>
</complexType>
```

Bezier curves are polynomial splines that use Bezier or Bernstein polynomials for interpolation purposes. It is a special case of the B-Spline curve with two knots.

GML supports two different ways to specify the control points of a curve segment.

- A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this curve segment, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this curve segment (reuse of existing points).

- The "coordinates" element allows for a compact way to specify the coordinates of the control points, if all control points are in the same coordinate reference systems and belong to this curve segment only.

The property element "degree" shall be the degree of the polynomial used for interpolation in this spline.

The property element "knot" shall be the sequence of distinct knots used to define the spline basis functions.

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 Bezier the interpolation is fixed as "polynomialSpline".

The attribute “isPolynomial” shall be “true” as this is a polynomial spline.

The property "knotType" is not relevant for Bezier curve segments.

Element Bezier

```
<element name="Bezier" type="gml:BezierType" substitutionGroup="gml:BSpline" />
```

Element baseCurve

```
<element name="baseCurve" type="gml:CurvePropertyType" />
```

The property element “baseCurve” either references a curve via the XLink-attributes or contains the curve element. A curve element is any element which is substitutable for "_Curve".

Complex Type OrientableCurveType

```
<complexType name="OrientableCurveType">
    <complexContent>
        <extension base="gml:AbstractCurveType">
            <sequence>
                <element ref="gml:baseCurve" />
            </sequence>
            <attribute name="orientation" type="gml:SignType" default="+" />
        </extension>
    </complexContent>
</complexType>
```

OrientableCurve consists of a curve and an orientation. If the orientation is "+", then the OrientableCurve is identical to the baseCurve. If the orientation is "-", then the OrientableCurve is related to another _Curve with a parameterization that reverses the sense of the curve traversal.

The element “baseCurve” references or contains the base curve (positive orientation).

> NOTE: This definition allows for a nested structure, i.e. an OrientableCurve may use another OrientableCurve as its base curve.

If the attribute “orientation” is "+", then the OrientableCurve is identical to the baseCurve. If the orientation is "-", then the OrientableCurve is related to another _Curve with a parameterization that reverses the sense of the curve traversal. "+" is the default value.

Element OrientableCurve

```
<element name="OrientableCurve" type="gml:OrientableCurveType" substitutionGroup="gml:_Curve" />
```

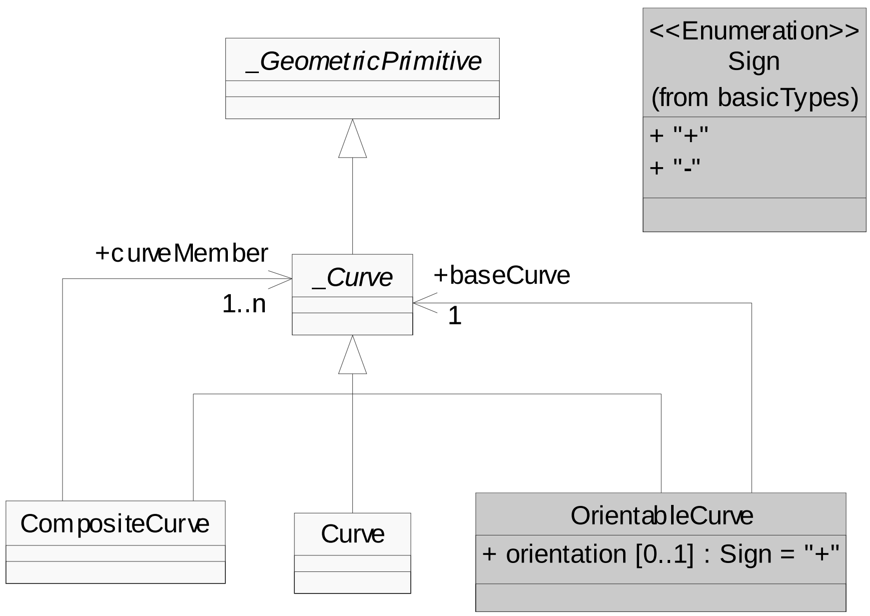


Figure 7.5-12 – Orientable Curve

7.5.4.3 Additional 2-dimensional geometric primitives

Complex Type SurfaceType

```
<complexType name="SurfaceType">
    <complexContent>
        <extension base="gml:AbstractSurfaceType">
            <sequence>
                <element ref="gml:patches" />
            </sequence>
        </extension>
    </complexContent>
```

</complexType>

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.

The element “patches” encapsulates the patches of the surface.

Element Surface

<element name="Surface" type="gml:SurfaceType" substitutionGroup="gml:_Surface" />

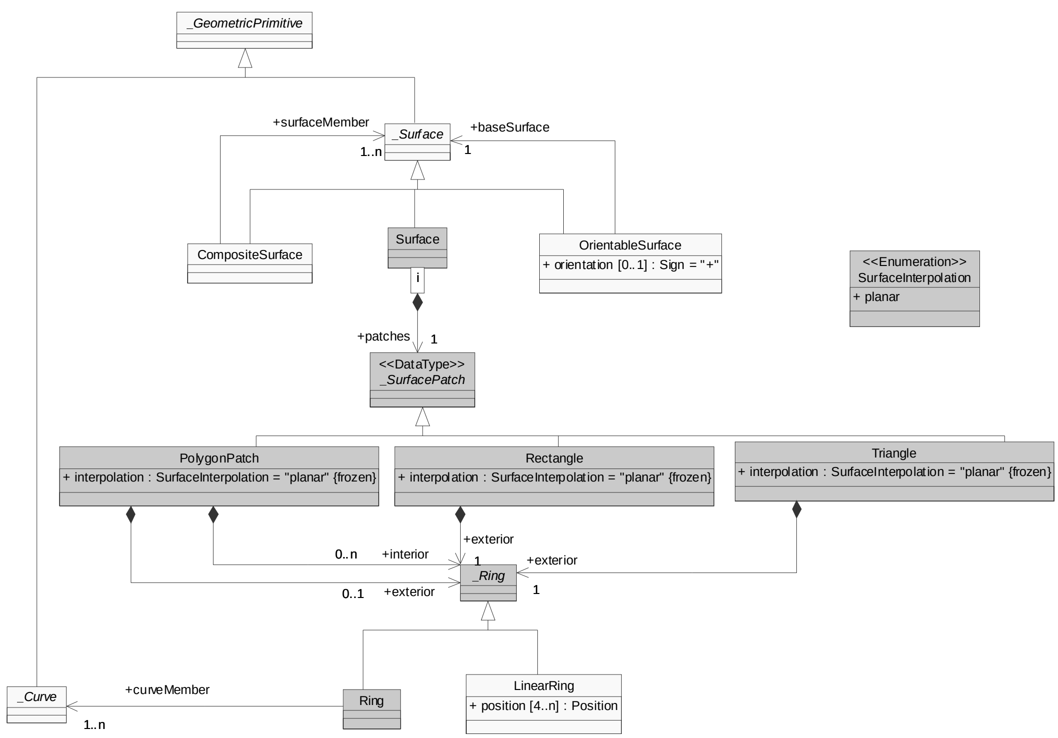


Figure 7.5-13 – Surface and Surface Patches

Complex Type AbstractSurfacePatchType

```
<complexType name="AbstractSurfacePatchType" abstract="true" />
```

A surface patch defines a homogenuous portion of a surface.

Element _SurfacePatch

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

The "_SurfacePatch" element is the abstract head of the substituition group for all surface patch elements describing a continuous portion of a surface.

Complex Type SurfacePatchArrayPropertyType

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

A container for an array of surface patches.

Element patches

```
<element name="patches" type="gml:SurfacePatchArrayPropertyType" />
```

This property element contains a list of surface patches. The order of the elements is significant and shall be preserved when processing the array.

Simple Type SurfaceInterpolationType

```
<simpleType name="SurfaceInterpolationType">
    <restriction base="string">
        <enumeration value="none" />
        <enumeration value="planar" />
        <enumeration value="spherical" />
        <enumeration value="elliptical" />
        <enumeration value="conic" />
        <enumeration value="tin" />
        <enumeration value="parametricCurve" />
        <enumeration value="polynomialSpline" />
        <enumeration value="rationalSpline" />
        <enumeration value="triangulatedSpline" />
    </restriction>
```

```
</simpleType>
```

SurfaceInterpolationType is a list of codes that may be used to identify the interpolation mechanisms specified by an application schema.

Complex Type PolygonPatchType

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

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.

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.

Element PolygonPatch

```
<element name="PolygonPatch" type="gml:PolygonPatchType" substitutionGroup="gml:_SurfacePatch" />
```

Complex Type TriangleType

```
<complexType name="TriangleType">
    <complexContent>
        <extension base="gml:AbstractSurfacePatchType">
            <sequence>
                <element ref="gml:exterior" />
            </sequence>
            <attribute name="interpolation" type="gml:SurfaceInterpolationType" fixed="planar" />
        </extension>
    </complexContent>
</complexType>
```

Represents a triangle as a surface with an outer boundary consisting of a linear ring. Note that this is a polygon (subtype) with no inner boundaries. The number of points in the linear ring must be four.

The Ring (element “exterior”) shall be a LinearRing and must form a triangle, the first and the last position must be co-incident.

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.

Element Triangle

```
<element name="Triangle" type="gml:TriangleType" substitutionGroup="gml:_SurfacePatch" />
```

Complex Type RectangleType

```
<complexType name="RectangleType">
    <complexContent>
        <extension base="gml:AbstractSurfacePatchType">
            <sequence>
                <element ref="gml:exterior" />
            </sequence>
            <attribute name="interpolation" type="gml:SurfaceInterpolationType" fixed="planar" />
        </extension>
    </complexContent>
</complexType>
```

Represents a rectangle as a surface with an outer boundary consisting of a linear ring. Note that this is a polygon (subtype) with no inner boundaries. The number of points in the linear ring must be five.

The Ring (element “exterior”) shall be a LinearRing and must form a rectangle; the first and the last position must be co-incident.

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.

Element Rectangle

```
<element name="Rectangle" type="gml:RectangleType" substitutionGroup="gml:_SurfacePatch" />
```

Element curveMember

```
<element name="curveMember" type="gml:CurvePropertyType" />
```

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".

The element is used, for example, in a Ring.

Complex Type RingType

```
<complexType name="RingType">
    <complexContent>
        <extension base="gml:AbstractRingType">
            <sequence>
                <element ref="gml:curveMember" maxOccurs="unbounded" />
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

A Ring is used to represent a single connected component of a surface boundary. It consists of a sequence of curves connected in a cycle (an object whose boundary is empty).

A Ring is structurally similar to a composite curve in that the endPoint of each curve in the sequence is the startPoint of the next curve in the Sequence. Since the sequence is circular, there is no exception to this rule. Each ring, like all boundaries, is a cycle and each ring is simple.

NOTE: Even though each Ring is simple, the boundary need not be simple. The easiest case of this is where one of the interior rings of a surface is tangent to its exterior ring.

The element "curveMember" references or contains one curve in the composite curve. The curves are contiguous, the collection of curves is ordered.

NOTE: This definition allows for a nested structure, i.e. a CompositeCurve may use, for example, another CompositeCurve as a curve member.

Element Ring

```
<element name="Ring" type="gml:RingType" substitutionGroup="gml:_Ring" />
```

Complex Type RingPropertyType

```
<complexType name="RingPropertyType">
```

```
        <sequence>
            <element ref="gml:Ring" />
        </sequence>
    </complexType>
```

Encapsulates a ring to represent properties in features or geometry collections.

Element baseSurface

```
    <element name="baseSurface" type="gml:SurfacePropertyType" />
```

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".

Complex Type OrientableSurfaceType

```
    <complexType name="OrientableSurfaceType">
        <complexContent>
            <extension base="gml:AbstractSurfaceType">
                <sequence>
                    <element ref="gml:baseSurface" />
                </sequence>
                <attribute name="orientation" type="gml:SignType" default="+" />
            </extension>
        </complexContent>
    </complexType>
```

OrientableSurface consists of a surface and an orientation. If the orientation is "+", then the OrientableSurface is identical to the baseSurface. If the orientation is "-", then the OrientableSurface is a reference to a Surface with an up-normal that reverses the direction for this OrientableSurface, the sense of "the top of the surface".

The element “baseSurface” references or contains the base surface (positive orientation).

If the attribute “orientation” is "+", then the OrientableSurface is identical to the baseSurface. If the orientation is "-", then the OrientableSurface is a reference to a Surface with an up-normal that reverses the direction for this OrientableSurface, the sense of "the top of the surface". "+" is the default value.

Element OrientableSurface

```
    <element name="OrientableSurface" type="gml:OrientableSurfaceType" substitutionGroup="gml:_Surface" />
```

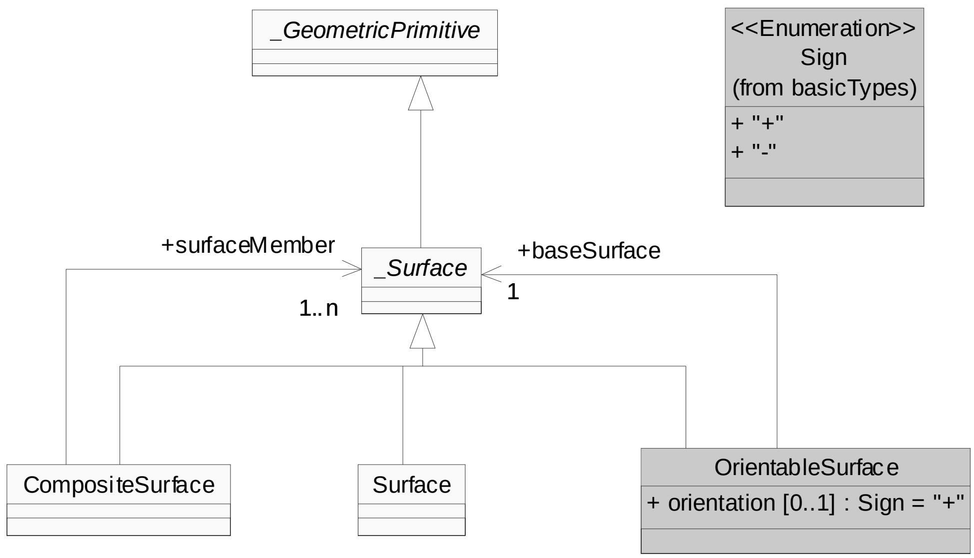


Figure 7.5-14 – Orientable Surface

7.5.4.4 3-dimensional geometric primitives

Complex Type AbstractSolidType

```
<complexType name="AbstractSolidType">
    <complexContent>
        <extension base="gml:AbstractGeometricPrimitiveType" />
    </complexContent>
</complexType>
```

An abstraction of a solid to support the different levels of complexity. A solid is always contiguous.

Element _Solid

```
<element name="_Solid" type="gml:AbstractSolidType" abstract="true"
    substitutionGroup="gml:_GeometricPrimitive" />
```

The "_Solid" element is the abstract head of the substituition group for all (continuous) solid elements.

Complex Type SolidPropertyType

```
<complexType name="SolidPropertyType">
    <sequence>
        <element ref="gml:_Solid"  minOccurs="0" />
    </sequence>
```

```
    <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```

A property that has a solid 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.

Element solidProperty

```
<element name="solidProperty" type="gml:SolidPropertyType" />
```

This property element either references a solid via the XLink-attributes or contains the solid element. solidProperty is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for _Solid.

Complex Type SolidArrayPropertyType

```
<complexType name="SolidArrayPropertyType">
    <sequence>
        <element ref="gml:_Solid"  minOccurs="0" maxOccurs="unbounded" />
    </sequence>
</complexType>
```

A container for an array of solids. The elements are always contained in the array property, referencing geometry elements or arrays of geometry elements is not supported.

Element solidArrayProperty

```
<element name="solidArrayProperty" type="gml:SolidArrayPropertyType" />
```

This property element contains a list of solid elements. solidArrayProperty is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for a list of _Solid.

Complex Type SolidType

```
<complexType name="SolidType">
    <complexContent>
        <extension base="gml:AbstractSolidType">
            <sequence>
                <element name="exterior" type="gml:SurfacePropertyType" minOccurs="0" />
                <element name="interior" type="gml:SurfacePropertyType" minOccurs="0" maxOccurs="unbounded"
/>
```



```
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

A solid is the basis for 3-dimensional geometry. The extent of a solid is defined by the boundary surfaces (shells). A shell is represented by a composite surface, where every shell is used to represent a single connected component of the boundary of a solid. It consists of a composite surface (a list of orientable surfaces) connected in a topological cycle (an object whose boundary is empty). Unlike a Ring, a shell's elements have no natural sort order. Like Rings, shells are simple.

The element “exterior” specifies the outer boundary of the solid. Boundaries of solids are similar to surface boundaries. In normal 3-dimensional Euclidean space, one (composite) surface is distinguished as the exterior. In the more general case, this is not always possible.

The element “interior” specifies the inner boundary of the solid. Boundaries of solids are similar to surface boundaries.

<u>*Element*</u> <u>Solid</u>

```
<element name="Solid" type="gml:SolidType" substitutionGroup="gml:_Solid" />
```

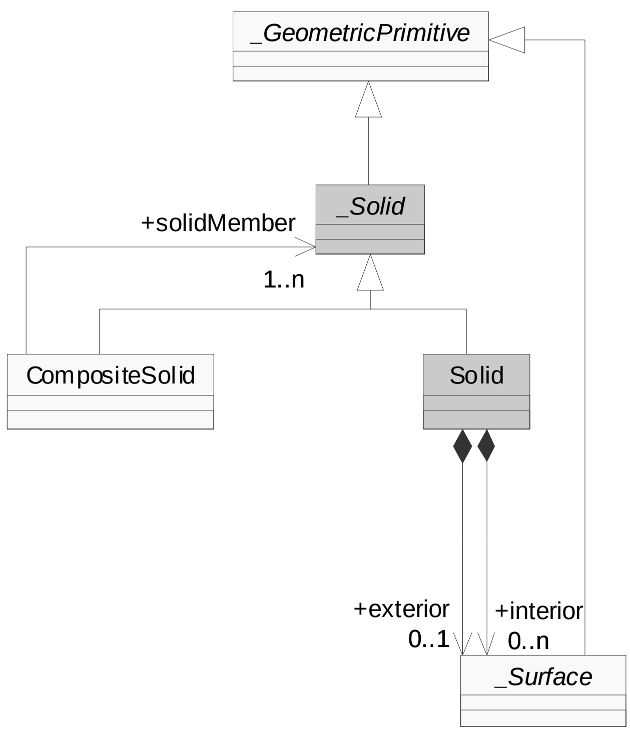


Figure 7.5-15 – Solid

7.5.5 Geometric Complex and geometric composites

A geometric complex is a set of primitive geometric objects (in a common coordinate system) whose interiors are disjoint. Further, if a primitive is in a geometric complex, then there exists a set of primitives in that complex whose point-wise union is the boundary of this first primitive.

A GeometricComplex in GML is a collection of geometrically disjoint, simple primitives. If a geometric primitive (other than a Point) is in a particular GeometricComplex, then there exists a set of primitives of lower dimension in the same complex that form the boundary of this primitive.

A geometric composite (GML specifies CompositeCurve, CompositeSurface or CompositeSolid) represents a geometric complex with an underlying core geometry that is isomorphic to a primitive. Thus, a composite curve is a collection of curves whose geometry could be viewed as a curve (albeit one with a complex inner structure). Composites are intended for use as attribute values in datasets in which the underlying geometry has been decomposed, usually to expose its topological nature.

The members of a geometric composite shall represent a homogeneous collection of geometric primitives whose union would be the core geometry of the composite. The complex would include all member primitives *and* all primitives on the boundary of these primitives, and so forth until Points are included. Thus the „member“ properties in CompositeCurve, CompositeSurface and CompositeSolid represent a subset of the „element“ property of GeometricComplex.

Geometric complexes and composites shall be used in application schemas where the sharing of geometry is important.

For more information about geometric complexes and an extensive discussion about complexes/composites/primitives and their commonalities/differences see Topic 1 of the Abstract Specification of OGC (ISO DIS 19107).

As XML Schema does not support the concept of “multiple inheritance” which is used in ISO DIS 19107 to express the duality of the geometric composites (as an open primitive and as a closed complex) in the GML geometry model, the composites inherit from AbstractGeometricPrimitiveType only. However, by using a <choice> element, a composite can be used in any property, which expects a GeometricComplex as its value.

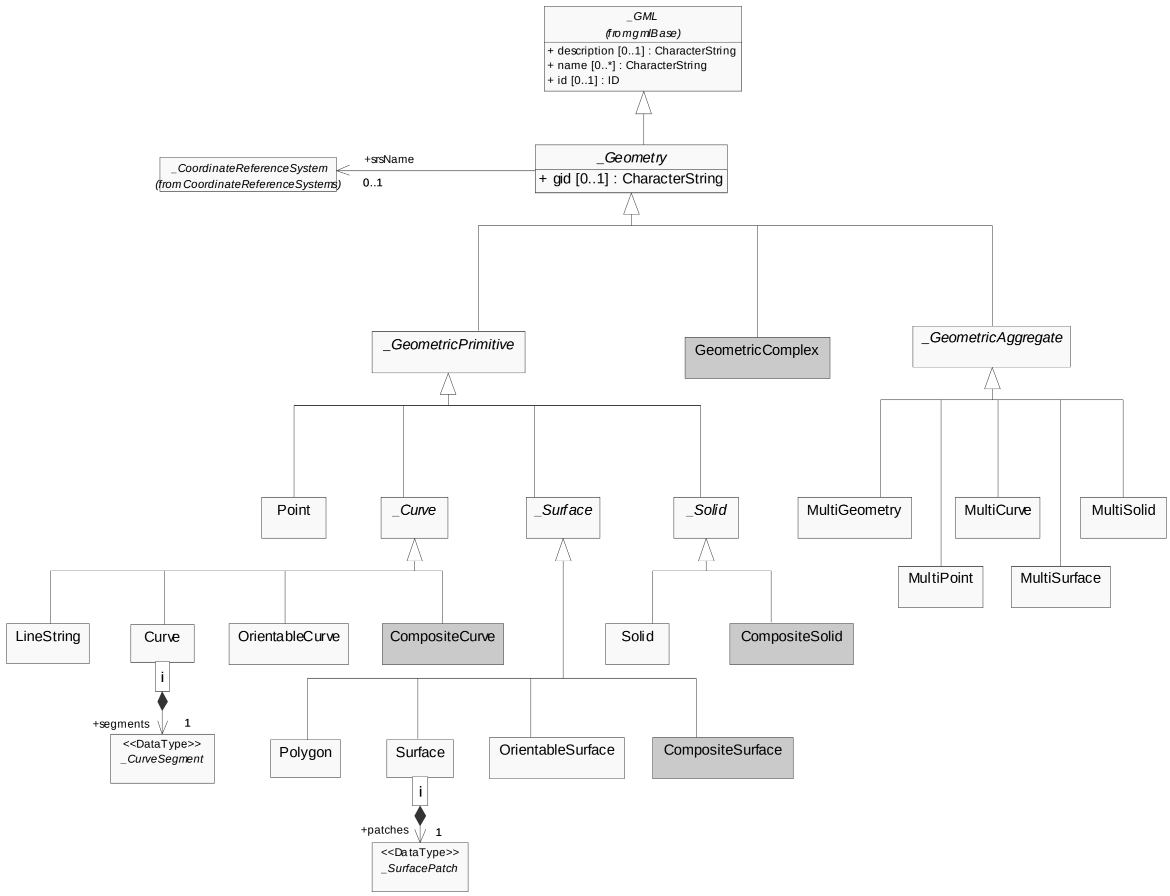


Figure 7.5-16 – Geometric Complex and composite geometries

Complex Type CompositeCurveType

```
<complexType name="CompositeCurveType">
    <complexContent>
        <extension base="gml:AbstractCurveType">
            <sequence>
                <element ref="gml:curveMember" maxOccurs="unbounded" />
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

A CompositeCurve is defined by a sequence of (orientable) curves such that the each curve in the sequence terminates at the start point of the subsequent curve in the list.

The element “curveMember” references or contains one curve in the composite curve.

The curves are contiguous, the collection of curves is ordered.

> NOTE: This definition allows for a nested structure, i.e. a CompositeCurve may use, for example, another CompositeCurve as a curve member.

Element CompositeCurve

```
<element name="CompositeCurve" type="gml:CompositeCurveType" substitutionGroup="gml:_Curve" />
```

Complex Type CompositeSurfaceType

```
<complexType name="CompositeSurfaceType">
    <complexContent>
        <extension base="gml:AbstractSurfaceType">
            <sequence>
                <element ref="gml:surfaceMember" maxOccurs="unbounded" />
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

A CompositeSurface is defined by a set of orientable surfaces. A composite surface is geometry type with all the geometric properties of a (primitive) surface. Essentially, a composite surface is a collection of surfaces that join in pairs on common boundary curves and which, when considered as a whole, form a single surface.

The element “surfaceMember” references or contains one surface in the composite surface. The surfaces are contiguous.

> NOTE: This definition allows for a nested structure, i.e. a CompositeSurface may use, for example, another CompositeSurface as a member.

Element CompositeSurface

```
<element name="CompositeSurface" type="gml:CompositeSurfaceType" substitutionGroup="gml:_Surface" />
```

Complex Type CompositeSolidType

```
<complexType name="CompositeSolidType">
    <complexContent>
        <extension base="gml:AbstractSolidType">
            <sequence>
                <element ref="gml:solidMember" maxOccurs="unbounded" />
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

A composite solid is a geometry type with all the geometric properties of a (primitive) solid.

Essentially, a composite solid is a collection of solids that join in pairs on common boundary surfaces and which, when considered as a whole, form a single solid.

The element “generator” references or contains one solid in the composite solid. The solids are contiguous.

> NOTE: This definition allows for a nested structure, i.e. a CompositeSolid may use, for example, another CompositeSolid as a member.

Element CompositeSolid

```
<element name="CompositeSolid" type="gml:CompositeSolidType" substitutionGroup="gml:_Solid" />
```

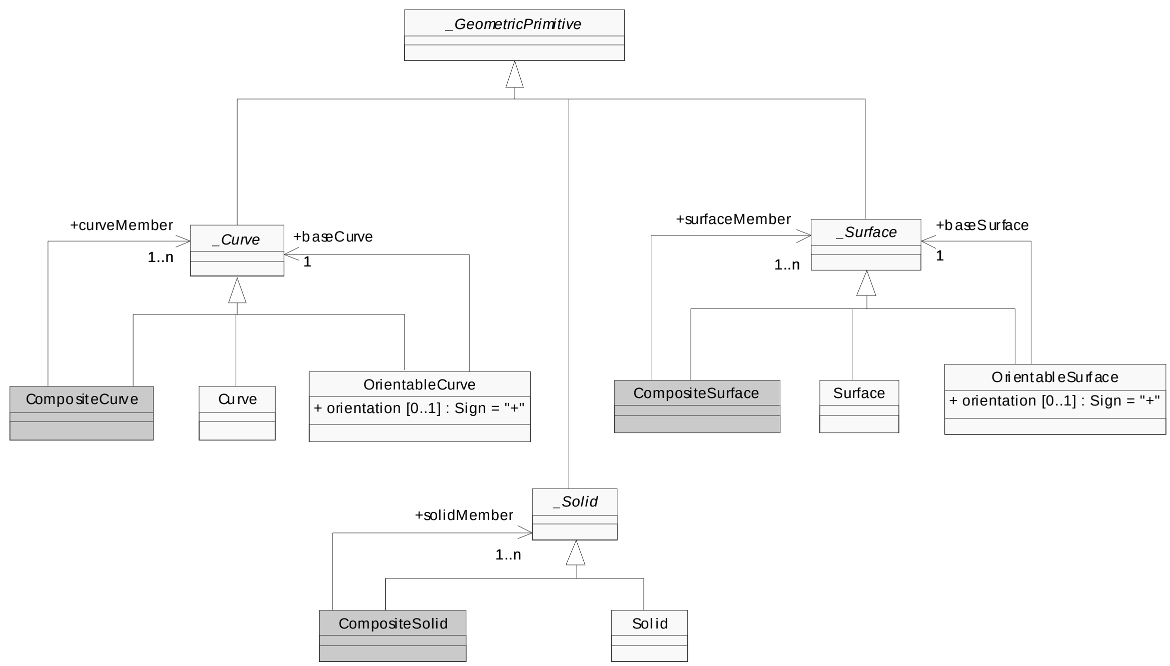


Figure 7.5-17 – Composite Curve, Surface and Solid

Complex Type GeometricComplexType

```
<complexType name="GeometricComplexType">
    <complexContent>
        <extension base="gml:AbstractGeometryType">
            <sequence>
                <element name="element" type="gml:GeometricPrimitivePropertyType" maxOccurs="unbounded" />
            </sequence>
```

```
        </extension>
    </complexContent>
</complexType>
```

A geometric complex. Every element “element” references or contains one geometric primitive (this includes composite geometries).

Element GeometricComplex

```
<element name="GeometricComplex" type="gml:GeometricComplexType" substitutionGroup="gml:_Geometry" />
```

Complex Type GeometricComplexPropertyType

```
<complexType name="GeometricComplexPropertyType">
    <choice minOccurs="0">
        <element ref="gml:GeometricComplex" />
        <element ref="gml:CompositeCurve" />
        <element ref="gml:CompositeSurface" />
        <element ref="gml:CompositeSolid" />
    </choice>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```

A property that has a geometric complex 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.

> NOTE: The allowed geometry elements contained in such a property (or referenced by it) have to be modeled by an XML Schema choice element since the composites (conceptually) inherit both from geometric complex *and* geometric primitive and are already part of the _GeometricPrimitive substitution group.

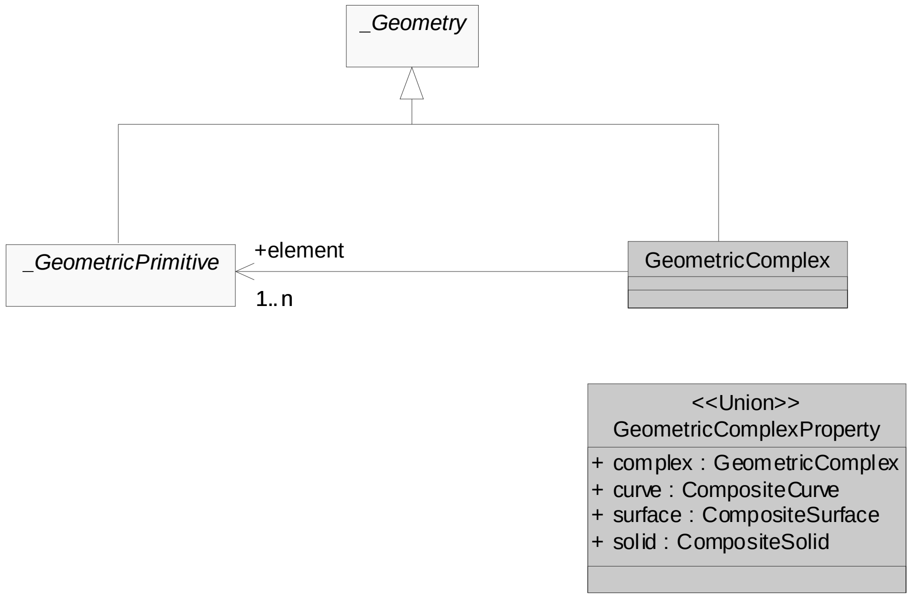


Figure 7.5-18 – Geometric Complex

7.5.6 Geometric Aggregates

7.5.6.1 Overview

Complex Type AbstractGeometricAggregateType

```
<complexType name="AbstractGeometricAggregateType" abstract="true">
    <complexContent>
        <extension base="gml:AbstractGeometryType" />
    </complexContent>
</complexType>
```

This is the abstract root type of the geometric aggregates.

Element _GeometricAggregate

```
<element name="_GeometricAggregate" type="gml:AbstractGeometricAggregateType" abstract="true"
    substitutionGroup="gml:_Geometry" />
```

The "_GeometricAggregate" element is the abstract head of the substitution group for all geometric aggremates.

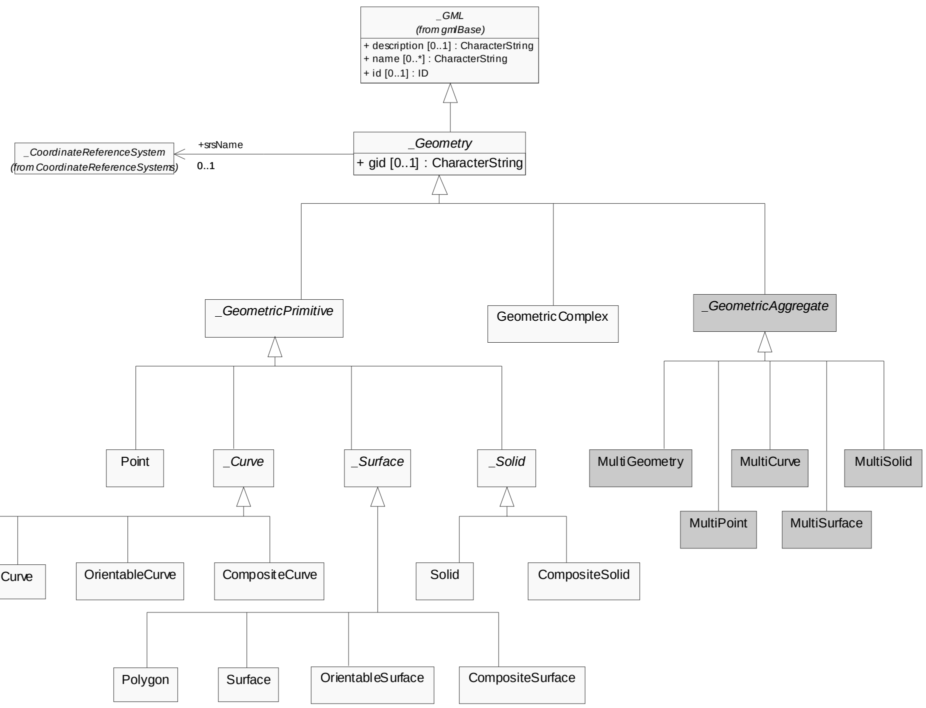


Figure 7.5-19 – Geometric Aggregates (Overview)

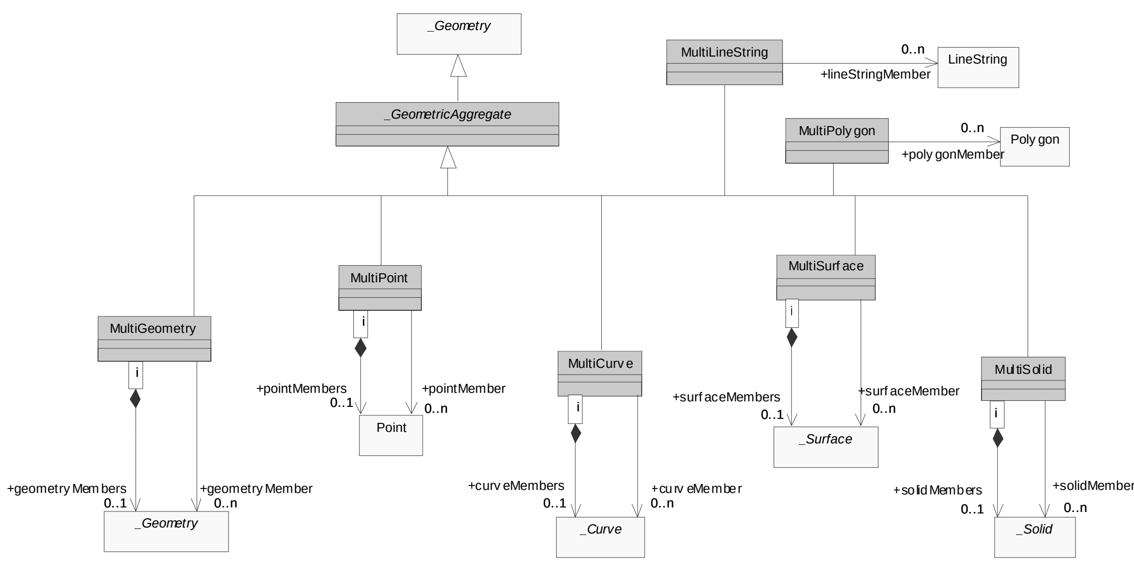


Figure 7.5-20 – Geometric Aggregates (Detail)

7.5.6.2 geometric aggregates of mixed dimensionality

Element geometryMember


```
<element name="geometryMember" type="gml:GeometryPropertyType" />
```

This property element either references a geometry element via the XLink-attributes or contains the geometry element.

Element geometryMembers

```
<element name="geometryMembers" type="gml:GeometryArrayPropertyType" />
```

This property element contains a list of geometry elements. The order of the elements is significant and shall be preserved when processing the array.

Complex Type MultiGeometryType

```
<complexType name="MultiGeometryType">
    <complexContent>
        <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
                <element ref="gml:geometryMember" minOccurs="0" maxOccurs="unbounded" />
                <element ref="gml:geometryMembers" minOccurs="0" />
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

A geometry collection must include one or more geometries, referenced through geometryMember elements.

The members of the geometric aggregate can be specified either using the "standard" property or the array property style. It is also valid to use both the "standard" and the array property style in the same collection.

NOTE: Array properties cannot reference remote geometry elements via XLinks.

Element MultiGeometry

```
<element name="MultiGeometry" type="gml:MultiGeometryType" substitutionGroup="gml:_GeometricAggregate" />
```

Complex Type MultiGeometryPropertyType

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

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.

Element multiGeometryProperty

```
<element name="multiGeometryProperty" type="gml:MultiGeometryPropertyType" />
```

This property element either references a geometric aggregate via the XLink-attributes or contains the "multi geometry" element. multiGeometryProperty is the predefined property, which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for _GeometricAggregate.

7.5.6.3 0-dimensional geometric aggregates

Element pointMember

```
<element name="pointMember" type="gml:PointPropertyType" />
```

This property element either references a Point via the XLink-attributes or contains the Point element.

Element pointMembers

```
<element name="pointMembers" type="gml:PointArrayPropertyType" />
```

This property element contains a list of points. The order of the elements is significant and shall be preserved when processing the array.

Complex Type MultiPointType

```
<complexType name="MultiPointType">
  <complexContent>
    <extension base="gml:AbstractGeometricAggregateType">
      <sequence>
        <element ref="gml:pointMember" minOccurs="0" maxOccurs="unbounded" />
        <element ref="gml:pointMembers" minOccurs="0" />
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

A MultiPoint is defined by one or more Points, referenced through pointMember elements.

The members of the geometric aggregate can be specified either using the "standard" property or the array property style. It is also valid to use both the "standard" and the array property style in the same collection.

NOTE: Array properties cannot reference remote geometry elements via XLinks.

Element MultiPoint

```
<element name="MultiPoint" type="gml:MultiPointType" substitutionGroup="gml:_GeometricAggregate" />
```

Complex Type MultiPointPropertyType

```
<complexType name="MultiPointPropertyType">
    <sequence>
        <element ref="gml:MultiPoint"  minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```

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.

Element multiPointProperty

```
<element name="multiPointProperty" type="gml:MultiPointPropertyType" />
```

This property element either references a point aggregate via the XLink-attributes or contains the "multi point" element. multiPointProperty is the predefined property, which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for MultiPoint.

7.5.6.4 1-dimensional geometric aggregates

Element curveMembers

```
<element name="curveMembers" type="gml:CurveArrayPropertyType" />
```

This property element contains a list of curves. The order of the elements is significant and shall be preserved when processing the array.

Complex Type MultiCurveType

```
<complexType name="MultiCurveType">
```

```
        <complexContent>
            <extension base="gml:AbstractGeometricAggregateType">
                <sequence>
                    <element ref="gml:curveMember" minOccurs="0" maxOccurs="unbounded" />
                    <element ref="gml:curveMembers" minOccurs="0" />
                </sequence>
            </extension>
        </complexContent>
    </complexType>
```

A MultiCurve is defined by one or more Curves, referenced through curveMember elements.

The members of the geometric aggregate can be specified either using the "standard" property or the array property style. It is also valid to use both the "standard" and the array property style in the same collection.

NOTE: Array properties cannot reference remote geometry elements via XLinks.

Element MultiCurve

```
<element name="MultiCurve" type="gml:MultiCurveType" substitutionGroup="gml:_GeometricAggregate" />
```

Complex Type MultiCurvePropertyType

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

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.

Element multiCurveProperty

```
<element name="multiCurveProperty" type="gml:MultiCurvePropertyType" />
```

This property element either references a curve aggregate via the XLink-attributes or contains the "multi curve" element. multiCurveProperty is the predefined property, which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for MultiCurve.

Element lineStringMember

```
<element name="lineStringMember" type="gml:LineStringPropertyType" />
```

Deprecated with GML 3.0 and included only for backwards compatibility with GML 2.0. Use "curveMember" instead.

This property element either references a line string via the XLink-attributes or contains the line string element.

Complex Type MultiLineStringType

```
<complexType name="MultiLineStringType">
    <complexContent>
        <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
                <element ref="gml:lineStringMember" minOccurs="0" maxOccurs="unbounded" />
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

A MultiLineString is defined by one or more LineStrings, referenced through lineStringMember elements. Deprecated with GML version 3.0. Use MultiCurveType instead.

Element MultiLineString

```
<element name="MultiLineString" type="gml:MultiLineStringType" substitutionGroup="gml:_GeometricAggregate" />
```

Deprecated with GML 3.0 and included for backwards compatibility with GML 2. Use the "MultiCurve" element instead.

Complex Type MultiLineStringPropertyType

```
<complexType name="MultiLineStringPropertyType">
    <sequence>
        <element ref="gml:MultiLineString" minOccurs="0" />
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```


This type is deprecated with GML 3 and shall not be used. It is included for backwards compatibility with GML 2. Use MultiCurvePropertyType instead.

A property that has a collection of line strings 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.

7.5.6.5 2-dimensional geometric aggregates

Element surfaceMember

```
<element name="surfaceMember" type="gml:SurfacePropertyType" />
```

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".

Element surfaceMembers

```
<element name="surfaceMembers" type="gml:SurfaceArrayPropertyType" />
```

This property element contains a list of surfaces. The order of the elements is significant and shall be preserved when processing the array.

Complex Type MultiSurfaceType

```
<complexType name="MultiSurfaceType">
    <complexContent>
        <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
                <element ref="gml:surfaceMember" minOccurs="0" maxOccurs="unbounded" />
                <element ref="gml:surfaceMembers" minOccurs="0" />
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

A MultiSurface is defined by one or more Surfaces, referenced through surfaceMember elements.

The members of the geometric aggregate can be specified either using the "standard" property or the array property style. It is also valid to use both the "standard" and the array property style in the same collection.

NOTE: Array properties cannot reference remote geometry elements via XLinks.

Element MultiSurface

```
<element name="MultiSurface" type="gml:MultiSurfaceType" substitutionGroup="gml:_GeometricAggregate" />
```

Complex Type MultiSurfacePropertyType

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

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.

Element multiSurfaceProperty

```
<element name="multiSurfaceProperty" type="gml:MultiSurfacePropertyType" />
```

This property element either references a surface aggregate via the XLink-attributes or contains the "multi surface" element. multiSurfaceProperty is the predefined property, which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for MultiSurface.

Element polygonMember

```
<element name="polygonMember" type="gml:PolygonPropertyType" />
```

Deprecated with GML 3.0 and included only for backwards compatibility with GML 2.0. Use "surfaceMember" instead.

This property element either references a polygon via the XLink-attributes or contains the polygon element.

Complex Type MultiPolygonType

```
<complexType name="MultiPolygonType">
    <complexContent>
        <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
                <element ref="gml:polygonMember" minOccurs="0" maxOccurs="unbounded" />
            </sequence>
```

```
        </extension>
    </complexContent>
</complexType>
```

A MultiPolygon is defined by one or more Polygons, referenced through polygonMember elements. Deprecated with GML version 3.0. Use MultiSurfaceType instead.

Element MultiPolygon

```
<element name="MultiPolygon" type="gml:MultiPolygonType" substitutionGroup="gml:_GeometricAggregate" />
```

Deprecated with GML 3.0 and included for backwards compatibility with GML 2. Use the "MultiSurface" element instead.

Complex Type MultiPolygonPropertyType

```
<complexType name="MultiPolygonPropertyType">
    <sequence>
        <element ref="gml:MultiPolygon"  minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```

This type is deprecated with GML 3 and shall not be used. It is included for backwards compatibility with GML 2. Use MultiSurfacePropertyType instead.

A property that has a collection of polygons 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.

7.5.6.6 3-dimensional geometric aggregates

Element solidMember

```
<element name="solidMember" type="gml:SolidPropertyType" />
```

This property element either references a solid via the XLink-attributes or contains the solid element. A solid element is any element, which is substitutable for "_Solid".

Element solidMembers

```
<element name="solidMembers" type="gml:SolidArrayPropertyType" />
```

This property element contains a list of solids. The order of the elements is significant and shall be preserved when processing the array.

Complex Type MultiSolidType

```
<complexType name="MultiSolidType">
    <complexContent>
        <extension base="gml:AbstractGeometricAggregateType">
            <sequence>
                <element ref="gml:solidMember" minOccurs="0" maxOccurs="unbounded" />
                <element ref="gml:solidMembers" minOccurs="0" />
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

A MultiSolid is defined by one or more Solids, referenced through solidMember elements.

The members of the geometric aggregate can be specified either using the "standard" property or the array property style. It is also valid to use both the "standard" and the array property style in the same collection.

NOTE: Array properties cannot reference remote geometry elements via XLinks.

Element MultiSolid

```
<element name="MultiSolid" type="gml:MultiSolidType" substitutionGroup="gml:_GeometricAggregate" />
```

Complex Type MultiSolidPropertyType

```
<complexType name="MultiSolidPropertyType">
    <sequence>
        <element ref="gml:MultiSolid"  minOccurs="0" />
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>
```

A property that has a collection of solids 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.

Element multiSolidProperty

```
<element name="multiSolidProperty" type="gml:MultiSolidPropertyType" />
```

This property element either references a solid aggregate via the XLink-attributes or contains the "multi solid" element. multiSolidProperty is the predefined property, which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for MultiSolid.

7.5.6.7 Aliases

As a conveniece for application schema developers and to support backwards compatibility with GML 2, the following aliases have been defined for properties that take a geometric aggregate as a value:

```
<element name="multiLocation" type="gml:MultiPointPropertyType" substitutionGroup="gml:multiPointProperty">
<element name="multiCenterOf" type="gml:MultiPointPropertyType" substitutionGroup="gml:multiPointProperty"/>
<element name="multiPosition" type="gml:MultiPointPropertyType" substitutionGroup="gml:multiPointProperty"/>
<element name="multiCenterLineOf" type="gml:MultiCurvePropertyType" substitutionGroup="gml:multiCurveProperty"/>
<element name="multiEdgeOf" type="gml:MultiCurvePropertyType" substitutionGroup="gml:multiCurveProperty"/>
<element name="multiCoverage" type="gml:MultiSurfacePropertyType" substitutionGroup="gml:multiSurfaceProperty"/>
<element name="multiExtentOf" type="gml:MultiSurfacePropertyType" substitutionGroup="gml:multiSurfaceProperty"/>
```

7.5.7 Geometric Properties

In general the definition of feature properties lies in the domain of application schemas. However, since the OGC abstract specification defines a limited set of basic geometry types, GML defines a set of geometric property elements to associate instances of these geometry types with features.

The GML Feature schema also provides descriptive names for the geometry properties, encoded as common English language terms. Overall, there are three levels of naming geometry properties in GML:

- **Formal names** that denote geometry properties in a manner based on the type of geometry allowed as a property value. These are names based on the name of the geometry type with a suffix “Property”.

Formal name	Geometry type
pointProperty	Point
curveProperty	LineString Curve OrientableCurve CompositeCurve

surfaceProperty	Polygon Surface OrientableSurface CompositeSurface
solidProperty	Solid CompositeSolid
geometryProperty	AbstractGeometry
multiPointProperty	MultiPoint
multiCurveProperty	MultiCurve
multiSurfaceProperty	MultiSurface
multiSolidProperty	MultiSolid
multiGeometryProperty	MultiGeometry
pointArrayProperty	Point(s)
curveArrayProperty	LineString(s) Curve(s) OrientableCurve(s) CompositeCurve(s)
surfaceArrayProperty	Polygon(s) Surface(s) OrientableSurface(s) CompositeSurface(s)
solidArrayProperty	Solid(s) CompositeSolid(s)

Table 7.5-1 – Pre-defined Formal Geometric Properities

The precise semantics of these geometry properties (e.g. "What does pointProperty of an object mean?") is not specified.

- **Descriptive names** that provide a set of standardized synonyms or aliases for the formal names; these allow use of a more user-friendly set of terms. See 7.4.1.9 and 7.5.6.7.

- **Application-specific names** chosen by users and defined in application schemas based on GML.

There are no inherent restrictions in the type of geometry property a feature type may have. For example, a `RadioTower` feature type could have a *location* that returns a Point geometry to identify its location, and have another geometry property called *extentOf* that returns a Polygon geometry describing its physical structure, and have yet a third geometry property called *serviceArea* that returns a Polygon geometry describing the area in which its transmissions can be received reliably.

7.5.8 User-defined Geometry Types and Geometry Property Types

7.5.8.1 User-defined Geometry Types

Authors of application schemas may create their own geometry types if GML lacks the desired construct. To do this, authors must ensure that these concrete geometry and geometry collection types are subtyped (either directly or indirectly) from the corresponding GML type: `AbstractGeometryType`.

EXAMPLE: The following complex type definition in an application schema extends the PointType of GML and adds a bearing (e.g. for the orientation of a symbol in portrayal).

```
<xsd:complexType name="PointWithBearingType">
    <xsd:complexContent>
        <xsd:extension base=" gml:PointType">
            <xsd:sequence>
                <xsd:element name="bearing" type="gml:AngleType"/>
            </xsd:sequence>
        </xsd:extension>
    </xsd:complexContent>
</xsd:complexType>
```

Any user-defined geometry subtypes shall inherit the elements and attributes of the base GML geometry types without restriction, but may extend these base types to meet application requirements, such as providing a finer degree of interoperability with legacy systems and data sets.

7.5.8.2 User-defined Geometry Property Types

Furthermore, authors of application schemas may create their own geometry property types that encapsulate geometry types they have defined according to clause 7.5.8.1. They must ensure that these properties follow the pattern used by `gml:GeometryPropertyType.` For standard properties and `gml:GeometryArrayPropertyType` for array properties. The target type must be a bonafide geometry construct.

A geometry property type may be a restriction of GeometryPropertyType, but this is not a requirement. Nevertheless, every geometry property shall follow the pattern of this type. It is allowed to support the choice between a by-value or a by-reference semantic or to restrict the use to either by-value (prohibit the use of the Xlink attributes) or by-reference (prohibit the containment of the geometry in the feature).

A geometry array property type may be a restriction of GeometryArrayPropertyType, but this is not a requirement. Nevertheless, every geometry property shall follow the pattern of this type. All geometry elements in the array are contained in the feature, only by-value semantics is supported by array properties.

EXAMPLE: The following complex type definitions in an application schema define a “standard” property type for an user-defined geometry type and an array property type for the same geometry type.

```
<complexType name="MyGeometryPropertyType">
    <sequence>
        <element ref="foo:PointWithBearingType"  minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup" />
</complexType>

<complexType name="MyGeometryArrayPropertyType">
    <sequence>
        <element ref="foo:PointWithBearingType" minOccurs="0" maxOccurs="unbounded" />
    </sequence>
</complexType>
```

7.6 Coordinate Reference Systems

GML requires a coordinate reference system (CRS) to be referenced whenever location coordinate information is given. This CRS provides the meaning for location coordinates. The referencing is generally given using the srsName attribute which is provided by gml:AbstractGeometryType which is the basis for the content models for all GML geometry elements:

```
<gml:Point srsName=”utm27n”>. . .
```

where ‘utm27n’ will lead to a full definition of some CRS.

7.6.1 Documentation of CRS schemas

The CRS portion of GML is intended to document the meaning of the CRS reference, which includes transformations and conversions between coordinate reference systems.

The relevant documents for understanding the CRS schemas are:

1. OGC Abstract Specification Topic 2 “Spatial Referencing by Coordinates”, document 02-102
2. Future OGC Recommendation Paper "Recommended XML encoding of coordinate reference system definitions", incomplete draft in document 02-036r6

There is a set of six XML schema documents for encoding CRS definitions. These six files include documentation text, and are named coordinateReferenceSystems.xsd, datums.xsd, coordinateSystems.xsd, coordinateOperations.xsd, dataQuality.xsd, and referenceSystems.xsd.

> NOTE: The XML schema documents for encoding CRS definitions that are included in this specification are not yet final, and some future changes are planned. For the convenience of GML3 users they are included in informative Annex D. The revised version of these schemas will be documented in successors to document 02-036r6.

7.6.2 Meaning of the CRS

The primary object to be referenced is a CRS. The method of referencing is specified in other clauses of this document which describe the various objects and their properties. The CRS instance will give information about the coordinates, including the order of the coordinates, the unit of measure that goes with each coordinate, and the physical meaning in terms of its attachment to the earth (that is, the datum). Also, it is sometimes necessary to express a conversion or transformation to another CRS. Definitions of coordinate conversions and transformations are also encoded using the CRS schemas.

Most CRSs used with GML will be references to standard or well-known coordinate reference systems. The XML schemas allow information for a set of standard CRSs to be stored in a dictionary. It is then necessary only to reference the proper CRS in such a dictionary, which will contain all the information needed to understand the coordinates. It is also possible to convert coordinate data into another CRS, using a coordinate transformation service. There is already an OGC Implementation Specification for such a service, document 01-009, and available implementations of this service. Using this service, a set of coordinates in one CRS can be converted to another CRS.

Another possible use is when coordinates are not in a standard coordinate reference system. In this case, it is necessary for the XML document to reference a non-standard CRS, which may be defined within the same document. The definition may be as simple as giving the meaning of the axes (order and units of measure) and the datum, or it may include a transformation or conversion to a standard CRS. Thus, the CRS schemas allow a XML document to specify the meaning of its coordinates, and their relationship to a standard, earth-related coordinate system.

7.7 Topology

7.7.1 Introduction

Topology is the branch of mathematics describing the properties of objects which are invariant under continuous deformation. For example, a circle is topologically equivalent to an ellipse because one can be transformed into the other by stretching. In geographic modelling, the foremost use of topology is in accelerating computational geometry. The constructs of topology allow characterisation of the spatial relationships between objects using simple combinatorial or algebraic algorithms. Topology, realised by the appropriate geometry, also allows a compact and unambiguous mechanism for expressing shared geometry among geographic features.

7.7.2 Topology Model

7.7.2.1 Overview

The conceptual model underlying the representation of topology in GML is that of Topic 1 of the OGC Abstract Specification (ISO DIS 19107). The model describes the correspondence of topological and geometric relationships up to 3 dimensions – volume topology. Relevant parts of the model are summarised here for convenience.

7.7.2.2 Topology Objects

There are four instantiable classes of primitive topology objects, one for each dimension up to 3D. In addition, topological complexes are supported, too. See figure 7.7-1.

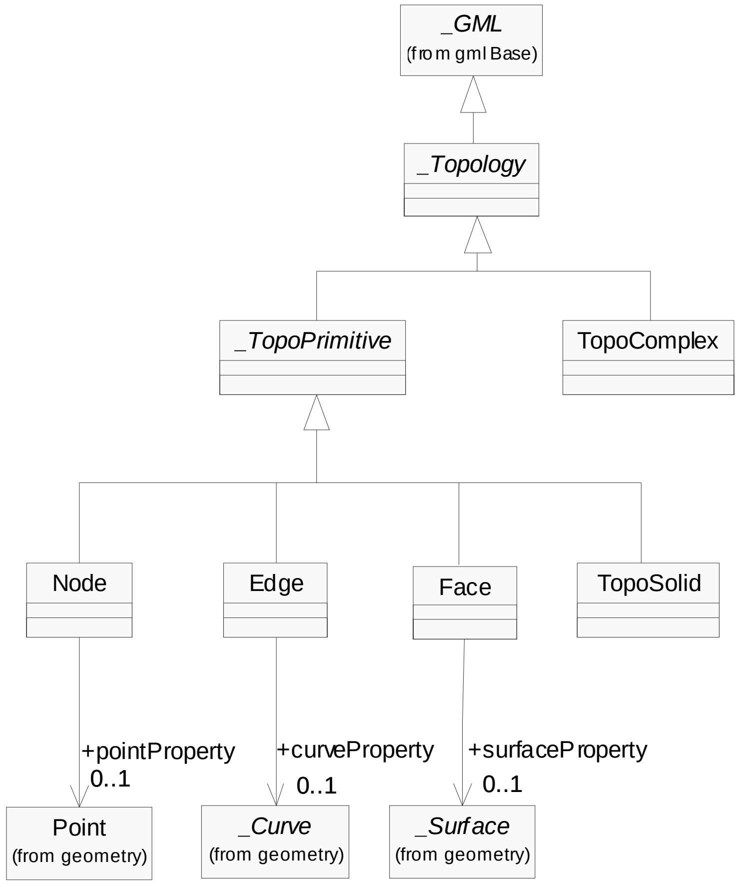


Figure 7.7-1 – Topology Objects

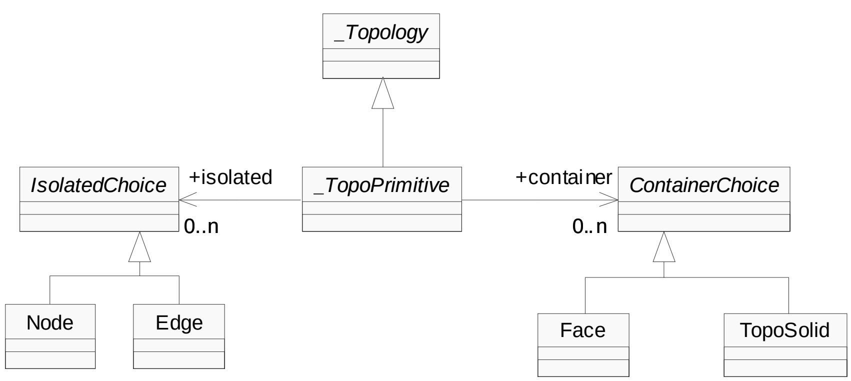


Figure 7.7-2 – TopoPrimitive

7.7.2.3 Topological Boundaries

Figures 7.7-3 and 7.7-4 illustrate the topological boundary and coboundary relationships inherent among the topology primitives. There is strong symmetry in the relationships between topology primitives of adjacent dimensions. Topology primitives are bounded by directed primitives of one lower dimension. The coboundary of each topology primitive is formed from directed topology primitives of one higher dimension.

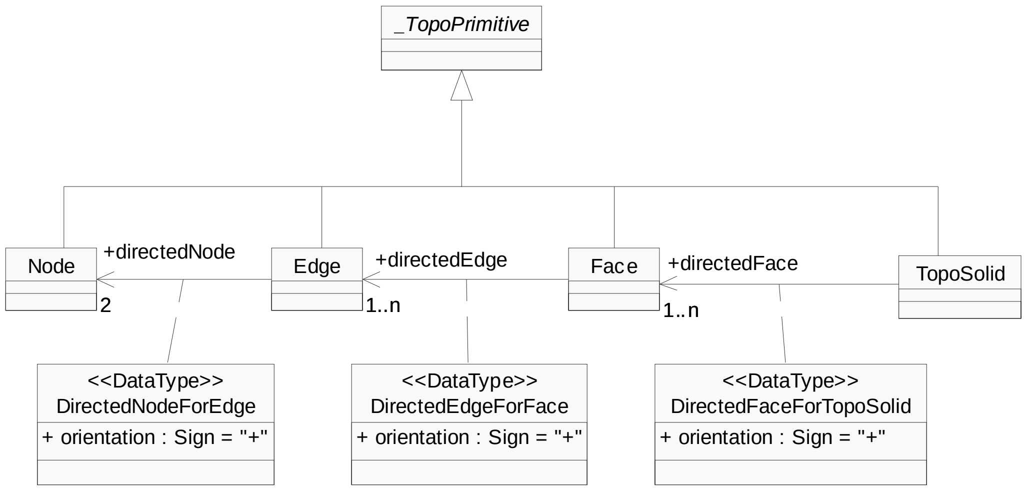


Figure 7.7-3 – Boundary relationships

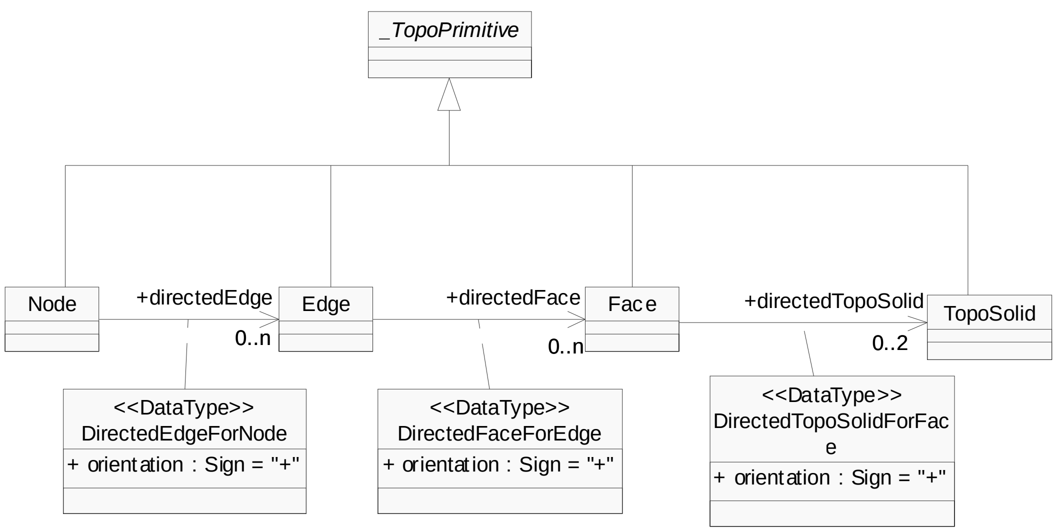


Figure 7.7-4 – Coboundary relationships

7.7.2.4 Topological Expressions

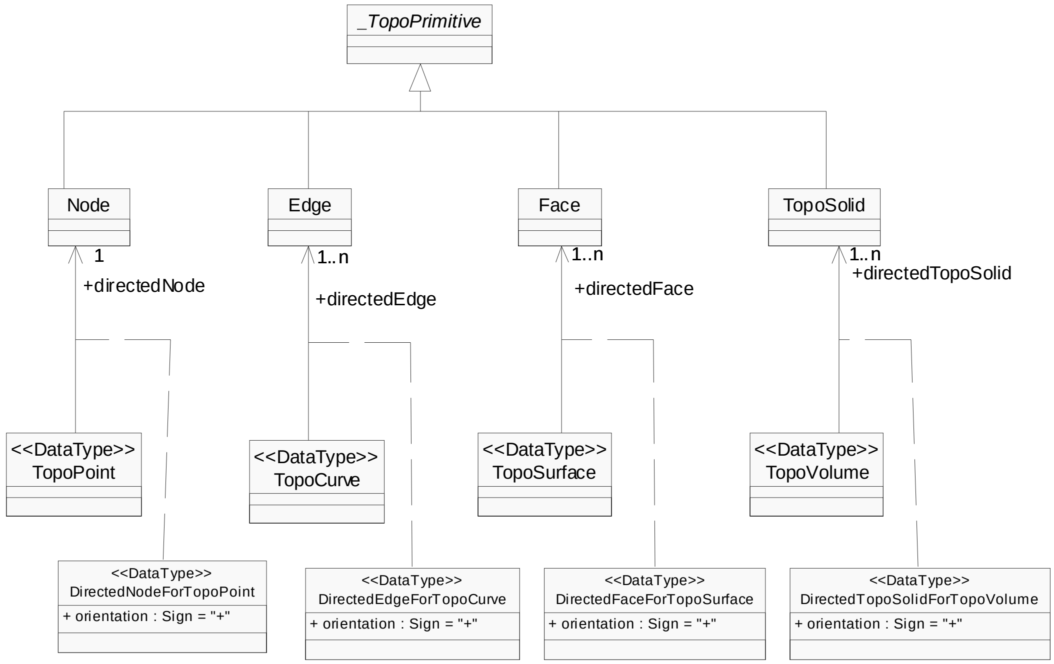


Figure 7.7-5 – Topological Expressions

7.7.2.5 Topological Complex

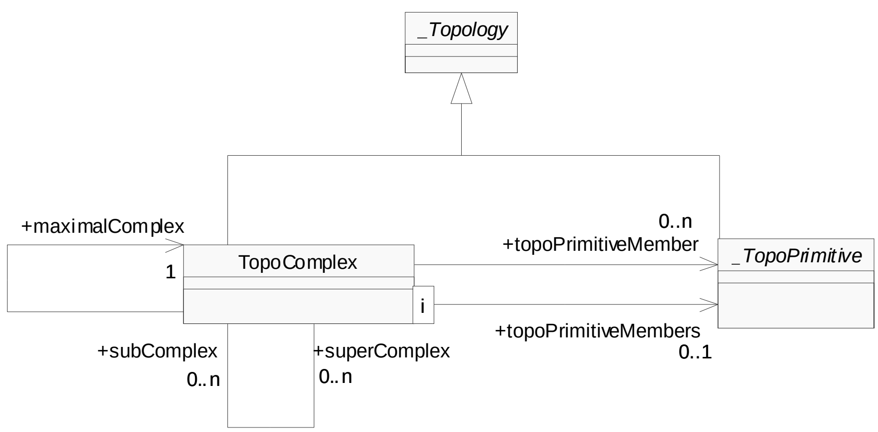


Figure 7.7-6 – TopoComplex

7.7.3 Types & Elements

7.7.3.1 Schema Includes

```
<include schemaLocation="geometryComplexes.xsd"/>
```


The topology schema includes definitions from the geometry schema yielding the ability to describe primitives and complexes with a geometric realisation.

7.7.3.2 AbstractTopology

```
<complexType name="AbstractTopologyType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType"/>
  </complexContent>
</complexType>

<element name="_Topology" type="gml:AbstractTopologyType" abstract="true"
substitutionGroup="gml:_Object"/>
```

This abstract type supplies the root or base type for all topological elements including primitives and complexes. It inherits AbstractGMLType and hence can be identified using the gml:id attribute.

7.7.3.3 AbstractTopoPrimitive

```
<complexType name="AbstractTopoPrimitiveType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractTopologyType">
      <sequence>
        <element ref="gml:isolated" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:container" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="_TopoPrimitive" type="gml:AbstractTopoPrimitiveType" abstract="true"
substitutionGroup="gml:_Topology"/>
```

This abstract type acts as the base type for all topological primitive elements. Topology primitives are the atomic (smallest possible) units of a topology complex. Each topology primitive may contain references to other topology primitives of codimension 2 or more. So faces may isolate nodes and TopoSolids may isolate nodes and edges. Conversely, nodes may have faces as containers and nodes and edges may have TopoSolids as containers.

7.7.3.4 Node

```
<complexType name="NodeType">
  <complexContent>
    <extension base="gml:AbstractTopoPrimitiveType">
      <sequence>
        <element ref="gml:directedEdge" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:pointProperty" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="Node" type="gml:NodeType" substitutionGroup="gml:_TopoPrimitive"/>
```

The Node type and global element represent the 0-dimensional primitive expressing point coincidence. The topological boundary of a node is empty and hence requires no representation. The optional coboundary of a node is a set of directed edges which are incident on this node.

NOTE: In a 2D complex, this set may be ordered as a clockwise circular sequence. In a 3D complex, the order of this set is arbitrary.

Edges emanating from this node appear in the node coboundary with a negative orientation.

A node may optionally be realised by a 0-dimensional (point) geometric primitive.

7.7.3.5 DirectedNode

```
<complexType name="DirectedNodePropertyType">
  <choice>
    <element ref="gml:Node" minOccurs="0"/>
  </choice>
  <attribute name="orientation" type="gml:SignType" default="+"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="directedNode" type="gml:DirectedNodePropertyType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:directedNode">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
```

The role of the DirectedNode type and element is in the boundary of topology edges and in the support of topological point features via the TopoPoint expression, see below. The orientation attribute of type gml:SignType expresses the sense in which the included node is used e.g. start (“-“) or end (“+”) node.

7.7.3.6 Edge

```
<complexType name="EdgeType">
  <complexContent>
    <extension base="gml:AbstractTopoPrimitiveType">
      <sequence>
        <element ref="gml:directedNode" minOccurs="2" maxOccurs="2"/>
        <element ref="gml:directedFace" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:curveProperty" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="Edge" type="gml:EdgeType" substitutionGroup="gml:_TopoPrimitive"/>
```

The Edge type and global element represent the 1-dimensional primitive expressing linear coincidence. The topological boundary of an Edge consists of a negatively directed start Node and a positively directed end Node. The optional coboundary of an edge is a circular sequence of directed faces which are incident on this edge in document order. Faces which use a particular boundary edge in its positive orientation appear with positive orientation on the coboundary of the same edge. In the 2D case, the orientation of the face on the left of the edge is "+"; the orientation of the face on the right on its right is "-". An edge may optionally be realised by a 1-dimensional (curve) geometric primitive.

7.7.3.7 DirectedEdge

```
<complexType name="DirectedEdgePropertyType">
  <choice>
    <element ref="gml:Edge" minOccurs="0"/>
  </choice>
  <attribute name="orientation" type="gml:SignType" default="+"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="directedEdge" type="gml:DirectedEdgePropertyType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:directedEdge">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
```

The role of the DirectedEdge type and element is in the boundary of topology faces, in the coBoundary of topology nodes and in the support of topological line features via the TopoCurve expression, see below. The orientation attribute of type gml:SignType expresses the sense in which the included edge is used e.g. forward or reverse.

7.7.3.8 Face

```
<complexType name="FaceType">
  <complexContent>
    <extension base="gml:AbstractTopoPrimitiveType">
      <sequence>
        <element ref="gml:directedEdge" maxOccurs="unbounded"/>
        <element ref="gml:directedTopoSolid" minOccurs="0" maxOccurs="2"/>
        <element ref="gml:surfaceProperty" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

```
<element name="Face" type="gml:FaceType" substitutionGroup="gml:_TopoPrimitive"/>
```

The Face type and global element represent the 2-dimensional topology primitive expressing surface overlap. The topological boundary of a face consists of a set of directed edges. Note that all edges associated with the face, including dangling edges, appear in the boundary. A dangling edge has the same face on both sides. Consequently, a dangling edge has two different nodes in its boundary. A dangling edge may share zero, one or two bounding nodes with other edges in the boundary of a face. Two directedEdge elements with opposite orientations reference each dangling edge in the boundary of a face. The non-dangling edges in the boundary of a face comprise one or more topological rings. Each such ring consists of directedEdges connected in a cycle, and is oriented with the face on its left. The optional coboundary of a face is a pair of directed solids which are bounded by this face. If present, there is precisely one positively directed and one negatively directed solid in the coboundary of every face. The positively directed solid corresponds to the solid which lies in the direction of the positively directed normal to the face in any geometric realisation. A face may optionally be realised by a 2-dimensional (surface) geometric primitive.

7.7.3.9 DirectedFace

```
<complexType name="DirectedFacePropertyType">
  <choice>
    <element ref="gml:Face" minOccurs="0"/>
  </choice>
  <attribute name="orientation" type="gml:SignType" default="+"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="directedFace" type="gml:DirectedFacePropertyType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:directedFace">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
```

The role of the DirectedFace type and element is in the boundary of topology solids, in the coBoundary of topology edges and in the support of surface features via the TopoSurface expression, see below. The orientation attribute of type gml:SignType expresses the sense in which the included face is used e.g. inward or outward with respect to the surface normal in any geometric realisation.

7.7.3.10 TopoSolid

```
<complexType name="TopoSolidType">
```

```
  <complexContent>
    <extension base="gml:AbstractTopoPrimitiveType">
      <sequence>
        <element ref="gml:directedFace" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<element name="TopoSolid" type="gml:TopoSolidType" substitutionGroup="gml:_TopoPrimitive"/>
```

The TopoSolid type and global element represent the 3-dimensional topology primitive expressing Volume interclause. The topological boundary of a TopoSolid consists of a set of directed faces. Note that all faces associated with the TopoSolid, including dangling faces, appear in the boundary. A dangling face has the same solid on both sides. Two directedFace elements with opposite orientations reference each dangling face in the boundary of a topological solid. The coboundary of a TopoSolid is empty and hence requires no representation.

7.7.3.11 DirectedTopoSolid

```
<complexType name="DirectedTopoSolidPropertyType">
  <choice>
    <element ref="gml:TopoSolid"/>
  </choice>
  <attribute name="orientation" type="gml:SignType" default="+"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="directedTopoSolid" type="gml:DirectedTopoSolidPropertyType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:directedTopoSolid">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
```

The role of the DirectedSolid type and element is in the coBoundary of topology faces and in the support of 3D volume features via the TopoVolume expression, see below. The orientation attribute of type gml:SignType expresses the sense in which the included solid appears in the face coboundary. Note that in the context of a TopoVolume the orientation attribute has no meaning in 3D.

7.7.3.12 IsolatedProperty

```
<complexType name="IsolatedPropertyType">
  <choice minOccurs="0">
```



```
    <element ref="gml:Node"/>
    <element ref="gml:Edge"/>
  </choice>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="isolated" type="gml:IsolatedPropertyType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:isolated">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
```

All of the adjacency relationships between topology primitives whose dimensions differ by +/- 1 are described using boundary and coboundary through directed topology primitive properties as described for each primitive. This includes instances where a dangling edge has the same face on both sides or a dangling face has the same solid on both sides. These primitives will appear twice in the boundary of the relevant face or solid with alternate sign. Primitives which are enclosed by another primitive of at least codimension 2 however, with no intermediate dimension primitive, are truly isolated. For faces this corresponds to nodes with no relationship to a bounding edge of the face. For solids, it corresponds to nodes or edges which are not referred to by the boundary faces of the solid.

Note a node may be isolated in a face and yet appear in the boundary of an edge which does *not* form part of the boundary of the face. In 3D this corresponds geometrically to a curve whose endpoint meets the interior of a surface.

7.7.3.13 ContainerProperty

```
<complexType name="ContainerPropertyType">
  <choice minOccurs="0">
    <element ref="gml:Face"/>
    <element ref="gml:TopoSolid"/>
  </choice>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="container" type="gml:ContainerPropertyType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:containerProperty">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
```

The container type and element optionally contain the reciprocal relationships for the isolated property described above.

7.7.3.14 TopoPoint

```
<complexType name="TopoPointType">
    <sequence>
        <element ref="gml:directedNode"/>
    </sequence>
</complexType>

<element name="TopoPoint" type="gml:TopoPointType"/>
```

The intended use of TopoPoint is to appear within a point feature to express the structural and possibly geometric relationships of this point to other features via shared node definitions. Note the orientation assigned to the directedNode has no meaning in this context. It is preserved for symmetry with the types and elements which follow.

7.7.3.15 TopoPointProperty

```
<complexType name="TopoPointPropertyType">
    <sequence>
        <element ref="gml:TopoPoint"/>
    </sequence>
</complexType>

<element name="topoPointProperty" type="gml:TopoPointPropertyType"/>
```

The topoPointProperty element can be used as a property of features to express their relationship to the referenced topology node.

7.7.3.16 TopoCurve

```
<complexType name="TopoCurveType">
    <sequence>
        <element ref="gml:directedEdge" maxOccurs="unbounded"/>
    </sequence>
</complexType>

<element name="TopoCurve" type="gml:TopoCurveType"/>
```

The TopoCurve type and element represent a homogeneous topological expression, a list of directed edges, which if realised are isomorphic to a geometric curve primitive. The intended use of TopoCurve is to appear within a line feature instance to express the structural and geometric relationships of this line to other features via the shared edge definitions.

7.7.3.17 TopoCurveProperty

```
<complexType name="TopoCurvePropertyType">
    <sequence>
        <element ref="gml:TopoCurve"/>
    </sequence>
</complexType>

<element name="topoCurveProperty" type="gml:TopoCurvePropertyType"/>
```

The topoCurveProperty element can be used as a property of features to express their relationship to the referenced topology edges.

7.7.3.18 TopoSurface

```
<complexType name="TopoSurfaceType">
    <sequence>
        <element ref="gml:directedFace" maxOccurs="unbounded"/>
    </sequence>
</complexType>

<element name="TopoSurface" type="gml:TopoSurfaceType"/>
```

The TopoSurface type and element represent a homogeneous topological expression, a set of directed faces, which if realised are isomorphic to a geometric surface primitive. The intended use of TopoSurface is to appear within a surface feature instance to express the structural and possibly geometric relationships of this surface to other features via the shared face definitions.

7.7.3.19 TopoSurfaceProperty

```
<complexType name="TopoSurfacePropertyType">
    <sequence>
        <element ref="gml:TopoSurface"/>
    </sequence>
</complexType>

<element name="topoSurfaceProperty" type="gml:TopoSurfacePropertyType"/>
```

The topoSurfaceProperty element can be used as a property of features to express their relationship to the referenced topology faces.

7.7.3.20 TopoVolume

```
<complexType name="TopoVolumeType">
    <sequence>
        <element ref="gml:directedTopoSolid" maxOccurs="unbounded"/>
    </sequence>
</complexType>

<element name="TopoVolume" type="gml:TopoVolumeType"/>
```

The TopoVolume type and element represent a homogeneous topological expression, a set of directed TopoSolids, which if realised are isomorphic to a geometric solid primitive. The intended use of TopoVolume is to appear within a 3D solid feature instance to express the structural and geometric relationships of this solid to other features via the shared TopoSolid definitions. Note the orientation assigned to the directedSolid has no meaning in three dimensions. It is preserved for symmetry with the preceding types and elements.

7.7.3.21 TopoVolumeProperty

```
<complexType name="TopoVolumePropertyType">
    <sequence>
        <element ref="gml:TopoVolume"/>
    </sequence>
</complexType>

<element name="topoVolumeProperty" type="gml:TopoVolumePropertyType"/>
```

The topoVolumeProperty element can be used as a property of features to express their relationship to the referenced topology Volume.

7.7.3.22 TopoComplex

```
  <complexType name="TopoComplexType">
   <annotation>
    <documentation>This type represents a TP_Complex capable of holding topological
primitives.</documentation>
   </annotation>
   <complexContent>
    <extension base="gml:AbstractTopologyType">
     <sequence>
      <element ref="gml:maximalComplex"/>
      <element ref="gml:superComplex" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="gml:subComplex" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="gml:topoPrimitiveMember" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="gml:topoPrimitiveMembers" minOccurs="0"/>
```



```
      </sequence>
      <attribute name="isMaximal" type="boolean" default="false"/>
    </extension>
  </complexContent>
</complexType>

<element name="TopoComplex" type="gml:TopoComplexType" substitutionGroup="gml:_Topology"/>
```

This type and element provide encoding for a topology complex comprising multiple topology primitive members. In addition to primitives, each complex holds a reference to a unique maximal complex (the complex which has no supercomplex) and optionally to some number of sub- or super- complexes. A topology complex contains its primitive and sub-complex members, and is contained by its super-complex(es). The primitive and sub-complex members of a topological complex have dimensionality less than or equal to the dimensionality of the topology complex. There is one and only one maximal complex per topological manifold.

7.7.3.23 Maximal, sub- & super- complexes

```
<complexType name="TopoComplexMemberType">
  <sequence>
    <element ref="gml:TopoComplex" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="subComplex" type="gml:TopoComplexMemberType" substitutionGroup="gml:_property">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:subComplex">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>

<element name="superComplex" type="gml:TopoComplexMemberType"
substitutionGroup="gml:_property">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:superComplex">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>

<element name="maximalComplex" type="gml:TopoComplexMemberType"
substitutionGroup="gml:_property">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:subComplex">
```



```
        <sch:extends rule="hrefOrContent"/>
      </sch:rule>
    </sch:pattern>
  </appinfo>
  <documentation>Need schamatron test here that isMaximal attribute value is true</documentation>
 </annotation>
</element>
```

These elements and type provide encoding for relationships between topology complexes as described for the TopoComplexType.

7.7.3.24 TopoPrimitiveMember

```
<complexType name="topoPrimitiveMemberType">
 <sequence>
  <element ref="gml:_TopoPrimitive" minOccurs="0"/>
 </sequence>
 <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="topoPrimitiveMember" type="gml:topoPrimitiveMemberType">
 <annotation>
  <appinfo>
   <sch:pattern>
    <sch:rule context="gml:topoPrimitiveMember">
     <sch:extends rule="hrefOrContent"/>
    </sch:rule>
   </sch:pattern>
  </appinfo>
 </annotation>
</element>
```

This type and element encode the relationship between a topology complex and a single topology primitive.

7.7.3.25 TopoPrimitiveArrayAssociation

```
<complexType name="TopoPrimitiveArrayAssociationType">
 <complexContent>
  <restriction base="gml:ArrayAssociationType">
   <sequence>
    <element ref="gml:_TopoPrimitive" minOccurs="0" maxOccurs="unbounded"/>
   </sequence>

  </restriction>
 </complexContent>
</complexType>

<element name="topoPrimitiveMembers" type="gml:TopoPrimitiveArrayAssociationType"
substitutionGroup="gml:members">
 <annotation>
  <appinfo>
   <sch:pattern>
    <sch:rule context="gml:topoPrimitiveMember">
     <sch:extends rule="hrefOrContent"/>
    </sch:rule>
   </sch:pattern>
  </appinfo>
```

```
  </annotation>
 </element>
```

This type and element encode the relationship between a topology complex and an arbitrary number of topology primitives. Note that because of the array style of encoding this type requires that the encodings for the topology primitives be in-line rather than remote properties.

7.7.3.26 TopoComplexProperty

```
<complexType name="TopoComplexMemberType">
  <sequence>
    <element ref="gml:TopoComplex" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>

<element name="topoComplexProperty" type="gml:TopoComplexMemberType" "/>
```

This type and element encode the relationship between a GML object such as a feature collection and a topological complex. They allow a feature collection to contain or reference a topological complex that contains topologies referenced by members of the feature collection.

7.8 Temporal information and dynamic features

7.8.1 Overview

The GML temporal schemas extend the core elements of GML to include elements for describing the temporal characteristics of geographic data; their purpose is to describe and define the constructs that comprise a language—provide a means of describing the history of a dynamic feature basic temporal schema. These constructs—expressed using the XML Schema definition of the lifecycle of a feature collection. The underlying spatiotemporal model strives to accommodate both feature-level and attribute-level time stamping; basic support for tracking moving objects is also included.

The conceptual model underlying the representation of temporal objects in GML constitutes a profile of the conceptual schema described in ISO/DIS 19108 [i]; however, topological types and temporal feature relationships are not included in the current versions of the temporal schemas.

Two other ISO standards are relevant to describing temporal objects: ISO 8601 describes encodings for time instants and time periods, as text strings with particular structure and punctuation; ISO 11404 provides a detailed description of time intervals as part of a general discussion of language independent datatypes.

The temporal schemas consist of two interrelated schemas: the main schema provides basic elements such as primitive types for representing temporal instants and periods; a more specialized schema defines types used to represent dynamic features. Instances of temporal geometric types are used as values for the temporal properties of geographic features. Both schemas are listed in Annex C.

7.8.2 Basic temporal types

The basic temporal types and elements are described in the schema listed in Annex C. The main temporal schema is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:temporal:v3.0c6

A UML representation of the schema provides a visual summary of the principal elements. Each class, attribute or rolename in Figure 7.8.2 corresponds to an element declared in the schema.

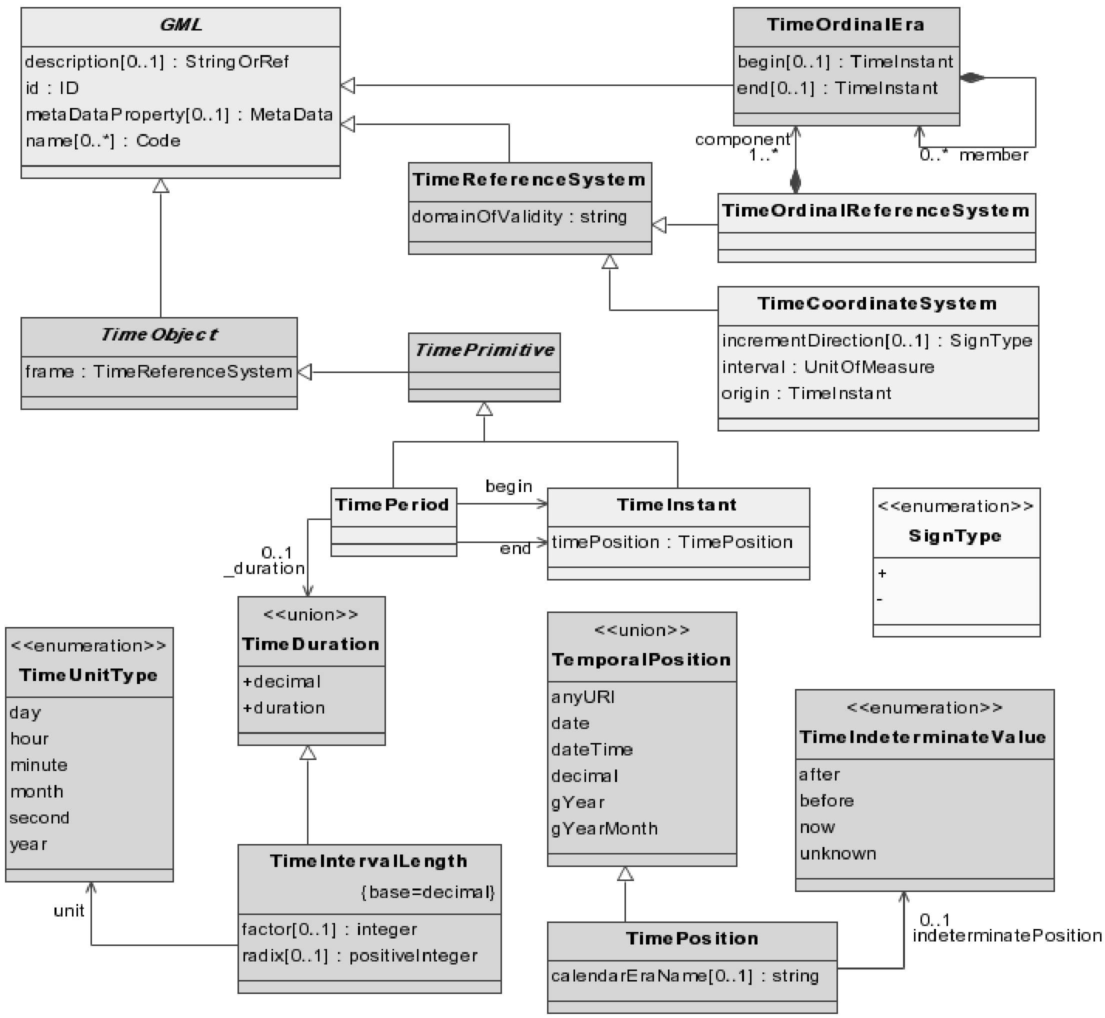


Figure 7.8.2 UML representation of the main temporal schema

7.8.2.1 gml:_TimeObject

Time is measured on two types of scales: interval and ordinal. An interval scale offers a basis for measuring duration; an ordinal scale provides information only about relative position in time (e.g. a stratigraphic sequence or the geological time scale). gml:_TimeObject is an abstract element which acts as the head of a substitution group for all temporal primitives and complexes; it uses gml:**AbstractTimeType** defined in the schema as follows:

```
 <element name="_TimeObject" type="gml:AbstractTimeType" abstract="true"
substitutionGroup="gml:_GML"/>

 <complexType name="AbstractTimeType" abstract="true">
  <complexContent>
   <extension base="gml:AbstractGMLType">
    <attribute name="frame" type="anyURI" use="optional" default="#ISO-8601"/>
   </extension>
  </complexContent>
 </complexType>
```

All TimeObjects must be associated with a temporal reference system through the **frame** attribute that provides a URI reference that identifies a description of the reference system. Following ISO 8601, the Gregorian calendar with UTC is the default reference system, but others may also be used. The GPS calendar and the Julian calendar are alternative reference systems in common use.

7.8.2.2 gml:_TimePrimitive

The abstract element TimePrimitive is used to represent temporal geometric objects. It is defined in the schema as follows:

```
 <element name="_TimePrimitive" type="gml:TimePrimitiveType" abstract="true"
substitutionGroup="gml:_TimeObject"/>

 <complexType name="TimePrimitiveType" abstract="true">
  <complexContent>
   <extension base="gml:AbstractTimeType"/>
  </complexContent>
 </complexType>
```

> NOTE: gml:_TimePrimitive is derived in a trivial manner from gml:_TimeObject. This hierarchy is primarily provided in anticipation of the addition of temporal complexes to the model and schema, which would also be substitutable for gml:TimeObject. Furthermore, only geometric primitives are currently defined—these provide information about temporal position – but in future updates we expect that TimePrimitive will also describe topological primitives.

The two geometric primitives in the temporal dimension are the instant and the period. GML components are defined to support these as follows.

7.8.2.3 gml:TimeInstant; gml:timePosition

An instant is a zero-dimensional geometric primitive that represents position in time (it is equivalent to a point in space). In practice, an instant is an interval whose duration is less than a chronon—the resolution of the time scale..

The element **gml:TimeInstant** is declared as follows:

```
<element name="TimeInstant" type="gml:TimeInstantType" substitutionGroup="gml:_TimePrimitive"/>

<complexType name="TimeInstantType" final="#all">
  <complexContent>
    <extension base="gml:TimePrimitiveType">
      <sequence>
        <element ref="gml:timePosition"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

The method for identifying a temporal position is specific to each type of temporal reference system. Values that conform to the date, dateTime, gYear, and gYearMonth data types use lexical formats that are based on the ISO 8601 standard, as described in Part 2 of the XML Schema recommendation. A decimal value may be used in concert with reference systems such as GPS time or UNIX time. A URI value may be used to provide a reference to some era in an ordinal reference system (e.g. a geological epoch).

The gml:timePosition element is declared as follows:

```
<element name="timePosition" type="gml:TimePositionType"/>
```

The content of the position element is defined in two steps. In the definition of gml:TemporalPositionType the hierarchy of subtypes for temporal position described in ISO 19108 is collapsed in a union of the XML Schema simple types dateTime, date, gYearMonth and gYear that indicate temporal position relative to a specific reference system.

```
<simpleType name="TemporalPositionType">
  <union memberTypes="dateTime date gYearMonth gYear anyURI decimal"/>
</simpleType>
```

Elements that use this type may thus indicate date and time with varying degrees of precision: year, year-month, date, or dateTime (all ISO 8601 format). Note that the XML Schema dateTime type does not allow right-truncation (i.e. omitting seconds). Alternatively, an ordinal era may be referenced via URI, and a decimal value can be used to indicate the distance from the scale origin (e.g. UNIX time, GPS calendar).

In the definition of gml:TimePositionType we add three XML attributes

```
<complexType name="TimePositionType" final="#all">
  <simpleContent>
    <extension base="gml:TemporalPositionType">
```

```
      <attribute name="indeterminatePosition" type="gml:TimeIndeterminateValueType" use="optional"/>
      <attribute name="calendarEraName" type="string" use="optional"/>
      <attribute name="frame" type="anyURI" use="optional" default="#ISO-8601"/>
    </extension>
     </extension>
   </simpleContent>
 </complexType>
```

A time value is associated with a temporal reference system through the **frame** attribute that provides a URI reference that identifies a description of the reference system. Following ISO 8601, the Gregorian calendar with UTC is the default reference system, but others may also be used. The GPS calendar and the Julian calendar are alternative reference systems in common use.

For time values that identify position within a calendar, the **calendarEraName** attribute provides the name of the calendar era to which the date is referenced (e.g. the Meiji era of the Japanese calendar).

Inexact or 'fuzzy' temporal positions may be expressed by qualifying a specific timePosition using the optional **indeterminatePosition** attribute. This takes a value from an enumeration defined as follows:

```
<simpleType name="TimeIndeterminateValueType">
  <restriction base="string">
    <enumeration value="after"/>
    <enumeration value="before"/>
    <enumeration value="now"/>
    <enumeration value="unknown"/>
  </restriction>
</simpleType>
```

These values are interpretated as follows:

- "unknown" indicates that no specific value for temporal position is provided.
- "now" indicates that the specified value shall be replaced with the current temporal position whenever the value is accessed.
- "before" indicates that the actual temporal position is unknown, but it is known to be before the specified value.
- "after" indicates that the actual temporal position is unknown, but it is known to be after the specified value.

A value for indeterminatePosition can be used either

i. alone, or

ii. can qualify a specific value for temporal position (e.g. before 2002-12, after 1019624400).

The following are examples of how a TimeInstant may appear in a data instance:

```
<gml:TimeInstant gml:id="time3">
  <gml:timePosition>2002-11-25T13:20:20</gml:position>
</gml:TimeInstant>
```

```
<gml:TimeInstant gml:id="ogcEra">
  <gml:timePosition indeterminatePosition="after">1994</gml:position>
</gml:TimeInstant>
```

```
<gml:TimeInstant>
  <gml:timePosition indeterminatePosition="now"></gml:position>
</gml:TimeInstant>
```

```
<gml:TimeInstant>
  <gml:timePosition>http://my.big.org/instants/i56</gml:position>
</gml:TimeInstant>
```

```
<gml:TimeInstant frame="http://my.history.org/eras">
  <gml:timePosition indeterminatePosition="unknown" calendarEraName="bronzeAge"></gml:position>
</gml:TimeInstant>
```

A content model for an element whose value is a gml:TimeInstant is defined following the standard GML property pattern as follows:

```
<complexType name="TimeInstantPropertyType">
  <sequence>
    <element ref="gml:TimeInstant" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

7.8.2.4 gml:TimePeriod, gml:_duration

A period is a one-dimensional geometric primitive that represents extent in time (it is equivalent to a curve in space); it is an open interval bounded by beginning and end points (i.e. instants), and has length (i.e. duration). Its location in time is described by the temporal positions of the instants at which it begins and ends, as indicated by the *begin* and *end* properties of the TimePeriod class. The duration of the period is equal to the temporal distance between the two bounding temporal positions.

The element **gml:TimePeriod** is declared as follows:

```
<element name="TimePeriod" type="gml:TimePeriodType" substitutionGroup="gml:_TimePrimitive"/>
```

```
<complexType name="TimePeriodType" final="#all">
  <complexContent>
    <extension base="gml:TimePrimitiveType">
      <sequence>
        <element ref="gml:begin"/>
        <element ref="gml:end"/>
        <element ref="gml:_duration" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

where the gml:begin and gml:end elements are declared

```
<element name="begin" type="gml:TimeInstantPropertyType"/>
<element name="end" type="gml:TimeInstantPropertyType"/>
```

The length of the period is described using the (optional) abstract element gml:_duration, which is declared in the schema as follows:

```
<element name="_duration" type="gml:TimeDurationType" abstract="true">

<simpleType name="TimeDurationType">
  <union memberTypes="duration decimal"/>
</simpleType>
```

gml:_duration acts as the head of a substitution group. Its content model is a union of the XML Schema duration and decimal simpleTypes.

One member of the substitution group is gml:duration, which conforms to the ISO 8601 syntax for temporal length (e.g. P5DT4H30M) as implemented by the XML Schema duration type:

```
<element name="duration" type="duration" substitutionGroup="gml:_duration"/>
```

Another member of the substitution group is gml:interval which conforms to the ISO 11404 standard which is based on floating point values for temporal length.

```
<element name="interval" type="gml:TimeIntervalLengthType" substitutionGroup="gml:_duration"/>

<complexType name="TimeIntervalLengthType" final="#all">
  <simpleContent>
    <extension base="decimal">
      <attribute name="unit" type="gml:TimeUnitType" use="required"/>
      <attribute name="radix" type="positiveInteger" use="optional"/>
      <attribute name="factor" type="integer" use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

ISO 11404 syntax specifies the use of a positiveInteger together with appropriate values for radix and factor. The resolution of the time interval is to one radix ^(-factor) of the specified time unit (e.g. unit="second", radix="10", factor="3" specifices a resolution of milliseconds).

For example, to express a period length of 5 days, 14 hours, and 30 minutes either of the following instances are acceptable:

```
<duration>P5DT14H30M</duration>
<interval unit=“hour” radix=“10” factor=“0”>134.5</interval>
```

7.8.2.5 gml:timePrimitiveProperty, gml:timeStamp

gml: timePrimitiveProperty is a utility element provided for attaching a gml:_TimePrimitive to an object. It follows the usual GML property pattern and is declared in the schema as follows:

```
  <element name="timePrimitiveProperty" type="gml:TimePrimitivePropertyType"/>
  <element name="timeStamp" type="gml:TimePrimitivePropertyType"
substitutionGroup="gml:timePrimitiveProperty"/>

  <complexType name="TimePrimitivePropertyType">
    <sequence>
      <element ref="gml:_TimePrimitive" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
```

gml:timeStamp is a convenient synonym for gml:timePrimitiveProperty.

7.8.3 Temporal reference systems

A value in the time domain is measured relative to a temporal reference system. Common types of reference systems include calendars, ordinal temporal reference systems, and temporal coordinate systems (time elapsed since some epoch, e.g. UNIX time). The primary temporal reference system for use with geographic information is the Gregorian Calendar and 24 hour local or Coordinated Universal Time (UTC), but special applications may entail the use of alternative reference systems.

Three classes in Figure 7.8.2 are used to describe temporal reference systems. In GML these are reflected in three concrete elements: TimeReferenceSystem, TimeOrdinalReferenceSystem, and TimeOrdinalEra.

7.8.3.1 gml:TimeReferenceSystem

A reference system is characterized in terms of its domain of validity: the spatial and temporal extent over which it is applicable. The basic GML element for temporal reference systems is gml:TimeReferenceSystem. Its content model simply extends gml:DefinitionType (see clause 7.9.2.1) with one property: domainOfValidity. In the schema this is implemented as follows:

```
  <element name="TimeReferenceSystem" type="gml:TimeReferenceSystemType"
substitutionGroup="gml:_GML"/>

  <complexType name="TimeReferenceSystemType">
    <complexContent>
      <extension base="gml:DefinitionType">
        <sequence>
          <element name="domainOfValidity" type="string"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

The standard properties of a GML object are inherited from gml:AbstractGMLType. This element might appear in in an instance document as follows:

```
<gml:TimeReferenceSystem gml:id="JulianCalendar">
  <gml:description xlink:href="http://aa.usno.navy.mil/data/docs/JulianDate.html"/>
  <gml:name>Julian Calendar</gml:name>
  <gml:domainOfValidity>Western Europe</gml:domainOfValidity>
</gml:TimeReferenceSystem>
```

7.8.3.2 gml:TimeCoordinateSystem

Specifying temporal position in terms of calendar date and time of day complicates the computation of distances between points and the functional description of temporal operations. A temporal coordinate system may be used to support applications of this kind. A temporal coordinate system shall be based on a continuous interval scale defined in terms of a single time interval.

The element gml:TimeCoordinateSystem is declared as follows:

```
<element name="TimeCoordinateSystem" type="gml:TimeCoordinateSystemType"
substitutionGroup="gml:TimeReferenceSystem"/>

  <complexType name="TimeCoordinateSystemType" final="#all">
    <complexContent>
      <extension base="gml:TimeReferenceSystemType">
        <sequence>
          <element name="origin" type="gml:TimeInstantPropertyType"/>
          <element name="interval" type="gml:UnitOfMeasureType"/>
          <element name="incrementDirection" type="gml:SignType" minOccurs="0" default="+"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

The content model follows TM_CoordinateSystem from ISO 19108 except in the following details:

1. the origin is specified using a TimeInstantProperty, which permits more flexibility in representation and also supports referring to a value fixed elsewhere

2. the interval uses the standard gml:UnitOfMeasureType, defined in clause 7.10.2.

Coordinate systems might be described in data instances as follows:

```
<gml:TimeCoordinateSystem gml:id="Laser36">
  <gml:description>Time scale used during a laser experiment</gml:description>
  <gml:name>Laser timescale 36</gml:name>
  <gml:domainOfValidity>Laser laboratory</gml:domainOfValidity>
  <gml:origin>
    <gml:TimeInstant>
      <gml:timePosition>2002-11-28T12:50:00+08:00</gml:timePosition>
    </gml:TimeInstant>
  </gml:origin>
  <gml:interval uom="http://my.big.org/units/time#ps"/>
</gml:TimeCoordinateSystem>
```

```
<gml:TimeCoordinateSystem gml:id="geologyMa">
  <gml:name>Geological time system</gml:name>
  <gml:domainOfValidity>Earth</gml:domainOfValidity>
  <gml:origin>
    <gml:TimeInstant>
      <gml:description xlink:href="http://www.c14dating.com/agecalc.html">Conventional origin used for
carbon dating.  Equivalent to "present" for other radiometric dating techniques which have much lower
precision.</gml:description>
      <gml:timePosition>1950</gml:timePosition>
    </gml:TimeInstant>
  </gml:origin>
  <gml:interval uom="http://my.big.org/units/time#Ma"/>
  <gml:incrementDirection>-</gml:incrementDirection>
</gml:TimeCoordinateSystem>
```

7.8.3.3 gml:TimeOrdinalReferenceSystem

In some applications of geographic information—such as geology and archaeology—relative position in time is known more precisely than absolute time or duration. The order of events in time can be well established, but the magnitude of the intervals between them cannot be accurately determined; in such cases, the use of an ordinal temporal reference system is appropriate. An ordinal temporal reference system consists of a set of named eras that may be hierarchically structured such that an ordinal era at a given level of the hierarchy includes a sequence of coterminous shorter ordinal eras.

In Figure 7.8.2, an ordinal reference system is defined as a composition of one or more component eras, each of which may contain subsidiary eras as members. A given era may specify beginning and ending positions that calibrate the relative time scale.

The element gml:TimeOrdinalReferenceSystem implements this by adding a set of gml:component properties. In the schema this is implemented as follows:

```
  <element name="TimeOrdinalReferenceSystem" type="gml:TimeOrdinalReferenceSystemType"
substitutionGroup="gml:TimeReferenceSystem"/>

  <complexType name="TimeOrdinalReferenceSystemType" final="#all">
```

```
  <complexContent>
    <extension base="gml:TimeReferenceSystemType">
      <sequence>
        <element name="component" type="gml:TimeOrdinalReferenceSystemMemberType"
maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

The content model for the gml:component property is a gml:TimeOrdinalEra element.

```
<complexType name="TimeOrdinalReferenceSystemMemberType">
  <sequence>
    <element ref="gml:TimeOrdinalEra"/>
  </sequence>
</complexType>
```

7.8.3.4 gml:TimeOrdinalEra

The content model for gml:TimeOrdinalEra includes optional gml:begin and gml:end properties, and a set of gml:member elements, which contain futher gml:TimeOrdinalEra elements. In the schema this is implemented as follows:

```
<element name="TimeOrdinalEra" type="gml:TimeOrdinalEraType" substitutionGroup="gml:_GML"/>

<complexType name="TimeOrdinalEraType">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:begin" minOccurs="0"/>
        <element ref="gml:end" minOccurs="0"/>
        <element name="member" type="gml:TimeOrdinalReferenceSystemMemberType" minOccurs="0"
maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

This recursive inclusion of gml:TimeOrdinalEra elements allow the contruction of am arbitrary depth hierarchical ordinal reference schema, such that an ordinal era at a given level of the hierarchy includes a sequence of shorter, coterminous ordinal eras.

The example below shows a portion of the geological time scale depicted as an ordinal reference system:

```xml
<gml:TimeOrdinalReferenceSystem gml:id="GeologicalTimeScale">
  <gml:description xlink:href="ftp://ftp.iugs.org/pub/iugs/iugs_intstratchart.pdf"/>
  <gml:name>Geological time scale</gml:name>
  <gml:domainOfValidity>Earth</gml:domainOfValidity>
  <!-- Earlier eras omitted -->
  <gml:component>
    <gml:TimeOrdinalEra gml:id="Cenozoic">
      <gml:name>Cenozoic Era</gml:name>
      <gml:begin xlink:href="#basePaleocene"/>
      <gml:end xlink:href="#now"/>
      <gml:member>
        <gml:TimeOrdinalEra gml:id="Tertiary">
          <gml:name>Tertiary Period</gml:name>
          <gml:begin xlink:href="#basePaleocene"/>
          <gml:end xlink:href="#basePleistocene"/>
          <gml:member>
            <gml:TimeOrdinalEra gml:id="Paleogene">
              <gml:name>Paleogene sub-period</gml:name>
              <gml:begin xlink:href="#basePaleocene"/>
              <gml:end xlink:href="#baseMiocene"/>
              <gml:member>
                <gml:TimeOrdinalEra gml:id="Paleocene">
                  <gml:name>Paleocene Epoch</gml:name>
                  <gml:begin>
                    <gml:TimeInstant gml:id="basePaleocene">
                      <gml:timePosition frame="#geologyMa">65.0</gml:timePosition>
                    </gml:TimeInstant>
                  </gml:begin>
                  <gml:end xlink:href="#baseEocene"/>
                </gml:TimeOrdinalEra>
              </gml:member>
              <gml:member>
                <gml:TimeOrdinalEra gml:id="Eocene">
                  <gml:name>Paleocene Epoch</gml:name>
                  <gml:begin>
                    <gml:TimeInstant gml:id="baseEocene">
                      <gml:timePosition frame="#geologyMa">57.8</gml:timePosition>
                    </gml:TimeInstant>
                  </gml:begin>
                  <gml:end xlink:href="#baseOligocene"/>
                </gml:TimeOrdinalEra>
              </gml:member>
              <gml:member>
                <gml:TimeOrdinalEra gml:id="Oligocene">
                  <gml:name>Oligocene Epoch</gml:name>
                  <gml:begin>
                    <gml:TimeInstant gml:id="baseOligocene">
                      <gml:timePosition frame="#geologyMa">33.7</gml:timePosition>
                    </gml:TimeInstant>
                  </gml:begin>
                  <gml:end xlink:href="#baseMiocene"/>
                </gml:TimeOrdinalEra>
              </gml:member>
            </gml:TimeOrdinalEra>
          </gml:member>
  <!-- Neogene sub-period and Quaternerary period omitted -->
        </gml:TimeOrdinalEra>
      </gml:member>
    </gml:TimeOrdinalEra>
  </gml:component>
</gml:TimeOrdinalReferenceSystem>
```

Note that the use of references on various begin and end elements allows the numeric value of the boundaries between eras to be recorded once and then re-used many times as appropriate.

7.8.4 Representing dynamic features

A number of types and relationships are defined to represent the time-varying properties of geographic features. The dynamic feature schema is listed in Annex C; it is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:dynamicFeature:v3.0

A UML representation of the schema provides a visual summary of the principal elements. Each class, attribute or rolename in Figure 7.8.3 corresponds to an element or model group declared in the schema.

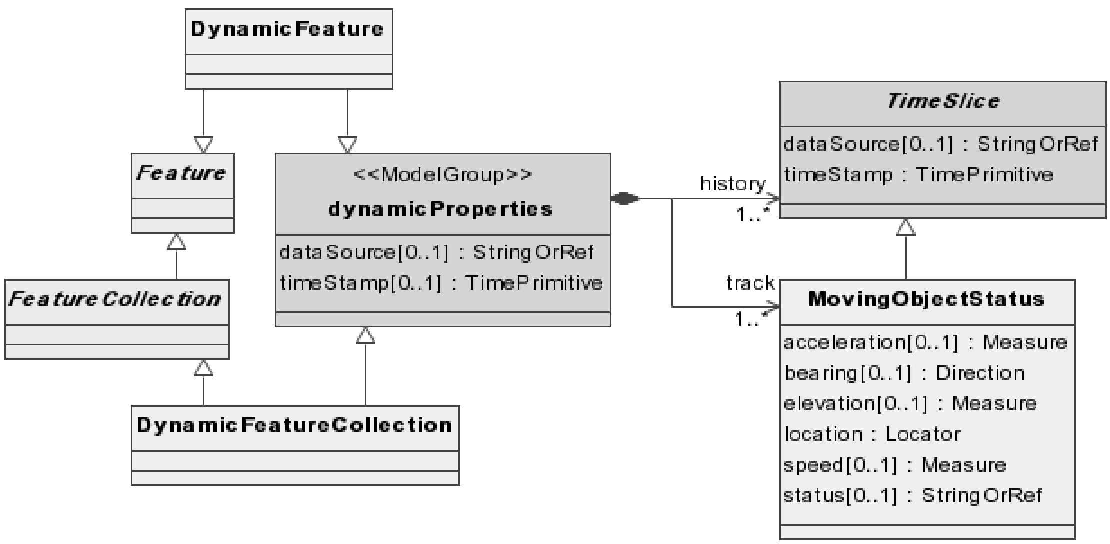


Figure 7.8.3 UML representation of the dynamic feature schema

In a comprehensive treatment of spatiotemporal modeling, Langran [see bibliography] distinguished three principal temporal entities: *states*, *events*, and *evidence*; this schema incorporates elements for each.

7.8.4.1 gml:dataSource

In GML, *evidence* is represented by a simple **dataSource** property that indicates the source of the temporal data (e.g. human observer, *in situ* sensor).

```
<element name="dataSource" type="gml:StringOrRefType" abstract="true"/>
```

7.8.4.2 Dynamic Properties

A utility group gml:dynamicProperties is defined in the schema as follows:

```
<group name="dynamicProperties">
  <sequence>
    <element ref="gml:timeStamp" minOccurs="0"/>
    <element ref="gml:history" minOccurs="0"/>
    <element ref="gml:dataSource" minOccurs="0"/>
  </sequence>
</group>
```

This allows an application schema developer to include dynamic properties in a content model in a standard fashion. The gml:timeStamp property is declared in temporal.xsd (described in clause 7.8.2.5 above). The other properties are declared elsewhere in this clause.

7.8.4.3 Dynamic Features

States are captured by time-stamped instances of a feature or f eature collection. The content model for a dynamic feature extends the standard AbstractFeatureType and FeatureCollectionType with the gml:dynamicProperties model group defined above (clause 7.8.4.2):

```
<complexType name="DynamicFeatureType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <group ref="gml:dynamicProperties"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="DynamicFeatureCollectionType">
  <complexContent>
    <extension base="gml:FeatureCollectionType">
      <group ref="gml:dynamicProperties"/>
    </extension>
  </complexContent>
</complexType>
```

Each time-stamped instance represents a ‘snapshot’ of a feature. The dynamic feature classes will normally be extended to suit particular applications. A dynamic feature bears either a time stamp or a history.

7.8.4.4 gml:_TimeSlice

To describe an *event* — an action that occurs at an instant — GML provides the abstract **TimeSlice** element, which is declared in the schema as follows:

```
  <element name="_TimeSlice" type="gml:AbstractTimeSliceType" abstract="true"
substitutionGroup="gml:_GML"/>

<complexType name="AbstractTimeSliceType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:timeStamp"/>
        <element ref="gml:dataSource" minOccurs="0"/>
      </sequence>
    </extension>
```



```
  </complexContent>
 </complexType>
```

A timeslice encapsulates the time-varying properties of a dynamic feature -- it must be extended to represent a timestamped projection of a specific feature. The dataSource property describes how the temporal data was acquired.

A TimeSlice instance is a GML object that encapsulates updates of the dynamic—or volatile—properties that reflect some change event; it thus includes only those feature properties that have actually changed due to some process. For example, suppose that ownership of a building changes and it is renamed. If no other building properties have changed, then the event will only include the updated name. The `TimeSlice` class basically provides a facility for *attribute-level* time stamping, in contrast to the *object-level* time stamping of dynamic feature instances.

The time slice can thus be viewed as event or process-oriented, whereas a snapshot is more state or structure-oriented. A timeslice has richer causality, whereas a snapshot merely portrays the status of the whole. For example, a feature collection might have a 'life cycle' represented by a sequence of snapshots (Figure 7.8.4).

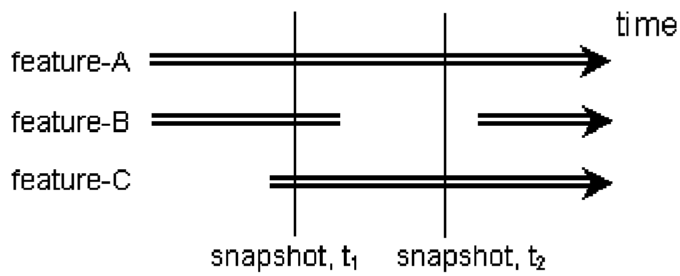


Figure 7.8.4 The life cycle of a feature collection

At instant t_1, feature-A, feature-B, and feature-C are all members of the collection. However, at instant t_2 only feature-A and feature-B are members. Closer examination of the history of feature-B will reveal its ephemeral nature (e.g. a building is dismantled and reconstructed on a seasonal basis).

7.8.4.5 gml:MovingObjectStatus

The gml:MovingObjectStatus element is one example of how the gml:TimeSlice may be extended. This element provides a standard method to capture a record of the status of a moving object. It is declared as follows:

```
 <element name="MovingObjectStatus" type="gml:MovingObjectStatusType"
substitutionGroup="gml:TimeSlice"/>

 <complexType name="MovingObjectStatusType">
  <complexContent>
   <extension base="gml:AbstractTimeSliceType">
    <sequence>
     <element ref="gml:location"/>
```

```
      <element name="speed" type="gml:MeasureType" minOccurs="0"/>
      <element name="bearing" type="gml:DirectionPropertyType" minOccurs="0"/>
      <element name="acceleration" type="gml:MeasureType" minOccurs="0"/>
      <element name="elevation" type="gml:MeasureType" minOccurs="0"/>
      <element ref="gml:status" minOccurs="0"/>
    </sequence>
   </extension>
  </complexContent>
 </complexType>
```

A gml:MovingObjectStatus element allows the user to describe the present location, along with the speed, bearing, acceleration and elevation of an object in a particular time slice.

Additional information about the current status of the object can be recorded in the gml:status element, declared as follows:

```
 <element name="status" type="gml:StringOrRefType"/>
```

7.8.4.6 gml:history

A generic sequence of events constitute a **history** of an object. This GML property element is declared in the schema as follows:

```
 <element name="history" type="gml:HistoryPropertyType"/>

 <complexType name="HistoryPropertyType">
  <sequence>
   <element ref="gml:_TimeSlice" maxOccurs="unbounded"/>
  </sequence>
 </complexType>
```

The gml:history element contains a set of elements in the substitution group headed by the abstract element gml:_TimeSlice, representing the time-varying properties of interest. The history property of a dynamic feature associates a feature instance with a sequence of time slices (i.e. change events) that encapsulate the evolution of the feature.

7.8.4.7 gml:track

A track is a specific kind of history. The gml:track element is declared in the schema as follows:

```
<element name="track" type="gml:TrackType" substitutionGroup="gml:history"/>

 <complexType name="TrackType">
  <complexContent>
   <restriction base="gml:HistoryPropertyType">
    <sequence>
     <element ref="gml:MovingObjectStatus" maxOccurs="unbounded"/>
    </sequence>
   </restriction>
  </complexContent>
 </complexType>
```

If the feature represents a moving object such as a ground vehicle or a ship, then the **track** property comprises a sequence of MovingObjectStatus elements. For example, a dynamic feature such as a cyclone may have a gml:track property such as shown in the following fragment:

```
<gml:track>
 <gml:MovingObjectStatus>
  <gml:timeStamp><gml:TimeInstant>
   <gml:timePosition>2005-11-28T13:00:00</gml:timePosition>
  </gml:TimeInstant></gml:timeStamp>
  <gml:location><gml:Point>
   <gml:pos>140. -35.</gml:pos>
  </gml:Point></gml:location>
  <gml:speed uom="#kph">12.</gml:speed>
  <gml:bearing>
   <gml:CompassPoint>SE</gml:CompassPoint>
  </gml:bearing>
 </gml:MovingObjectStatus>
 <gml:MovingObjectStatus>
  <gml:timeStamp><gml:TimeInstant>
   <gml:timePosition>2005-11-28T14:00:00</gml:timePosition>
  </gml:TimeInstant></gml:timeStamp>
  <gml:location><gml:Point>
   <gml:pos>140.1 -34.9</gml:pos>
  </gml:Point></gml:location>
  <gml:speed uom="#kph">23.</gml:speed>
  <gml:bearing>
   <gml:CompassPoint>ESE</gml:CompassPoint>
  </gml:bearing>
 </gml:MovingObjectStatus>
</gml:track>
```

7.9 Definitions and Dictionaries

7.9.1 Overview

Many applications require definitions of terms, which are used within instance documents as the values of certain properties or as reference information to tie properties to standard information values in some way. Units of measure, and descriptions of measurable phenomena, are two particular examples. Components to support this functionality are provided in GML in the form of

1. a generic Definition, which may serve as the basis for more specialized definitions
2. a generic Dictionary, also known as DefinitionCollection, which allows a set of definitions or references to definitions to be collected

7.9.2 Dictionary schema

The dictionary schema is listed in Annex C; it is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:dictionary:v3.0

A UML representation of the schema provides a visual summary of the principal elements. Each class, attribute or rolename in Figure 7.9.1 corresponds to an element declared in the schema.

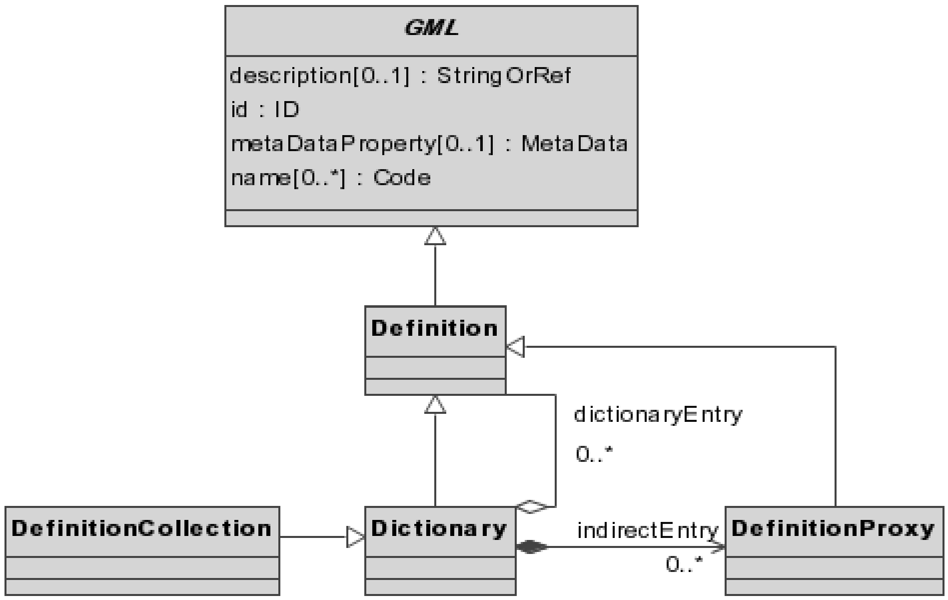


Figure 7.9.1 Generic definition and dictionary components

7.9.2.1 gml:Definition

The basic Definition element is declared in the schema as follows:

```
<element name="Definition" type="gml:DefinitionType" substitutionGroup="gml:_GML"/>

<complexType name="DefinitionType">
  <complexContent>
    <restriction base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:description" minOccurs="0"/>
        <element ref="gml:name" minOccurs="1" maxOccurs="unbounded"/>
      </sequence>
      <attribute ref="gml:id" use="required"/>
    </restriction>
  </complexContent>
</complexType>
```

The content model for a generic definition is a simple derivation from AbstractGMLType. Since it is necessary to be able to refer to a definition, the gml:id attribute which provides the database handle for a GML object is mandatory. The "description" element should hold the definition whenever only text is needed *or carry a link to a description elsewhere.* The "name" elements may provide one or more terms and synonyms for which this is the definition. The "metaDataProperty" elements can be used to reference or include more information about this definition.

7.9.2.2 gml:Dictionary, gml:DefinitionCollection

Sets of definitions may be collected into dictionaries or collections. These are declared in the schema as follows:

```
<element name="Dictionary" type="gml:DictionaryType" substitutionGroup="gml:Definition"/>
<element name="DefinitionCollection" type="gml:DictionaryType" substitutionGroup="gml:Definition"/>

<complexType name="DictionaryType">
  <complexContent>
    <extension base="gml:DefinitionType">
      <choice minOccurs="0" maxOccurs="unbounded">
        <element ref="gml:dictionaryEntry"/>
        <element ref="gml:indirectEntry"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

A Dictionary is a non-abstract collection of definitions. These definitions are referenced from other places, in the same and different XML documents.

The Dictionary content model adds a list of dictionaryEntry and indirectEntry properties that contain or reference Definition objects. A database handle (gml:id attribute) is required, in order that this collection may be referred to. The standard metaDataProperty, description and name are available to reference or contain more information about this dictionary. The inherited "description" element can be used for a description of this dictionary. The inherited "name" element can be used for the name(s) of this dictionary.

7.9.2.3 gml:dictionaryEntry

These elements contain or refer to the definitions which are members of this dictionary. The element gml:dictionaryEntry is declared as follows:

```
<element name="dictionaryEntry" type="gml:DictionaryEntryType"/>
<element name="definitionMember" type="gml:DictionaryEntryType"
substitutionGroup="gml:dictionaryEntry"/>

<complexType name="DictionaryEntryType">
  <sequence>
    <element ref="gml:Definition" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

The content model follows the standard GML property pattern, so a gml:dictionaryEntry may either contain or refer to a single gml:Definition. Since gml:Dictionary is substitutable for gml:Definition, the content of an entry can itself be a lower level dictionary or definition collection.

Note that if the value is provided by reference, this definition does *not* carry a handle (gml:id) in this context, so does *not* allow external references to this specific definition in this context. When used in this way the referenced definition will usually be in a dictionary in the same XML document.

7.9.2.4 gml:indirectEntry, gml:DefinitionProxy

If a definition must be included by reference, in its context within the current collection, then gml:indirectEntry must be used. This is declared as follows:

```
<element name="indirectEntry" type="gml:IndirectEntryType"/>

<complexType name="IndirectEntryType">
  <sequence>
    <element ref="gml:DefinitionProxy"/>
  </sequence>
</complexType>
```

A gml:indirectEntry contains a proxy object gml:DefinitionProxy which is declared as follows:

```
<element name="DefinitionProxy" type="gml:DefinitionProxyType" substitutionGroup="gml:Definition"/>

<complexType name="DefinitionProxyType">
  <complexContent>
    <extension base="gml:DefinitionType">
      <sequence>
        <element ref="gml:definitionRef"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

A gml:DefinitionProxy carries a mandatory handle (gml:id), and contains a reference to a definition represented elsewhere. This entry is expected to be convenient in allowing multiple elements in one XML document to contain short (abbreviated XPointer) references, which are resolved to an external definition provided in a Dictionary element in the same XML document.

The reference is carried by a gml:definitionRef element which is declared as follows.

```
<element name="definitionRef" type="gml:ReferenceType"/>
```

This uses the gml:ReferenceType which is described in clause 7.2.2.3.3. The remote entry referenced can be in a dictionary in the same or different XML document.

The following examples shows two instances of dictionaries:

```
<gml:Dictionary gml:id="rockTypes">
```

```
<gml:Dictionary gml:id="AbridgedGMLdictionary">
  <description>Abridged GML dictionary.  See section 4 Terms and definitions for the rest.</description>
  <name>GML dictionary</name>
  <dictionaryEntry>
    <Definition gml:id="association">
      <description>A structural relationship that describes a set of links, in which a link is a connection among objects; the semantic relationship between two or more classifiers that involves the connections among their instances (Booch, 1999).</description>
      <name>association</name>
    </Definition>
  </dictionaryEntry>
  <dictionaryEntry>
    <DefinitionCollection gml:id="attribute">
      <description>named property</description>
      <name>attribute</name>
      <definitionMember>
        <Definition gml:id="attribute_in_UML">
          <description>Named property of a class that describes the range of values that instances of the property may hold. (Booch,1999)</description>
          <name codeSpace="UML">attribute</name>
        </Definition>
      </definitionMember>
      <definitionMember>
        <Definition gml:id="attribute_in_XML">
          <description>An information item in the XML Information Set [Infoset] </description>
          <name codeSpace="XML">attribute</name>
        </Definition>
      </definitionMember>
    </DefinitionCollection>
  </dictionaryEntry>
  <dictionaryEntry>
    <DefinitionCollection gml:id="property">
      <description>A characteristic of an object</description>
      <name>property</name>
      <definitionMember>
        <Definition gml:id="property_in_GML">
          <description>A characteristic of a GML object.</description>
          <name codeSpace="GML">property</name>
        </Definition>
      </definitionMember>
      <definitionMember>
        <DefinitionCollection gml:id="property_in_UML">
          <description>A characteristic of a UML object.</description>
          <name codeSpace="UML">property</name>
          <dictionaryEntry xlink:href="#association"/>
          <dictionaryEntry xlink:href="#attribute_in_UML"/>
        </DefinitionCollection>
      </definitionMember>
    </DefinitionCollection>
  </dictionaryEntry>
  <indirectEntry>
    <DefinitionProxy gml:id="property_proxy">
      <name>property</name>
      <definitionRef xlink:href="#property"/>
    </DefinitionProxy>
  </indirectEntry>
</gml:Dictionary>
```

7.9.2.5 Using Definitions and Dictionaries

Dictionaries and Definitions are GML Objects, so may be found in independent GML data instance documents.

In application schemas it might be useful to attach a Dictionary or Definition Collection, or Definitions to a feature collection in order to record definitions used in properties of members of the collection.

7.10 Units, Measures and Values

7.10.1 Introduction

Several GML schemas introduce components that concern or require quantitative values which use a reference scale or units of measure. In Clause 7.2, basicTypes.xsd provides MeasureType, MeasureListType and MeasureOrNullListType definitions to enable GML properties and objects to carry units of measure, according to the following pattern:

```
<abc:length uom = “#m”>100</abc:length>
```

The attribute “uom” means “unit of measure” and holds a URI which refers to a resource that defines the units. The following schema documents describe components concerning this topic:

- units.xsd declares a set of components for defining units of measure
- measures.xsd declares a set of typed measures;
- valueObjects.xsd describes structures for aggregates and lists of measures.

7.10.2 Units schema

Several GML schemas define components that concern or require a reference scale or units of measure. Units are required for quantities that may occur as values of properties of feature types, as the results of observations, in the range parameters of a coverage, and for measures used in Coordinate Reference System definitions. The schema document units.xsd defines components to support the definition of units of measure. The units schema is listed in Annex C; it is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:units:v3.0

The basic unit definition is an extension of the general gml:Definition element defined in clause 7.9.2. Three specialized elements for unit definition are further derived from this, summarized by the UML class diagram shown in Figure 7.9.2.

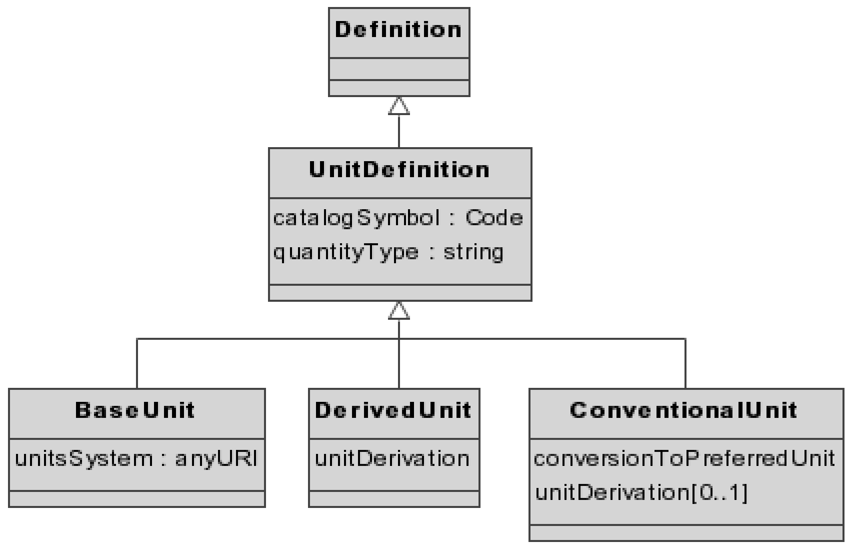


Figure 7.9.2 General model for unit definitions

This model is based on the SI system of units [ISO 1000], which distinguishes between Base Units and Derived Units.

- **Base Units** are the preferred units for a set of orthogonal fundamental quantities which define the particular system of units, which may not be derived by combination of other base units.
- **Derived Units** are the preferred units for other quantities in the system, which may be defined by algebraic combination of the base units.

In some application areas **Conventional units** are used, which may be converted to the preferred units using a scaling factor or a formula which defines a re-scaling and offset. The set of preferred units for all quantity types in a particular system of units is composed of the union of its base units and derived units.

7.10.2.1 Using Unit Definitions

Unit definitions are substitutable for the gml:Definition element declared as part of the dictionary model. A dictionary that contains only unit definitions and references to unit definitions is a units dictionary.

7.10.2.2 gml:unitOfMeasure

The element gml:unitOfMeasure is a property element to refer to a unit of measure. It is declared in the schema as follows:

```
<element name="unitOfMeasure" type="gml:UnitOfMeasureType"/>

<complexType name="UnitOfMeasureType">
  <sequence/>
  <attribute name="uom" type="anyURI" use="required"/>
</complexType>
```


This is an empty element which carries a reference to a unit of measure definition. This element may appear in a data instance as follows:

```
<unitOfMeasure uom="#m"/>
<unitOfMeasure uom="http://my.standards.org/units/length/metre"/>
```

If the unit of measure definition is within the same XML document, the URI may be a barename Xpointer, in which the "#" symbol is prepended to the value of the attribute of XML type ID carried by the definition (e.g. gml:id). For convenience, the ID may be a mnemonic abbreviation of the unit name. Thus, in the first case the reference is to an element in the same document which carries gml:id="m". In the second example the reference is to a definition provided by an external service.

7.10.2.3 gml:UnitDefinition

A UnitDefinition is a general definition of a unit of measure. This generic element is used only for units for which no relationship with other units or units systems is known. It is declared in the schema as follows:

```
<element name="UnitDefinition" type="gml:UnitDefinitionType" substitutionGroup="gml:Definition"/>

<complexType name="UnitDefinitionType">
  <complexContent>
    <extension base="gml:DefinitionType">
      <sequence>
        <element ref="gml:quantityType"/>
        <element ref="gml:catalogSymbol" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

The content model of gml:UnitDefinition adds two additional properties to gml:Definition (described in clause 7.9.2.1), gml:quantityType and gml:catalogSymbol. gml:quantityType.

7.10.2.4 gml: quantityType

The gml:quantityType property indicates the phenomenon to which the units apply. It is declared as follows:

```
<xsd:element name="quantityType" type="gml:StringOrRefType"/>
```

This element contains an informal description of the phenomenon or type of quantity that is measured or observed. For example, "length", "angle", "time", "pressure", or "temperature". When the quantity is the result of an observation or measurement, this term is known as Observable Type or Measurand.

7.10.2.5 gml:catalogSymbol

The catalogSymbol is the preferred lexical symbol used for this unit of measure. It is declared as follows:

```
<xsd:element name="catalogSymbol" type="gml:CodeType"/>
```

The "codeSpace" attribute in "CodeType" identifies a namespace for the catalog symbol value, and might reference the external catalog. The "string" value in "CodeType" contains the value of a symbol that should be unique within this catalog namespace. This symbol often appears explicitly in the catalog, but it could be a combination of symbols using a specified algebra of units. For example, the symbol "cm" might indicate that it is the "m" symbol combined with the "c" prefix.

7.10.2.6 gml:BaseUnit, gml:unitsSystem

A base unit is a unit of measure that cannot be derived by combination of other base units within a particular system of units. For example, in the SI system of units, the base units are metre, kilogram, second, Ampere, Kelvin, mole, and candela, for the quantity types length, mass, time interval, electric current, thermodynamic temperature, amount of substance and luminous intensity, respectively.

This is supported using the gml:BaseUnit element which is declared as follows:

```
<element name="BaseUnit" type="gml:BaseUnitType" substitutionGroup="gml:UnitDefinition"/>

<complexType name="BaseUnitType">
  <complexContent>
    <extension base="gml:UnitDefinitionType">
      <sequence>
        <element name="unitsSystem" type="gml:ReferenceType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

gml:BaseUnit extends generic gml:UnitDefinition with the property gml:unitsSystem, which carries a reference to the units system to which this base unit is asserted to belong.

7.10.2.7 gml:DerivedUnit

Derived units are defined by combination of other units. Derived units are used for quantities other than those corresponding to the base units, such as hertz (s^{-1}) for frequency, Newton ($kg.m/s^2$) for force. Derived units based directly on base units are usually preferred for quantities other than the fundamental quantities within a system. If a derived unit is not the preferred unit, the ConventionalUnit element should be used instead. The gml:DerivedUnit element is declared as follows:

```
<element name="DerivedUnit" type="gml:DerivedUnitType" substitutionGroup="gml:UnitDefinition"/>

<complexType name="DerivedUnitType">
  <complexContent>
```

```
    <extension base="gml:UnitDefinitionType">
      <sequence>
        <element ref="gml:unitDerivation"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

The gml:DerivedUnit extends gml:UnitDefinition with the property gml:unitDerivation.

7.10.2.8 gml:unitDerivation, gml:unitTerm

The element gml:unitDerivation is declared as follows:

```
<element name="unitDerivation" type="gml:UnitDerivationType"/>

<complexType name="UnitDerivationType">
  <sequence>
    <element ref="gml:unitTerm" maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

A gml:unitDerivation element contains a set of gml:unitTerm elements that are combined to form the derived unit of measure. Each unitTerm carries an integer exponent. The terms are combined by raising each referenced unit to the power of their exponent and forming the product. The gml:unitTerm element is declared as follows:

```
<element name="unitTerm" type="gml:UnitTermType"/>

<complexType name="UnitTermType">
  <complexContent>
    <extension base="gml:UnitOfMeasureType">
      <attribute name="exponent" type="integer"/>
    </extension>
  </complexContent>
</complexType>
```

This unit term references another unit of measure (uom) and provides an integer exponent applied to that unit in defining the compound unit. The exponent can be positive or negative, but not zero.

7.10.2.9 gml:ConventionalUnit

Conventional units that are neither base units nor defined by direct combination of base units are used in many application domains. For example electronVolt for energy, feet and nautical miles for length. In most cases there is a known, usually linear, conversion to a preferred unit which is either a base unit or derived by direct combination of base units. The gml:ConventionalUnit element is declared as follows:

```
<element name="ConventionalUnit" type="gml:ConventionalUnitType"
substitutionGroup="gml:UnitDefinition"/>

<complexType name="ConventionalUnitType">
  <complexContent>
```

```
    <extension base="gml:UnitDefinitionType">
      <sequence>
        <choice>
          <element ref="gml:conversionToPreferredUnit"/>
          <element ref="gml:roughConversionToPreferredUnit"/>
        </choice>
        <element ref="gml:unitDerivation" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

The gml:ConventionalUnit extends gml:UnitDefinition with a property that describes a conversion to a preferred unit for this quantity. When the conversion is exact, the element gml:conversionToPreferredUnit should be used, or when the conversion is not exact the element gml:roughConversionToPreferredUnit is available. Both of these elements have the same content model. The gml:unitDerivation property defined above is included to allow a user to optionally record how this unit may be derived from other ("more primitive") units.

7.10.2.10 gml:conversionToPreferredUnit, gml:factor, gml:formula

The elements gml:conversionToPreferredUnit and gml:roughConversionToPreferredUnit represent parameters used to convert conventional units to preferred units for this quantity type. A preferred unit is either a Base Unit or a Derived Unit that is selected for all values of one quantity type. These conversions are declared in the schema as follows:

```
<element name="conversionToPreferredUnit" type="gml:ConversionToPreferredUnitType"/>

<element name="roughConversionToPreferredUnit" type="gml:ConversionToPreferredUnitType"/>

<complexType name="ConversionToPreferredUnitType">
  <complexContent>
    <extension base="gml:UnitOfMeasureType">
      <choice>
        <element ref="gml:factor"/>
        <element ref="gml:formula"/>
      </choice>
    </extension>
  </complexContent>
</complexType>
```

The inherited attribute "uom" references the preferred unit that this conversion applies to. The conversion of a unit to the preferred unit for this quantity type is specified by an arithmetic conversion (scaling and/or offset). The content model extends gml:UnitOfMeasureType, which has a mandatory attribute uom which identifies the preferred unit for the quantity type that this conversion applies to. The conversion is specified by a choice of gml:factor or gml:formula, declared as follows.

```
<element name="factor" type="double"/>
```

This defines the scale factor by which a value using the conventional unit of measure can be multiplied to obtain the corresponding value using the preferred unit of measure.

```
<element name="formula" type="gml:FormulaType"/>

<complexType name="FormulaType">
  <sequence>
    <element name="a" type="double" minOccurs="0"/>
    <element name="b" type="double"/>
    <element name="c" type="double"/>
    <element name="d" type="double" minOccurs="0"/>
  </sequence>
</complexType>
```

This formula defines the parameters of a simple formula by which a value using the conventional unit of measure can be converted to the corresponding value using the preferred unit of measure. The formula element contains elements a, b, c and d, whose values use the XML Schema type "double". These values are used in the formula y = (a + bx) / (c + dx), where x is a value using this unit, and y is the corresponding value using the base unit. The elements a and d are optional, and if values are not provided, those parameters are considered to be zero. If values are not provided for both a and d, the formula is equivalent to a fraction with numerator and denominator parameters.

7.10.2.11 Example of units dictionary

This dictionary contains definitions corresponding to all the base and derived units defined by in the SI system [SI], plus a selection of conventional units to illustrate the usage of these components.

```xml
  <gml:description>A dictionary of units of measure</gml:description>
  <gml:name>OWS-1.2 Units</gml:name>
  <gml:dictionaryEntry>
    <gml:DefinitionCollection gml:id="SIBaseUnits">
      <gml:description>The Base Units from the SI units system.</gml:description>
      <gml:name>SI Base Units</gml:name>
      <gml:dictionaryEntry>
        <gml:BaseUnit gml:id="m">
          <gml:description>The metre is the length of the path travelled by light in vacuum during a time interval
of 1/299 792 458 of a second.</gml:description>
          <gml:name codeSpace="http://www.bipm.fr/en/3_SI/base_units.html">metre</gml:name>
          <gml:name xml:lang="en/US">meter</gml:name>
          <gml:quantityType>length</gml:quantityType>
          <gml:catalogSymbol
codeSpace="http://www.bipm.fr/en/3_SI/base_units.html">m</gml:catalogSymbol>
          <gml:unitsSystem xlink:href="http://www.bipm.fr/en/3_SI"/>
        </gml:BaseUnit>
      </gml:dictionaryEntry>
      <gml:dictionaryEntry>
        <gml:BaseUnit gml:id="kg">
          <gml:description>The kilogram is the unit of mass; it is equal to the mass of the international prototype
of the kilogram. </gml:description>
          <gml:name codeSpace="http://www.bipm.fr/en/3_SI/base_units.html">kilogram</gml:name>
          <gml:quantityType>Mass</gml:quantityType>
          <gml:catalogSymbol
codeSpace="http://www.bipm.fr/en/3_SI/base_units.html">kg</gml:catalogSymbol>
          <gml:unitsSystem xlink:href="http://www.bipm.fr/en/3_SI"/>
        </gml:BaseUnit>
      </gml:dictionaryEntry>
      <gml:dictionaryEntry>
        <gml:BaseUnit gml:id="s">
          <gml:description>The second is the duration of 9 192 631 770 periods of the radiation corresponding
to the transition between the two hyperfine levels of the ground state of the caesium 133
atom.</gml:description>
          <gml:name codeSpace="http://www.bipm.fr/en/3_SI/base_units.html">second</gml:name>
          <gml:quantityType>Time</gml:quantityType>
          <gml:catalogSymbol
codeSpace="http://www.bipm.fr/en/3_SI/base_units.html">s</gml:catalogSymbol>
          <gml:unitsSystem xlink:href="http://www.bipm.fr/en/3_SI"/>
        </gml:BaseUnit>
      </gml:dictionaryEntry>
      <gml:dictionaryEntry>
        <gml:BaseUnit gml:id="A">
          <gml:description>The ampere is that constant current which, if maintained in two straight parallel
conductors of infinite length, of negligible circular cross-section, and placed 1 metre apart in vacuum, would
produce between these conductors a force equal to 2 × 10-7 newton per metre of length.
</gml:description>
          <gml:name codeSpace="http://www.bipm.fr/en/3_SI">Ampere</gml:name>
          <gml:quantityType>Electric current</gml:quantityType>
          <gml:catalogSymbol codeSpace="http://www.bipm.fr/en/3_SI">A</gml:catalogSymbol>
          <gml:unitsSystem xlink:href="http://www.bipm.fr/en/3_SI"/>
        </gml:BaseUnit>
      </gml:dictionaryEntry>
      <gml:dictionaryEntry>
        <gml:BaseUnit gml:id="K">
          <gml:description>The kelvin, unit of thermodynamic temperature, is the fraction 1/273.16 of the
thermodynamic temperature of the triple point of water.</gml:description>
          <gml:name codeSpace="http://www.bipm.fr/en/3_SI">kelvin</gml:name>
          <gml:quantityType>Thermodynamic temperature</gml:quantityType>
          <gml:catalogSymbol codeSpace="http://www.bipm.fr/en/3_SI">K</gml:catalogSymbol>
          <gml:unitsSystem xlink:href="http://www.bipm.fr/en/3_SI"/>
        </gml:BaseUnit>
      </gml:dictionaryEntry>
      <gml:dictionaryEntry>
        <gml:BaseUnit gml:id="mol">
          <gml:description>1. The mole is the amount of substance of a system which contains as many
elementary entities as there are atoms in 0.012 kilogram of carbon 12.
2. When the mole is used, the elementary entities must be specified and may be atoms, molecules, ions,
electrons, other particles, or specified groups of such particles. </gml:description>
          <gml:name codeSpace="http://www.bipm.fr/en/3_SI">mole</gml:name>
          <gml:quantityType>Amount of substance</gml:quantityType>
          <gml:catalogSymbol codeSpace="http://www.bipm.fr/en/3_SI">mol</gml:catalogSymbol>
          <gml:unitsSystem xlink:href="http://www.bipm.fr/en/3_SI"/>
        </gml:BaseUnit>
      </gml:dictionaryEntry>
```

7.10.3 Measures schema

A schema for specific measure types is listed in Annex C; it is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:measures:v3.0

gml:MeasureType is defined in basicTypes.xsd. The measure types defined here correspond with a set of utility measure types described in ISO 19103. The XML implementation is based on the XML Schema simpleType “double” which supports both decimal and scientific notation, and includes an XML attribute “uom” which refers to the units of measure for the value. Note that, there is no requirement to *store* values using any particular format, and applications receiving elements of this type may choose to coerce the data to any other type as convenient.

7.10.3.1 gml:measure

This is the value of a quantity, together with its units. It is declared as follows:

```
<xsd:element name="measure" type="gml:MeasureType"/>
```

In an instance document the element might appear:

```
<gml:measure uom="#m">1.76</gml:measure>
```

The XML attribute **uom** is used to hold a reference to the scale or units by which the amount should be multiplied.

The "uom" attribute uses a URI to refer to a unit of measure definition. The definition may be within the same XML document or external. A definition within the same document would normally use the UnitDefinition element described above, which carries an ID handle as the value of its gml:id attribute. A measure element referring to this definition would normally use the abbreviated Xpointer form of URI.

For convenience the handle on the definition (the value of the gml:id attribute) will normally be mnemonic based on the unit name, such as “m” for metre. So if the definition is in the same document as the reference, the URI may be an abbreviated Xpointer reference [URI] which is presented with the "#" symbol prepended to the handle, i.e. “#m” as in this example.

7.10.3.2 Scalar measure types

A set of specific measure types are defined as simple aliases of gml:MeasureType. A prototypical definition is as follows:

```
<xsd:complexType name="LengthType">
  <xsd:simpleContent>
    <xsd:extension base="gml:MeasureType"/>
  </xsd:simpleContent>
```

```
</xsd:complexType>
```

This content model supports the description of a length (or distance) quantity, with its units. The unit of measure referenced by uom must be suitable for a length, such as metres or feet.

The other measure types that are defined following this pattern are:

ScaleType, TimeType, GridLengthType, AreaType, VolumeType, VelocityType.

Elements using these content models might appear in a data instance as follows:

```
<my:length uom="#m">1.76</my:length>

<my:scale uom="#percent">20.</my:scale>

<my:time uom="#minutes">30.</my:time>
```

Note that this element addresses the same functional requirements as the elements in the **gml:_duration** substitution group, defined in the temporal.xsd schema document.

```
<my:gridLength uom="#pixelSpacing">480</my:gridLength>
<my:gridLength uom="#imageHeight">0.00208333333333</my:gridLength>

<my:area uom="#Ha">1.76</my:area>

<my:volume uom="#l">0.45</my:volume>

<my:velocity uom="#kmph">73.0</my:velocity>
```

7.10.3.3 gml:angle

The gml:angle element is used to record the value of an angle quantity as a single number, with its units. gml:AngleType is derived trivially from gml:MeasureType using the method described above, with the restriction that the unit of measure referenced by **uom** must be suitable for an angle, such as degrees or radians. It is declared in the schema as follows:

```
<element name="angle" type="gml:AngleType"/>

<complexType name="AngleType">
  <simpleContent>
    <extension base="gml:MeasureType"/>
  </simpleContent>
</complexType>
```

In an instance document the element might appear:

```
<gml:angle uom="#gradians">95.</gml:angle>
```

7.10.3.4 gml:dmsAngle

The gml:dmsAngle element is used to record the value of an angle in degree-minute-second or degree-minute format. It uses the following schema declarations:

```
<element name="dmsAngle" type="gml:DMSAngleType"/>

<complexType name="DMSAngleType">
  <sequence>
    <element ref="gml:degrees"/>
    <choice minOccurs="0">
      <element ref="gml:decimalMinutes"/>
      <sequence>
        <element ref="gml:minutes"/>
        <element ref="gml:seconds" minOccurs="0"/>
      </sequence>
    </choice>
  </sequence>
</complexType>

<element name="degrees" type="gml:DegreesType"/>

<complexType name="DegreesType">
  <simpleContent>
    <extension base="gml:DegreeValueType">
      <attribute name="direction">
        <simpleType>
          <restriction base="string">
            <enumeration value="N"/>
            <enumeration value="E"/>
            <enumeration value="S"/>
            <enumeration value="W"/>
            <enumeration value="+"/>
            <enumeration value="-"/>
          </restriction>
        </simpleType>
      </attribute>
    </extension>
  </simpleContent>
</complexType>

<simpleType name="DegreeValueType">
  <restriction base="nonNegativeInteger">
    <maxInclusive value="359"/>
  </restriction>
</simpleType>

<element name="decimalMinutes" type="gml:DecimalMinutesType"/>

<simpleType name="DecimalMinutesType">
  <restriction base="decimal">
    <minInclusive value="0.00"/>
    <maxExclusive value="60.00"/>
  </restriction>
</simpleType>

<element name="minutes" type="gml:ArcMinutesType"/>

<simpleType name="ArcMinutesType">
  <restriction base="nonNegativeInteger">
    <maxInclusive value="59"/>
  </restriction>
</simpleType>
```

```
<element name="seconds" type="gml:ArcSecondsType"/>

<simpleType name="ArcSecondsType">
  <restriction base="decimal">
    <minInclusive value="0.00"/>
    <maxExclusive value="60.00"/>
  </restriction>
</simpleType>
```

In an instance document the element might appear in one of the following forms:

```
<gml:dmsAngle>
  <gml:degrees direction="S">35</gml:degrees>
  <gml:minutes>45</gml:minutes>
  <gml:seconds>21.98</gml:seconds>
</gml:dmsAngle>

<gml:dmsAngle>
  <gml:degrees direction="S">35</gml:degrees>
  <gml:decimalMinutes>45.43</gml:decimalMinutes>
</gml:dmsAngle>
```

7.10.3.5 gml:degrees

The **degrees** element allows an integer number of degrees with identification of the angle direction. This element is intended to be used within geographic positions, and has an XML attribute **direction** that can take values

"N" or "S" for Latitude, meaning North or South of the equator;

"E" or "W" for Longitude, meaning East or West of the prime meridian;

"+" or "-" for other angles, in the specified rotational direction from a specified reference direction.

7.10.3.6 gml:decimalMinutes

Decimal number of arc-minutes for use within a degree-minute angular value.

7.10.3.7 gml:minutes

Integer number of arc-minutes for use within a degree-minute-second angular value.

7.10.3.8 gml:seconds

Number of arc-seconds for use within a degree-minute-second angular value.

7.10.3.9 gml:AngleChoiceType

To support the choice of either encoding for angles in a content model, a utility type **gml:AngleChoiceType** is provided.   This element contains another element, either a angle or a dmsAngle. It is declared in the schema as follows:

```
<complexType name="AngleChoiceType">
```

```
  <choice>
    <element ref="gml:angle"/>
    <element ref="gml:dmsAngle"/>
  </choice>
</complexType>
```

In an instance document an element of this type might appear:

```
<my:angle>
  <gml:angle uom="#gradians">95.</gml:simpleAngle>
</my:angle>

<my:angle>
  <gml:dmsAngle>
    <gml:degrees direction="S">35</gml:degrees>
    <gml:decimalMinutes>45.43</gml:decimalMinutes>
  </gml:dmsAngle>
</my:angle>
```

7.10.4 Value Objects schema

7.10.4.1 Introduction

The schema document valueObjects.xsd describing components for generic Values is listed in Annex C. It is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:valueObjects:v3.0

The elements declared in valueObjects.xsd build on elements and types from other GML schemas, in particular _TimeObject from temporal.xsd, _Geometry from geometry.xsd, and the following types from basicTypes.xsd: MeasureType, QuantityListType, CodeType, CategoryListType, BooleanOrNullListType, IntegerOrNullList.

Of particular interest are elements declared in valueObjects.xsd that are the heads of substitution groups, and one named choice group. These can act as variables in the definition of content models, such as Observations, when it is desired to permit alternative value types to occur and where it is useful not to prescribe the actual value type in advance. The members of the groups include quantities, category classifications, Boolean, count, temporal and spatial values, and aggregates of these.

NOTE: The elements declared in this schema are used for **direct representation** of values. Their content models are in general not derived from gml:AbstractGMLType and they do not carry an identifier.

7.10.4.2 Value element hierarchy

The component hierarchy is illustrated in the following UML class diagram. UML generalization relationships are used to indicate XML Schema substitution group and choice group membership. UML composition relationships are used to indicate membership in an XML Schema type content model.

The following relationships are defined in the valueObjects.xsd schema:

- Concrete elements Quantity, Category, Count and Boolean are substitutable for the abstract element _ScalarValue.
- Concrete elements QuantityList, CategoryList, CountList and BooleanList are substitutable for the abstract element _ScalarValueList.
- Concrete element ValueArray is substitutable for the concrete element CompositeValue.
- Abstract elements _ScalarValue and _ScalarValueList, and concrete elements CompositeValue, ValueExtent, CategoryExtent, CountExtent and QuantityExtent are substitutable for abstract element _Value.
- Abstract elements _Value, _TimeObject (from temporal.xsd) and _Geometry (from geometry.xsd), and concrete element Null (from gmlBase.xsd) are all in a choice group named Value, which is used for compositing in CompositeValue and ValueExtent.
- Schemas which need values may use the abstract element _Value in a content model in order to permit any of the _ScalarValue's, _ScalarValueList's, CompositeValue or ValueExtent to occur in an instance, or the named group Value to also permit TimeObject's, Geometry's, and Null's.

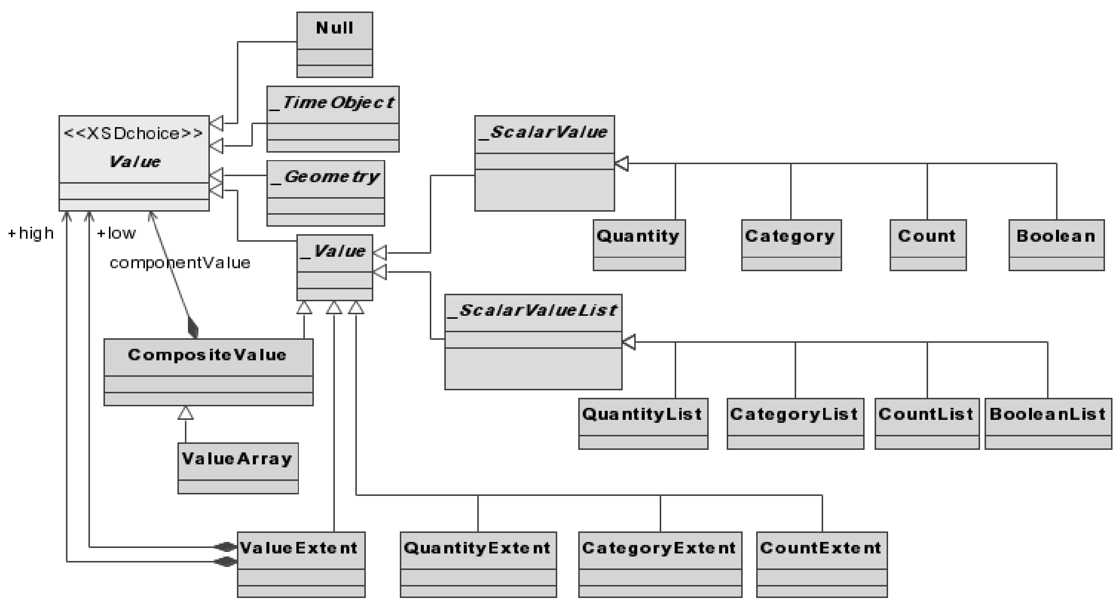


Figure 7.9.4 Substitution groups and composition relationships in the value schema

7.10.4.3 gml:Boolean, gml:BooleanList

For recording a value or list of values from two-valued logic, using the XML Schema boolean type. These elements use the following schema declarations:

```
<element name="Boolean" type="boolean" substitutionGroup="gml:_ScalarValue"/>

<element name="BooleanList" type="gml:booleanOrNullList" substitutionGroup="gml:_ScalarValueList"/>
```

In an instance the following examples may be found:

```
<gml:Boolean>1</gml:Boolean>
<gml:Boolean>false</gml:Boolean>

<gml:BooleanList>1 missing  0 1 1 http://my.big.org/explanations/theDogAteIt 0 1</gml:BooleanList>
```

These examples illustrate the use of the various Boolean values {1, 0, true, false} and also the fact that null values such as "missing" or a URI can be embedded within a list.

7.10.4.4 gml:Category, gml:CategoryList

For recording terms representing a classification. These elements use the following schema declarations:

```
<element name="Category" type="gml:CodeType" substitutionGroup="gml:_ScalarValue"/>

<element name="CategoryList" type="gml:CodeOrNullListType" substitutionGroup="gml:_ScalarValueList"/>
```

A Category has an optional XML attribute **codeSpace**, whose value is a URI which identifies a dictionary, codelist or authority for the term. In an instance the following examples may be found:

```
<gml:Category>good</gml:Category>
<gml:Category codeSpace="http:// my.big.org /dictionaries/rocktypes">Syenite</gml:Category>
```

```
<gml:CategoryList codeSpace="http:// my.big.org /dictionaries/rocktypes">Syenite Granite missing
Tuff</gml:CategoryList>
<gml:CategoryList codeSpace="http:// my.big.org /species">bettong numbat phasogale wallaby possum</
gml:CategoryList>
```

7.10.4.5 gml:Count, gml:CountList

For recording integers representing a rate of occurrence. These elements use the following schema declarations:

```
<element name="Count" type="integer" substitutionGroup="gml:_ScalarValue"/>

<element name="CountList" type="gml:integerOrNullList" substitutionGroup="gml:_ScalarValueList"/>
```

In an instance the following examples may be found:

```
<gml:Count>513</gml:Count>

<gml:CountList>34 56 2 inappplicable 153</gml:CountList>
```

7.10.4.6 gml:Quantity, gml:QuantityList

For recording numeric values with a scale. The content of the element is an amount using the XML Schema type double which permits decimal or scientific notation. These elements use the following schema declarations:

```
<element name="Quantity" type="gml:MeasureType" substitutionGroup="gml:_ScalarValue"/>

<element name="QuantityList" type="gml:MeasureOrNullListType" substitutionGroup="gml:_ScalarValueList"/>
```

An XML attribute **uom** ("unit of measure") is required, whose value is a URI which identifies the definition of a ratio scale or units by which the numeric value must be multiplied, or an interval or position scale on which the value occurs. In an instance the following examples may be found:

```
<gml:Quantity uom="#m">4.32e-4</gml:Quantity>
<gml:QuantityList uom="#C">21. 37. withheld 25.</gml:QuantityList>
```

7.10.4.7 gml:_Value, gml:_ScalarValue, gml:_ScalarValueList

Value is an abstract element which acts as the head of a substitution group which contains _ScalarValue, _ScalarValueList, CompositeValue and ValueExtent, and (transitively) the elements in their substitution groups.

ScalarValue is an abstract element which acts as the head of a substitution group which contains Boolean, Category, Count and Quantity, and (transitively) the elements in their substitution groups.

ScalarValueList is an abstract element which acts as the head of a substitution group which contains BooleanList, CategoryList, CountList and QuantityList, and (transitively) the elements in their substitution groups.

These elements use the following schema declarations:

```
<element name="_Value" abstract="true" substitutionGroup="gml:_Object">
<element name="_ScalarValue" abstract="true" substitutionGroup="gml:_Value"/>
<element name="_ScalarValueList" abstract="true" substitutionGroup="gml:_Value"/>
```

These elements may be used in an application schema as variables, so that in an XML instance document any member of its substitution group may occur.

7.10.4.8 gml:Value

This is a utility choice group which unifies generic Values defined in this schema document with Geometry and Temporal objects and the Measures described above, so that any of these may be used within aggregate Values. This element uses the following schema declaration:

```
<group name="Value">
  <choice>
    <element ref="gml:_Value"/>
    <element ref="gml:_Geometry"/>
    <element ref="gml:_TimeObject"/>
    <element ref="gml:Null"/>
    <element ref="gml:measure"/>
  </choice>
</group>
```

7.10.4.9 gml:valueProperty, gml:valueComponent, gml:valueComponents

Elements that instantiates a GML property which refers to, or contains, a Value or Values. These elements use the following schema declarations:

```
<element name="valueProperty" type="gml:ValuePropertyType"/>

<element name="valueComponent" type="gml:ValuePropertyType"/>

 <complexType name="ValuePropertyType">
  <sequence minOccurs="0">
    <group ref="gml:Value"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
 </complexType>

 <element name="valueComponents" type="gml:ValueArrayPropertyType"/>

 <complexType name="ValueArrayPropertyType">
  <sequence>
    <group ref="gml:Value" maxOccurs="unbounded"/>
  </sequence>
 </complexType>
```

Note that both ValuePropertyType and ValueArrayPropertyType have the group named “Value” as their content. This means that any of the elements in the Value choice group, or in the substitution groups of the members of the choice group can occur as the content of a valueProperty.

The valueProperty element is a utility element for general use. The valueComponent and valueComponents elements are specifically used in compositing. The Values contained within a valueComponents element should be homogeneous.

7.10.4.10 gml:CompositeValue

CompositeValue is an aggregate value built from other Values using the Composite pattern. It contains zero or an arbitrary number of valueComponent elements, and zero or one valueComponents elements. It may be used for strongly coupled aggregates (vectors, tensors) or for arbitrary collections of values. This element uses the following schema declarations:

```
<element name="CompositeValue" type="gml:CompositeValueType" substitutionGroup="gml:_Value"/>
```

```
<complexType name="CompositeValueType">
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:valueComponent" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:valueComponents" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

In an instance a CompositeValue may appear as in the following examples:

```
<gml:CompositeValue>
  <gml:valueComponent>
    <gml:QuantityList uom="#C">21. 37. withheld 25.</gml:QuantityList>
  </gml:valueComponent>
  <gml:valueComponent>
    <gml:Category>good</gml:Category>
  </gml:valueComponent>
  <gml:valueComponent>
    <gml:Null>template</gml:Null>
  </gml:valueComponent>
  <gml:valueComponents>
    <gml:Point srsName="epsg:4326"><gml:pos>171. -32.</gml:pos></gml:Point>
    <gml:Point srsName="epsg:4326"><gml:pos>170. -35.</gml:pos></gml:Point>
    <gml:Point srsName="epsg:4326"><gml:pos>174. -37.</gml:pos></gml:Point>
  </gml:valueComponents>
</gml:CompositeValue>
```

```
<gml:CompositeValue>
  <gml:valueComponents>
    <gml:Point srsName="epsg:4326"><gml:pos>-67.563 -13.834</gml:pos></gml:Point>
    <gml:Quantity uom="#km">632.</gml:Quantity>
    <gml:TimeInstant><gml:timePosition>1994-06-09T00:33:16.4</gml:timePosition></gml:TimeInstant>
    <gml:Quantity uom="#mom">-1.00</gml:Quantity>
    <gml:Quantity uom="#mom">0.92</gml:Quantity>
    <gml:Quantity uom="#mom">0.09</gml:Quantity>
    <gml:Quantity uom="#mom">-1.69</gml:Quantity
    <gml:Quantity uom="#mom">-0.09</gml:Quantity>
    <gml:Quantity uom="#mom">-0.37</gml:Quantity>
  </gml:valueComponents>
</gml:CompositeValue>
```

7.10.4.11 gml:ValueArray

A Value Array is used for homogeneous arrays of primitive and aggregate values.

The member values may be scalars, composites, arrays or lists. This element uses the following schema declarations:

```
<element name="ValueArray" type="gml:ValueArrayType" substitutionGroup="gml:CompositeValue">
  <annotation>
    <appinfo>
      <sch:pattern name="Check either codeSpace or uom not both">
```



```
        <sch:rule context="gml:ValueArray">
          <sch:report test="@codeSpace and @uom">ValueArray may not carry both a reference to a codeSpace
and a uom</sch:report>
        </sch:rule>
      </sch:pattern>
      <sch:pattern name="Check components are homogeneous">
        <sch:rule context="gml:ValueArray">
          <sch:assert test="count(gml:valueComponent/*) = count(gml:valueComponent/*[name() =
name(../../gml:valueComponent[1]/*[1])])">All components of <sch:name/> must be of the same type</sch:assert>
          <sch:assert test="count(gml:valueComponents/*) = count(gml:valueComponents/*[name() =
name(../*[1])])">All components of <sch:name/> must be of the same type</sch:assert>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>

<complexType name="ValueArrayType">
  <complexContent>
    <extension base="gml:CompositeValueType">
      <attributeGroup ref="gml:referenceSystem"/>
    </extension>
  </complexContent>
</complexType>

<attributeGroup name="referenceSystem">
  <attribute name="codeSpace" type="anyURI" use="optional"/>
  <attribute name="uom" type="anyURI" use="optional"/>
</attributeGroup>
```

ValueArray has the same content model as CompositeValue, but the member values must be homogeneous. The element declaration contains a Schematron constraint which expresses this restriction precisely. Since the members are homogeneous, the referenceSystem (**uom**, **codeSpace**) may be specified on the ValueArray itself and inherited by all the members if desired.

The ValueArray element may appear in instances as follows.

In the first example a set of Points are each the value of a valueComponent property. One of the values is provided by-reference, using the standard xlink:href syntax:

```
<gml:ValueArray>
  <gml:valueComponent>
    <gml:Point srsName="epsg:4326">
      <gml:pos>171. -32.</gml:pos>
    </gml:Point>
  </gml:valueComponent>
  <gml:valueComponent>
    <gml:Point srsName="epsg:4326">
      <gml:pos>170. -35.</gml:pos>
    </gml:Point>
  </gml:valueComponent>
  <gml:valueComponent xlink:href="http://my.big.org/locations/points/point456"/>
</gml:ValueArray>
```

In the second example a set of Quantities are contained within a valueComponents property. One of the values is not available, indicated by a Null:

```
<gml:ValueArray>
 <gml:valueComponents>
  <gml:Quantity uom="#C">21.</gml:Quantity>
  <gml:Quantity uom="#C">37.</gml:Quantity>
  <gml:Null>missing</gml:Null>
 </gml:valueComponents>
</gml:ValueArray>
```

Note that a_ScalarValueList is usually preferred for arrays of Scalar Values since this is a more efficient encoding. The information in the previous example can be expressed:

```
<gml:QuantityList uom="#C">21. 37. missing</gml:QuantityList>
```

However, if the values of the components are not scalars, then the explicit form is required.

7.10.4.12 Typed ValueExtents: gml:CategoryExtent, gml:CountExtent, gml:QuanttyExtent

Three elements are provided for typed value extents, for categories, counts and quantities. Their content models are defined by restricting the relevant scalar list types to contain exactly two items as follows:

```
<element name="CategoryExtent" type="gml:CategoryExtentType" substitutionGroup="gml:_Value"/>

 <complexType name="CategoryExtentType">
  <simpleContent>
   <restriction base="gml:CodeOrNullListType">
    <length value="2"/>
   </restriction>
  </simpleContent>
 </complexType>

 <element name="CountExtent" type="gml:CountExtentType" substitutionGroup="gml:_Value"/>

 <simpleType name="CountExtentType">
  <restriction base="gml:integerOrNullList">
   <length value="2"/>
  </restriction>
 </simpleType>

 <element name="QuantityExtent" type="gml:QuantityExtentType" substitutionGroup="gml:_Value"/>

 <complexType name="QuantityExtentType">
  <simpleContent>
   <restriction base="gml:MeasureOrNullListType">
    <length value="2"/>
   </restriction>
  </simpleContent>
 </complexType>
```

A gml:QuantityExtent element or another element using this type will contain two values and a scale as follows:

```
<gml:QuantityExtent uom="#mm">0. 9.5</gml:QuantityExtent>
```

An element of gml:CategoryExtentType is useful if the codeSpace defines a set of ordered terms, for example:

```
<my:AgeRange codeSpace="http://iugg.org/geologicalTimePeriods">Cambrian Devonian</my:AgeRange>
```

Any value extent may describe a single-ended interval by using a Null value for one of the limits, for example:

```
<gml:CountExtent>53 inapplicable</gml:CountExtent>
```

describes the integers starting with 53.

7.10.4.13 gml:BooleanPropertyType, gml:CategoryPropertyType, gml:CountPropertyType, gml:QuantityPropertyType

A set of utility types are provided for properties whose content is a specific member of the gml:_ScalarValue substitution group. Their definitions follow the same pattern, as exemplified by the definition of gml:BooleanPropertyType:

```
<xs:complexType name="BooleanPropertyType">
  <xs:complexContent>
    <xs:restriction base="gml:ValuePropertyType">
      <xs:sequence >
        <xs:element ref="gml:Boolean" minOccurs="0"/>
      </xs:sequence>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
```

.

7.11 Directions

7.11.1 Direction schema

The direction schema, direction.xsd provides the GML Application Schema developer with a standard property element to describe direction, and associated objects that can be used to express orientation, direction, heading, bearing or other directional aspects of geographic features. The schema is listed in Annex C. It is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:direction:v3.0

7.11.1.1 gml:direction

The property gml:direction is intended as a property to be assigned to features defined in a GML application schema. It is declared in the schema as follows:

```
<element name="direction" type="gml:DirectionPropertyType"/>

<complexType name="DirectionPropertyType">
  <annotation>
    <documentation/>
  </annotation>
  <choice>
    <element ref="gml:DirectionVector"/>
    <element ref="gml:CompassPoint"/>
    <element name="DirectionKeyword" type="gml:CodeType"/>
    <element name="DirectionString" type="gml:StringOrRefType"/>
  </choice>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

The various kinds of direction specifications follow.

7.11.1.2 gml:DirectionVector

Director vectors are specified by providing components of a vector or two angles as follows:

```
<element name="DirectionVector" type="gml:DirectionVectorType"/>

<complexType name="DirectionVectorType">
  <choice>
    <element ref="gml:vector"/>
    <sequence>
      <element name="horizontalAngle" type="gml:AngleType"/>
      <element name="verticalAngle" type="gml:AngleType"/>
    </sequence>
  </choice>
</complexType>
```

The gml:vector element is described in clause 7.5.1.4. This form may appear in an data instance as follows:

```
<gml:direction>
  <gml:DirectionVector>
    <gml:horizontalAngle uom="#degreesEastOfNorth">45.0</gml:horizontalAngle>
    <gml:verticalAngle uom="#degreesDown">0.0</gml:verticalAngle>
  </gml:DirectionVector>
</gml:direction>
```

Angles are specified via gml:AngleType whose content model is defined in clause 7.10.3.3. This form may appear in an data instance as follows:

```
<gml:direction>
  <gml:DirectionVector>
    <gml:horizontalAngle uom="#degreesEastOfNorth">45.0</gml:horizontalAngle>
    <gml:verticalAngle uom="#degreesDown">0.0</gml:verticalAngle>
  </gml:DirectionVector>
</gml:direction>
```

7.11.1.3 gml:CompassPoint

A compass point is specified by a simple enumeration string type. The gml:CompassPoint element is declared in the schema as follows:

```
  <element name="CompassPoint" type="gml:CompassPointEnumeration"/>

  <simpleType name="CompassPointEnumeration">
    <restriction base="string">
      <enumeration value="N"/>
      <enumeration value="NNE"/>
      <enumeration value="NE"/>
      <enumeration value="ENE"/>
      <enumeration value="E"/>
      <enumeration value="ESE"/>
      <enumeration value="SE"/>
      <enumeration value="SSE"/>
      <enumeration value="S"/>
      <enumeration value="SSW"/>
      <enumeration value="SW"/>
      <enumeration value="WSW"/>
      <enumeration value="W"/>
      <enumeration value="WNW"/>
      <enumeration value="NW"/>
      <enumeration value="NNW"/>
    </restriction>
  </simpleType>
```

This form may appear in an data instance as follows:

```
<gml:direction>
  <gml:CompassPoint>WNW</gml:CompassPoint>
</gml:direction>
```

7.11.1.4 Text Based Directions: gml:DirectionKeyword, gml:DirectionString

Two elements to contain text-based descriptions of direction are provided.

If the direction is specified using a term from a list, gml:KeyWord should be used, and the list indicated using the value of the codeSpce attribute. This form may appear in an data instance as follows:

```
<gml:direction>
  <gml:DirectionKeyword codeSpace="http://my.big.org/terms/direction">onshore</gml:DirectionKeyword>
</gml:direction>
```

If the direction is decribed in prose, gml:TextDirection should be used, allowing the value to be included inline or by reference. This form may appear in an data instance as follows:

```
<gml:direction>
  <gml:DirectionString>Towards the lighthouse</gml:DirectionString>
</gml:direction>
```

```
<gml:direction>
  <gml:DirectionString xlink:href="http://my.big.org/logbook/20021127/paragraph6"/>
</gml:direction>
```

7.12 Observations

A GML observation models the act of observing, often with a camera, a person or some form of instrument ("an act of recognizing and noting a fact or occurrence often involving measurement with instruments"). A GML observation is considered to be a GML feature with a time at which the observation took place, and with a value for the observation. This covers a broad range of cases, from a tourist photo (not the photo but the act of taking the photo), to images acquired by space borne sensors or the measurement of a temperature 5 meters below the surfaces of a Lake. See also Clause 7.9. (measures.xsd).

The basic structures introduced in this schema are intended to serve as the foundation for more comprehensive schemas for scientific, technical and engineering measurement schemas.

7.12.1 Observation schema

Observations are described in the schema, observations.xsd. The schema is listed in Annex C. It is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:observation:v3.0

This schema describes two kinds of observations, gml:Observation and gml:DirectedObservation.

7.12.1.1 gml:Observation

The gml:Observation element is declared in the schema as follows:

```
<element name="Observation" type="gml:ObservationType" substitutionGroup="gml:_Feature"/>
```



```
<complexType name="ObservationType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:timeStamp"/>
        <element ref="gml:using" minOccurs="0"/>
        <element ref="gml:target" minOccurs="0"/>
        <element ref="gml:resultOf"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

The content model is a straightforward extension of gml:AbstractFeatureType it automatically has gml:metadataProperty, gml:description, gml:name, gml:location and gml:boundedBy properties.

The gml:timeStamp element is declared in clause 7.8.2.5. In this context it describes the time of the observation. Note that this can be a time instant or a time period (interval).

7.12.1.2 gml:using

The gml:using property contains or points to a description of a sensor, instrument or procedure used for the observation. It is declared in the schema as follows:

```
<element name="using" type="gml:FeaturePropertyType"/>
```

7.12.1.3 gml:target

The gml:target property contains or points to the specimen, region or station which is the object of the observation. This property element is declared in the schema as follows:

```
<element name="target" type="gml:TargetPropertyType"/>
<element name="subject" type="gml:TargetPropertyType" substitutionGroup="gml:target">

<complexType name="TargetPropertyType">
  <annotation>
    <documentation>Container for an object representing the target or subject of an
observation.</documentation>
  </annotation>
  <choice minOccurs="0">
    <element ref="gml:_Feature"/>
    <element ref="gml:_Geometry"/>
  </choice>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

This property is particularly useful for remote observations, such as photographs, where the gml:location property might apply to the location of the camera or the location of the field of view, and thus may be ambiguous.

The subject element is provided as a convenient synonym for target. This is the term commonly used in phtotography.

7.12.1.4 gml:resultOf

The gml:resultOf property contains or carries a reference to the result of the observation. It is declared in the schema as follows:

```
  <element name="resultOf" type="gml:AssociationType">
```

Examples of gml:ObservationSome examples of simple observations are as follows:

```
<gml:Observation>
  <gml:location xlink:href="http://www.environment.org/stations/l456"/>
  <gml:timeStamp>
    <gml:TimeInstant>
      <gml:timePosition>2002-11-12T09:12:00</gml:timePosition>
    </gml:TimeInstant>
  </gml:timeStamp>
  <gml:using xlink:href="http://www.my.org/sensors/thermometer4"/>
  <gml:target xlink:href="http://www.environment.org/stations/l456"/>
  <gml:resultOf>
    <gml:Quantity uom="#C">18.4</gml:Quantity>
  </gml:resultOf>
</gml:Observation>
```

```
<gml:Observation>
  <gml:location xlink:href="http://www.tourist.org/lookouts/platform4"/>
  <gml:timeStamp>
    <gml:TimeInstant>
      <gml:timePosition>2002-11-12T09:12:00</gml:timePosition>
    </gml:TimeInstant>
  </gml:timeStamp>
  <gml:using xlink:href="http://www.my.org/cameras/leica2"/>
  <gml:target xlink:href="http://www.tourist.org/sights/mountain3"/>
  <gml:resultOf xlink:href="http://www.my.org/photos/landscape1.jpg"/>
</gml:Observation>
```

```
<gml:Observation>
  <gml:location>
    <gml:LocationString>Home</gml:LocationString>
  </gml:location>
  <gml:timeStamp>
    <gml:TInstant>
      <gml:tPosition>2002-10-25T11:37:25</gml:tPosition>
    </gml:TInstant>
  </gml:timeStamp>
  <gml:target xlink:href="http://www.people.org/kids/abby"/>
  <gml:resultOf xlink:href="myDaughtersPortrait.jpg"/>
</gml:Observation>
```

7.12.1.5 gml:DirectedObservation

A DirectedObservation is the same as an observation except that it adds an additional direction property. This is the direction in which the observation was acquired. Clearly this applies only to certain types of observations such as visual observations by people, or observations obtained from terrestrial cameras.

```
  <xsd:element name="DirectedObservation" type="gml:DirectedObservationType"
```

```
    substitutionGroup="gml:_Feature"/>
  <xsd:complexType name="DirectedObservationType">
    <xsd:complexContent>
      <xsd:extension base="gml:ObservationType">
        <xsd:sequence>
          <xsd:element ref="gml:direction"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
```

Example: DirectedObservation

```
<gml:DirectedObservation>
  <gml:timeStamp>
    <gml:TimeInstant>
      <gml:timePosition>2002-11-12T09:12:00</gml:timePosition>
    </gml:TimeInstant>
  </gml:timeStamp>
 <gml:using xlink:href="http://www.my.org/cameras/leica2"/>
  <gml:target xlink:href="http://www.tourist.org/sights/mountain3"/>
 <gml:resultOf xlink:href="http://www.my.org/photos/landscape1.jpg"/>
  <gml:direction>
    <gml:CompassPoint>NW</gml:CompassPoint>
  </gml:direction>
</gml:DirectedObservation>
```

7.13 Coverages

7.13.1 The coverage model and representations

This clause defines the GML encoding for coverages and is in agreement with the conceptual model outlined in ISO 19123 and in the OGC Abstract Specification, Topic 6.

ISO 19123 provides a definition:

> *Coverages support mapping from a spatiotemporal domain to attribute values where attribute types are common to all geographic positions within the spatiotemporal domain. A spatiotemporal domain consists of a collection of direct positions in a coordinate space. Examples of coverages include rasters, triangulated irregular networks, point coverages, and polygon coverages. Coverages are the prevailing data structures in a number of application areas, such as remote sensing, meteorology, and bathymetric, elevation, soil, and vegetation mapping.*

The information describing a coverage is conventionally represented in one of two ways:

i. As a set of discrete location-value pairs.

ii. As a description of the spatio-temporal domain (multi-geometry, grid) and a description of the set of values from the range, together with a method or rule (which may be implicit) that assigns a value from the range set to each position within the domain.

The first method only applies to domains that are partitioned into discrete components. This representation may be realised in GML as a **homogeneous feature collection** (i.e. all the features have the same set of properties), where the set of locations from the features compose the domain (remember: gml:location may refer to any geometry, not just points), and the set of property values compose the range. The mapping from domain to range is trivial: the properties on each feature are assigned to the location of that feature. For coverages whose domain is composed of a large set of locations this explicit representation may, however, be bulky.

The second method is more flexible in a number of ways.

a) Since the domain and range are homogeneous sets, there may be efficiences in the representation of either or both domain and range

b) The values in the range may be represented in analytic form rather than as discrete explicit values, which is also related to the fact that

c) When the attribute values vary continuously across the domain, a functional form covering the complete domain is required to be able to provide values of the range at arbitrary locations. The function typically involves interpolation, possibly using a process model.

The first representation is typically used during data collection where a set or properties relating to a single location are managed together, or update of a datastore where only a small number of features are manipulated at one time. The second representation is more suitable for analysis, where spatio-temporal patterns and anomalies within a specific property are of interest.

It is the second method, using a functional map over the whole domain, that is the subject of the GML coverage encoding.

7.13.2 Formal description of a coverage

A coverage incorporates a mapping from a spatiotemporal domain to a range set, the latter providing the set in which the attribute values live. The range set can be an arbitrary set including discrete lists, integer or floating point ranges, and multi-dimensional vector spaces. This conceptual model of a coverage is described in Figure 7.13-1.

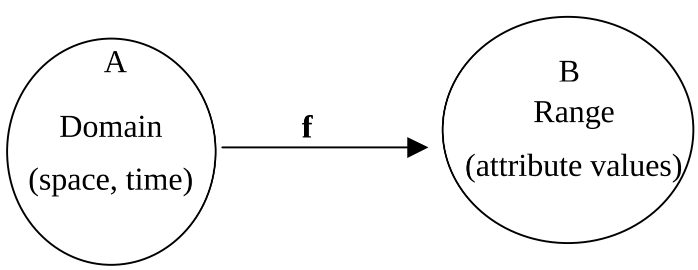


Figure 7.13.1 Conceptual Model of a Coverage

A coverage can be viewed as the subset of the graph of the coverage function $f{:}A \longrightarrow B$, that is as the set of ordered pairs

$$\{(x, f(x)) \mid \text{where } x \text{ is in } A\}$$

This view is especially applicable to the GML encoding of a coverage. In some cases the domain set A is partitioned into a collection of subsets (typically a disjoint collection) A = UA_i and the function f is constant on each A_i. For a spatial domain, the A_i are geometry elements, hence the coverage can be viewed as a collection of (geometry,value) pairs, where the value is an element of the range set. If the spatial domain A is a topological space then the coverage can be viewed as a collection of (topology,value) pairs, where the topology element in the pair is a topological n-chain (in GML terms this is a gml:TopoPoint, gml:TopoCurve, gml:TopoSurface or gml:TopoSolid).

7.13.3 Coverage in GML

The basic model of a coverage is shown as a UML class diagram in Figure 7.13-2.

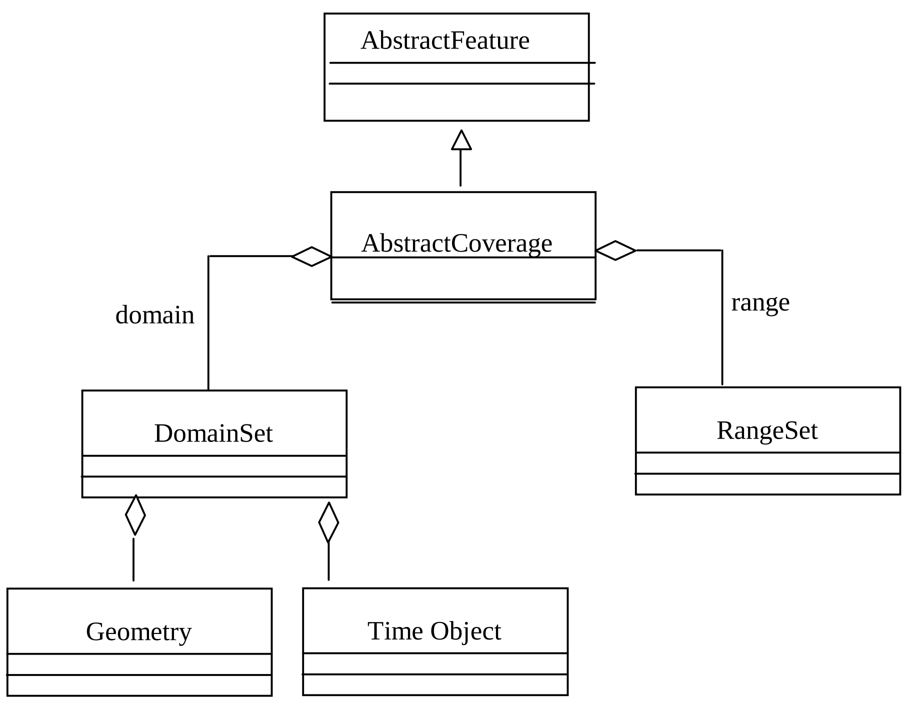


Figure 7.13-2 UML Class Diagram for Coverage

A coverage is implemented as a GML feature. We can thus speak of a “temperature distribution feature”, or a “remotely sensed image feature”, or a “soil distribution feature”.

As is the case for any GML object, a coverage object may also be the value of a property of a feature. For example, the temperature distribution might be a property of a city feature, so a description of the city of Ottawa might be represented in GML as:

feature

feature (coverage)

```
<abc:City gml:id = "Ottawa">
                        <abc:population>500000</abc:population>
                        <abc:temperatureDistribution>
                                    <abc:TemperatureCoverage> … </abc:TemperatureCoverage>
                        </abc:temperatureDistribution>
</abc:City>
```

Coverages in GML 3.0 are supported by two schemas,

- coverage.xsd
- grids.xsd.

Coverages.xsd provides the basic GML 3.0 coverage model. Grids.xsd provides grid geometry structures that are used in the description of gridded coverages but which could be employed for other applications.

Future releases of GML will provide other geometries and temporal complexes for use in coverages.

7.13.4 Grids schema

An implicit geometry is one in which the items of the geometry do not explicitly appear in the encoding. Instead, a compact notation records a set of parameters, and a set of objects may be generated using a rule with these parameters.

The schema grids.xsd provides some grid geometries that are used in the description of gridded coverages and other applications. The schema is listed in Annex C. It is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:grids:v3.0

In GML 3.0 two grid structures are defined, namely gml:Grid and gml:RectifiedGrid.

7.13.4.1 Unrectified Grid (gml:Grid)

The gml:Grid element is defined in the schema as follows:

```
<element name="Grid" type="gml:GridType" substitutionGroup="gml:_ImplicitGeometry"/>

  <complexType name="GridType">
    <complexContent>
      <extension base="gml:AbstractGeometryType">
        <sequence>
          <element name="limits" type="gml:GridLimitsType"/>
          <element name="axisName" type="string" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="dimension" type="positiveInteger" use="required"/>
```

```
    </extension>
  </complexContent>
</complexType>
```

The gml:Grid implicitly defines an unrectified grid, which is a network composed of two or more sets of equally spaced parallel lines in which the members of each set intersect each other. . The region of interest within the grid is given in terms of its **limits**, being the grid coordinates of diagonally opposed corners of a rectangular region. An **axisName** is provided for each of the **dimension** axes of the grid, the number of which – normally 2 or 3 –is given as the value of an attribute.

In GML3 the **limits** element contains a single GridEnvelope, according to the following schema definitions:

```
<complexType name="GridLimitsType">
  <sequence>
    <element name="GridEnvelope" type="gml:GridEnvelopeType"/>
  </sequence>
</complexType>

<complexType name="GridEnvelopeType">
  <sequence>
    <element name="low" type="gml:integerList"/>
    <element name="high" type="gml:integerList"/>
  </sequence>
</complexType>
```

The low and high elements are each integerLists, which are coordinate tuples, the coordinates being measured as offsets from the origin of the grid along each axis, of the diagonally opposing corners of a “rectangular” region of interest.

The following example illustrates a simple Grid.

```
<gml:Grid dimension="2">
  <gml:limits>
    <gml:GridEnvelope>
      <gml:low>0 0</gml:low>
      <gml:high>3 3</gml:high>
    </gml:GridEnvelope>
  </gml:limits>
  <gml:axisName>x</gml:axisName>
  <gml:axisName>y</gml:axisName>
</gml:Grid>
```

In this example the Grid has posts (points) at locations (0,0), (0,1),(1,0),(1,1) through to (4,4).

7.13.4.2 Rectified Grid (gml:RectifiedGrid)

A rectified grid is a kind of grid in which the points of the grid have geometric locations. It is defined by specifying the position (in some geometric space) of the grid "origin" and of the vectors that specify the post locations. The RectifiedGrid element is declared in the schema as follows:

```
<element name="RectifiedGrid" type="gml:RectifiedGridType" substitutionGroup="gml:Grid"/>

  <complexType name="RectifiedGridType">
    <complexContent>
      <extension base="gml:GridType">
        <sequence>
          <element name="origin" type="gml:PointPropertyType"/>
          <element name="offsetVector" type="gml:VectorType" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
```

Note that the grid limits (post indexes) and axis name properties are inherited from the gml:GridType and that the gml:RectifiedGrid adds an origin (contains a gml:Point) and a set of offset vectors specified using gml:VectorType as described in clause 7.5.1.4.

Figure 7.13-3 shows the geometry of the RectifiedGrid.

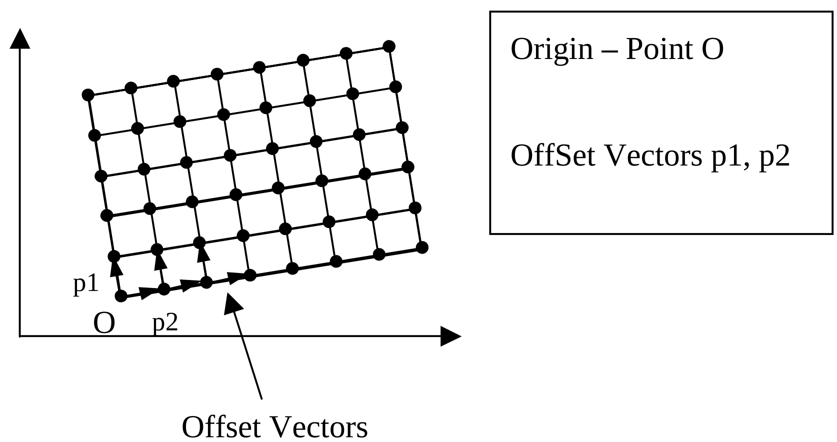


Figure 7.13-3 RectifiedGrid Geometry

An example RectifiedGrid is as follows:

```
<gml:RectifiedGrid dimension="2">
  <gml:limits>
    <gml:GridEnvelope>
      <gml:low>1 1</gml:low>
      <gml:high>3 3</gml:high>
    </gml:GridEnvelope>
  </gml:limits>
  <gml:axisName>u</gml:axisName>
  <gml:axisName>v</gml:axisName>
  <gml:origin>
    <gml:Point gml:id="palindrome">
      <gml:coordinates>1.2,3.3,2.1</gml:coordinates>
    </gml:Point>
  </gml:origin>
  <gml:offsetVector>1,2,3</gml:offsetVector>
  <gml:offsetVector >2,1,0</gml:offsetVector>
</gml:RectifiedGrid>
```

Note that in this example the RectifiedGrid starts at integer offset 1 1 relative to the origin as shown in Figure 7.13-4.

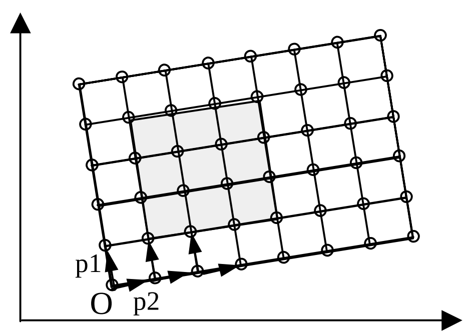


Figure 7.13-4 RectifiedGrid with non-zero low limit

7.13.5 Coverage schema

The coverage.xsd schema is listed in Annex C. It is identified by the following location-independent name (using URN syntax):

urn:opengis:specification:gml:schema-xsd:coverage:v3.0

7.13.5.1 gml:AbstractCoverageType, gml:_Coverage

The base type for coverages is AbstractCoverageType, defined in the schema as follows:

```
<complexType name="AbstractCoverageType" abstract="true">
  <complexContent>
```

```
    <extension base="gml:AbstractFeatureType">
     <sequence>
       <element ref="gml:domainSet"/>
       <element ref="gml:rangeSet"/>
       <element ref="gml:coverageFunction" minOccurs="0"/>
     </sequence>
     <attribute name="dimension" type="positiveInteger" use="optional"/>
    </extension>
  </complexContent>
 </complexType>
```

NOTE: gml:AbstractCoverageType is derived by extension from AbstractFeatureType, so any coverage whose content model is derived from this type is a GML feature.

The basic elements of a coverage can be seen in this content model: the coverage contains domainSet, rangeSet and coverageFunction properties. The gml:domainSet property describes the domain of the coverage. The rangeSet property describes the range of the coverage, and the coverageFunction describes the mapping “f” as shown in Figure 7.12.1.

The abstract element gml:_Coverage is declared as follows:

```
 <element name="_Coverage" type="gml:AbstractCoverageType" abstract="true"
substitutionGroup="gml:_Feature"/>
```

This element serves as the head of a substitution group which may contain any coverage whose type is derived from gml:AbstractCoveragetype. It may act as a variable in the definition of content models where it is required to permit any coverage to be valid.

7.13.5.2 gml:domainSet

The Domain Set describes the spatio-temporal region of interest, within which the coverage is defined. Its content model is given by gml:DomainSetType which is defined as follows:

```
 <element name="domainSet" type="gml:DomainSetType"/>

 <complexType name="DomainSetType">
  <choice>
    <element ref="gml:_Geometry"/>
    <element ref="gml:_TObject"/>
  </choice>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
 </complexType>
```

The value of the domain is thus a choice between a gml:_Geometry and a gml:_TimeObject. In the instance these abstract elements will normally be substituted by a geometry complex or temporal complex, to represent spatial coverages and time-series, respectively.

NOTE: Following the ISO 19100 and OGC abstract specifications, GML 3.0 does not support combined spatial-temporal domains.

The presence of the gml:AssociationAttributeGroup means that domainSet follows the usual property model and can use of the xlink:href attribute to point to the Domain Set, as an alternative to describing the domain inline.

7.13.5.3 gml:rangeSet

The Range Set contains the values of the coverage (sometimes called the attribute values). Its content model is given by gml:RangeSetType which is defined as follows::

```
<element name="rangeSet" type="gml:RangeSetType"/>

 <complexType name="RangeSetType">
 <choice>
  <element ref="gml:_Value"/>
  <element ref="gml:DataBlock"/>
  <element ref="gml:File"/>
 </choice>
  </complexType>
```

This content model supports a structural description of the Range. The semantic information describing the range set is embedded using a uniform method, as part of the explicit values, or as a template value accompanying the representation using gml:DataBlock and gml:File.

The gml:_Value is defined in the schema values.xsd. See clause 7.9. In order to satisfy the requirement that the values in the range of a coverage are homogeneous, only certain members of the gml:_Value substitution group are permitted to occur. These are

1. a (set of?) gml:ValueArray, in which the members are homogeneously typed values
2. a (set of) member(s) of the gml:_ScalarValueList substitution group
3. a gml:ValueCollection whose members are a set of gml:ValueArray or members of the gml:_ScalarValueList substitution group (probably an unnecessary container)

7.13.5.4 gml:DataBlock

gml:DataBlock describes the Range as a block of text encoded values like a Common Separated Value (CSV) representation. The content model is as follows:

```
  <element name="DataBlock" type="gml:DataBlockType"/>

 <complexType name="DataBlockType">
 <sequence>
  <element ref="gml:rangeParameters"/>
  <element ref="gml:tupleList"/>
 </sequence>
  </complexType>
```

The meaning of the range set is determined by its parameterization, one can also look upon this parameterization as a description of the range set attributes. The range set parameterization is described by the property gml:rangeParameters.

7.13.5.5 gml:rangeParameters

gml:rangeParameters is declared int eh schema as follows:

```
<element name="rangeParameters" type="gml:RangeParametersType"/>

<complexType name="RangeParametersType">
  <sequence>
    <element ref="gml:_Value"  minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </complexType>
```

gml:RangeParameterType is a framework for the description of the range parameters each of which is a gml:_Value, as described in clause 7.10.4. Specific range parameters are defined through the creation of a GML Application Schema that provides elements that are substitutable for gml:_Value.

7.13.5.6 gml:tupleList

The gml:tupleList property is declared as follows:

```
<element name="tupleList" type="gml:CoordinatesType"/>
```

gml:CoordinatesType is described in clause 7.3.2.9. It consists of a list of coordinate tuples, with each coordinate tuple separated by the ts or tuple separator (by default this is whitespace), and each coordinate in the tuple by the cs or coordinate separator (by default this is a comma).

An example of a set of pairs of temperature and pressure observations might be recorded in a gml:DataBlock as follows:

```
  <gml:DataBlock>
    <gml:rangeParameters>
      <gml:CompositeValue>
        <gml:valueComponent>
          <my:Temperature uom="#C"/>
        </gml:valueComponent>
        <gml:valueComponent>
          <my:Pressure uom="#kPa"/>
        </gml:valueComponent>
      </gml:CompositeValue>
    </gml:rangeParameters>
    <gml:tupleList>3,101.2 5,101.3 7,101.4 11,101.5  13,101.6  17,101.7  19,101.7  23,101.8  29,101.9
31,102.0  37,102.1  41,102.2  43,102.3  47,102.4  53,102.5  59,102.6</gml:tupleList>
  </gml:DataBlock>
```

where my:Temperature and my:Pressure are elements defined in a local schema, using gml:MeasureType.

7.13.5.7 gml:File

For efficiency reasons, GML 3.0 also provides a means of encoding the range set as a binary file. This is recorded using the gml:File element, which is declared in the schema as follows:

```
<element name="File" type="gml:FileType"/>

<complexType name="FileType">
  <sequence>
    <element ref="gml:rangeParameters"/>
    <element name="fileName" type="anyURI"/>
    <element name="fileStructure" type="gml:FileValueModelType"/>
    <element name="mimeType" type="anyURI" minOccurs="0"/>
    <element name="compression" type="anyURI" minOccurs="0"/>
  </sequence>
</complexType>
```

In this version of the coverage encoding, the values of the coverage (attribute values in the Range set) are transmitted in a binary file that is referenced from the XML structure described by gml:FileType. The binary file is referenced by the fileName property that is an anyURI. This means that the binary file can be located remotely from the referencing GML instance. This can support, for example, both an http reference and a SOAP attachment.

The compression property points to a definition of a compression algorithm through an anyURI. This may be a retrievable, computable definition or simply a reference to an unambiguous name for the compression method.

The mime type property points to a definition of the file mime type.

The fileStructure property is defined by the gml:FileValueModelType. This is simple enumerated type restriction on string. The only value supported in GML 3.0 is “Record Interleaved”. Additional values may be supported in future releases of GML. Note further that all values must be enclosed in a single file. Multi-file structures for values are not supported in GML 3.0.

The semantics of the Range set is described as above using gml:rangeParameters.

Example: Binary File Encoding:

```
<gml:File>
  <gml:rangeParameters>
    <gml:CompositeValue>
      <gml:valueComponent>
        <my:Temperature uom="#C"/>
      </gml:valueComponent>
      <gml:valueComponent>
        <my:Pressure uom="#kPa"/>
      </gml:valueComponent>
    </gml:CompositeValue>
  </gml:rangeParameters>
  <gml:fileName>http://www.somedata.org/temperature_pressure.dat</gml:fileName>
  <gml:fileStructure>Record Interleaved</gml:fileStructure>
</gml:File>
```

The referenced file structure must be as follows:

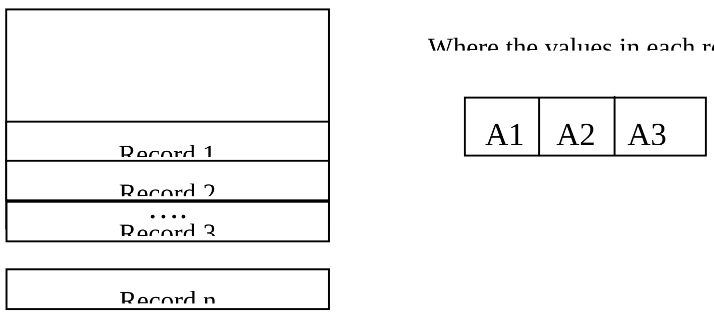


Figure 7.12.5 File Record Structure or Coverage File

Note that if any compression algorithm is applied, the structure above applies only to the pre-compression or post-decompression structure of the file.

Note that the fields within a record match the valueMembers of the ValueCollection in document order.

7.13.5.8 gml:coverageFunction

This clause describes the Coverage Function (gml:coverageFunction) property, that is, the mapping “f” (see Figure 7.12.1.) from the domain to the range of the coverage. The content model for the coverage function is given by:

```
<element name="coverageFunction" type="gml:CoverageFunctionType"/>

<complexType name="CoverageFunctionType">
  <choice>
    <element ref="gml:MappingRule"/>
    <element ref="gml:GridFunction"/>
  </choice>
</complexType>
```


Note that the value of the CoverageFunction is one of gml:MappingRule and gml:GridFunction.

7.13.5.9 gml:MappingRule

gml:MappingRule provides a formal (e.g. MathML) or informal (e.g. free text) description of the coverage function, per:

```
<element name="MappingRule" type="gml:StringOrRefType"/>
```

The mapping rule may be defined as an in-line string or via a remote reference through xlink:href.

7.13.5.10 gml:GridFunction

gml:GridFunction provides an explicit mapping rule for grid geometries, i.e. the domain must be a geometry of type grid. It describes the mapping of grid posts (discrete point grid coverage) or grid cells (discrete surface coverage) to the values in the RangeSet. The content model is as follows:

```
<element name="GridFunction" type="gml:GridFunctionType"/>

 <complexType name="GridFunctionType">
  <sequence>
   <element name="sequenceRule" type="gml:SequenceRuleType" minOccurs="0"/>
   <element name="startPoint" type="gml:integerList" minOccurs="0">
  </sequence>
 </complexType>
```

The startPoint is the index position of a point in the grid that is mapped to the first point in the Range Set (this is also the index position of the first grid post*). If the startPoint is omitted the startPoint is assumed to be equal to the value of gml:low in the gml:Grid geometry*. Subsequent points in the mapping are determined by the value of the sequenceRule.

7.13.5.11 gml:sequenceRule

The sequenceRule is described by the content model:

```
 <complexType name="SequenceRuleType">
  <simpleContent>
   <extension base="gml:SequenceRuleNames">
    <attribute name="order" type="gml:IncrementOrder" use="optional"/>
   </extension>
  </simpleContent>
 </complexType>
```

The SequenceRuleType is derived from the gml:SequenceRuleNames through the addition of an order attribute. The gml:SequenceRuleNames is an enumerated type defined as:

```
<simpleType name="SequenceRuleNames">
  <restriction base="string">
    <enumeration value="Linear"/>
    <enumeration value="Boustrophedonic"/>
    <enumeration value="Cantor-diagonal"/>
    <enumeration value="Spiral"/>
    <enumeration value="Morton"/>
    <enumeration value="Hilbert"/>
  </restriction>
</simpleType>
```

These rule names are defined in ISO 19123.

If no rule name is specified the default is "Linear".

The order attribute is also defined in ISO 19123 and its value is determined by the content model:

```
<simpleType name="IncrementOrder">
  <annotation>
    <documentation> </documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="+x+y"/>
    <enumeration value="+y+x"/>
    <enumeration value="+x-y"/>
    <enumeration value="-x-y"/>
  </restriction>
</simpleType>
```

The enumeration value here indicates the incrementing order to be used on the first 2 axes, i.e. "+x-y" means that the points on the first axis are to be traversed from lowest to highest and the points on the second axis are to be traversed from highest to lowest. The points on all other axes (if any) beyond the first 2 are assumed to increment from lowest to highest.

If the order attribute is omitted it is assumed to have the value "+x+y".

If the coverageFunction property is omitted for Gridded Coverages (included RectifiedGridded Coverages) the startPoint is assumed to be the value of the gml:low property in the gml:Grid geometry, and the sequenceRule is assumed to be linear and the order attribute is assumed to be "+x+y". This is best illustrated by a simple example as follows:

```xml
<AverageTempPressure xmlns="http://www.opengis.net/app" xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/app
CoverageExamples.xsd">
  <gml:domainSet>
    <gml:Grid dimension="2">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>0 0</gml:low>
          <gml:high>3 3</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisName>x</gml:axisName>
      <gml:axisName>y</gml:axisName>
    </gml:Grid>
  </gml:domainSet>
  <gml:rangeSet>
    <gml:DataBlock>
      <gml:rangeParameters>
        <gml:CompositeValue>
          <gml:valueComponent>
            <my:Temperature uom="#C"/>
          </gml:valueComponent>
          <gml:valueComponent>
            <my:Pressure uom="#kPa"/>
          </gml:valueComponent>
        </gml:CompositeValue>
      </gml:rangeParameters>
      <gml:tupleList>3,101.2 5,101.3 7,101.4 11,101.5  13,101.6  17,101.7  19,101.7  23,101.8  29,101.9
31,102.0  37,102.1  41,102.2  43,102.3  47,102.4  53,102.5  59,102.6</gml:tupleList>
    </gml:DataBlock>
  </gml:rangeSet>
  <gml:coverageFunction/>
</AverageTempPressure>
```

Since no coverageFunction is specified the function is assumed to be that of linear scanning with "+x+y" order starting at the location (0 0). If we look at the DataBlock we see that we have the following mapping.

Grid Location	**Data Value**
0 0	3,101.2
1 0	5,101.3
2 0	7,101.4
3 0	11,101.5
0 1	13,101.6
1 1	17,101.7
2 1	19,101.7
3 1	23,101.8
0 2	29,101.9

1 2	31,102.0
2 2	37,102.1
3 2	41,102.2
0 3	43,102.3
1 3	47,102.4
2 3	53,102.5
3 3	59,102.6

7.13.5.12 Specific Coverage Types in GML 3

GML 3.0 supports only a limited subset of the coverage types defined in ISO 19123. The supported coverages are show in a UML class diagram in Figure 7.12.6.

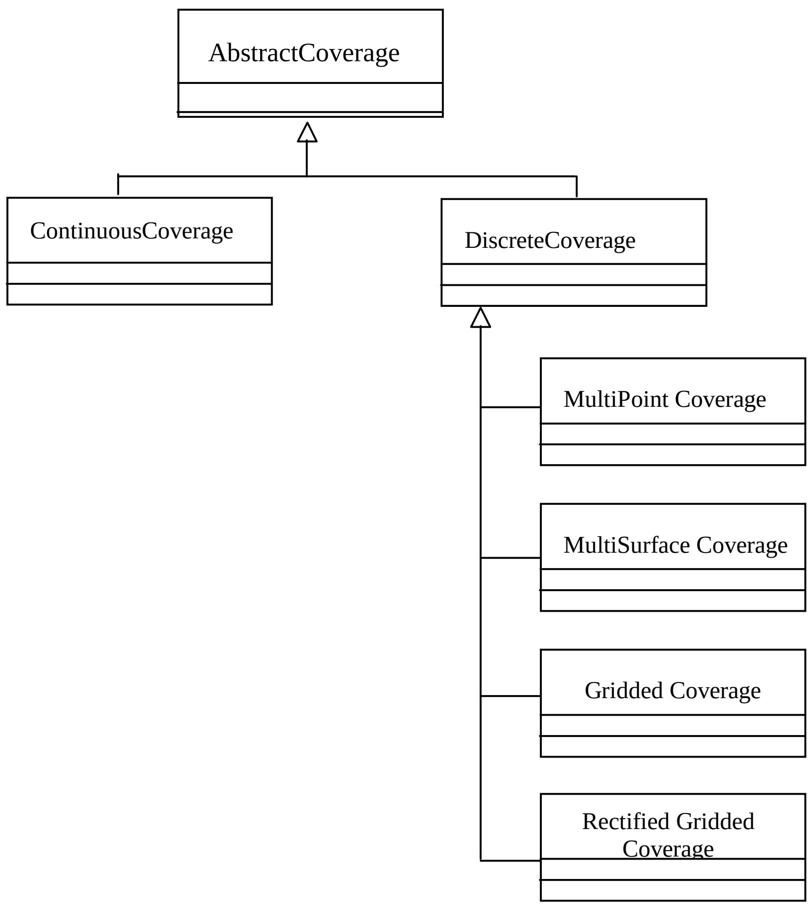


Figure 7.12.6 Coverage Type Hierarchy

These types are derived from gml:AbstractCoverageType and include:

- MultiPointCoverage
- MultiSurfaceCoverage
- Gridded Coverage (discrete point coverage)
- Rectified Grid Coverage (discrete point coverage)

Additional specific coverage types can be anticipated in future releases of GML. Users can also construct their own coverage types from gml:AbstractCoverageType or by derivation from the specific concrete coverage types above.

Note that the same Range Set encodings apply for each of the different coverage types as the latter are specified by the geometry type of the domain.

7.13.5.13 MultiPoint Coverage

In a MultiPoint Coverage the domain set is a gml:MultiPoint, that is a collection of arbitrarily distributed geometric points. The content model for a MultiPoint Coverage is as follows:

```
<complexType name="MultiPointCoverageType">
  <complexContent>
    <restriction base="gml:AbstractCoverageType">
      <sequence>
        <element ref="gml:multiPointDomain"/>
        <element ref="gml:rangeSet"/>
        <element ref="gml:coverageFunction" minOccurs="0"/>
      </sequence>
      <attribute name="dimension" type="positiveInteger" use="required"/>
    </restriction>
  </complexContent>
</complexType>
```

Note that this is defined by restriction on gml:AbstractCoverageType. Note that the restriction replaces the generic gml:domainSet by the specific gml:multiPointDomain whose value is a gml:MultiPoint.

```
<complexType name="MultiPointDomainType">
  <complexContent>
    <restriction base="gml:DomainSetType">
      <sequence minOccurs="0">
        <element ref="gml:MultiPoint"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
```

Note that in a MultiPoint Coverage the mapping from the domain to the range is straightforward.

1. For DataBlock encodings the points of the MultiPoint are mapped in document order to the tuples of the DataBlock.
2. For ValueCollection encodings the points of the MultiPoint are mapped to the members of the ValueCollection in document order.
3. For File encodings the points of the MultiPoint are mapped to the records of the File in sequential order.

Example: MultiPoint Coverage (uses Value encoding)

```
<AverageTempPressure xmlns="http://www.opengis.net/app" xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/app
CoverageExamples.xsd">
	<gml:domainSet>
		<gml:MultiPoint>
			<gml:pointMember>
				<gml:Point>
					<gml:pos>1 1</gml:pos>
				</gml:Point>
			</gml:pointMember>
			<gml:pointMember>
				<gml:Point>
					<gml:pos>2 2</gml:pos>
				</gml:Point>
			</gml:pointMember>
			<gml:pointMember>
				<gml:Point>
					<gml:pos>3 3</gml:pos>
				</gml:Point>
			</gml:pointMember>
			<gml:pointMember>
				<gml:Point>
					<gml:pos>4 4</gml:pos>
				</gml:Point>
			</gml:pointMember>
		</gml:MultiPoint>
	</gml:domainSet>
	<gml:rangeSet>
		<gml:ValueCollection>
			<gml:valueMembers>
				<TemperatureReading>3</TemperatureReading>
				<TemperatureReading>5</TemperatureReading>
				<TemperatureReading>7</TemperatureReading>
				<TemperatureReading>11</TemperatureReading>
				<TemperatureReading>13</TemperatureReading>
				<TemperatureReading>17</TemperatureReading>
				<TemperatureReading>19</TemperatureReading>
				<TemperatureReading>23</TemperatureReading>
				<TemperatureReading>29</TemperatureReading>
				<TemperatureReading>31</TemperatureReading>
				<TemperatureReading>37</TemperatureReading>
				<TemperatureReading>41</TemperatureReading>
				<TemperatureReading>43</TemperatureReading>
				<TemperatureReading>47</TemperatureReading>
				<TemperatureReading>53</TemperatureReading>
				<TemperatureReading>59</TemperatureReading>
			</gml:valueMembers>
		</gml:ValueCollection>
	</gml:rangeSet>
</AverageTempPressure>
```

7.13.5.14 MultiSurface Coverage

In a multi-surface coverage the domain is partioned into a collection of surface elements comprising a gml:MultiSurface. The coverage function than maps each surface element in the collection to a value in the Range Set. The content model for the MultiSurface coverage is as follows:

```
<complexType name="MultiSurfaceCoverageType">
  <complexContent>
    <restriction base="gml:AbstractCoverageType">
      <sequence>
        <element ref="gml:multiSurfaceDomain"/>
        <element ref="gml:rangeSet"/>
        <element ref="gml:coverageFunction" minOccurs="0"/>
      </sequence>
      <attribute name="dimension" type="positiveInteger" use="required"/>
    </restriction>
  </complexContent>
</complexType>
```

Note that the domainSet is now a multiSurfaceDomain with value a gml:MultiSurface.

The coverage function provides a mapping from the elements of the MultiSurface to the values in the Range Set.

In the DataBlock encoding, the members of the MultiSurface (value of the multiSurfaceDomain) are mapped to the tuples in the DataBlock in document order.

In the File encoding, the members of the MultiSurface are mapped to the records in the File in document order.

In the ValueCollection encoding the members of the MultiSurface are mapped to the members of the ValueCollection in document order.

Example: MultiSurface Coverage (uses File encoding for values)

```xml
<SoilData>
 <gml:domainSet>
  <gml:MultiPolygon>
   <gml:polygonMember>
    <gml:Polygon gid="p1">
     <gml:innerBoundaryIs>
      <gml:LinearRing>
       <gml:coordinates/>
      </gml:LinearRing>
     </gml:innerBoundaryIs>
    </gml:Polygon>
   </gml:polygonMember>
   <gml:polygonMember>
    <gml:Polygon gid="p6">
     <gml:innerBoundaryIs>
      <gml:LinearRing>
       <gml:coordinates/>
      </gml:LinearRing>
     </gml:innerBoundaryIs>
    </gml:Polygon>
   </gml:polygonMember>
   <gml:polygonMember>
    <gml:Polygon gid="p11">
     <gml:innerBoundaryIs>
      <gml:LinearRing>
       <gml:coordinates/>
      </gml:LinearRing>
     </gml:innerBoundaryIs>
    </gml:Polygon>
   </gml:polygonMember>
   <gml:polygonMember>
    <gml:Polygon gid="p16">
     <gml:innerBoundaryIs>
      <gml:LinearRing>
       <gml:coordinates/>
      </gml:LinearRing>
     </gml:innerBoundaryIs>
    </gml:Polygon>
   </gml:polygonMember>
  </gml:MultiPolygon>
 </gml:domainSet>
 <gml:rangeSet>
  <gml:File>
   <gml:fileName>soil.dat</gml:fileName>
   <gml:fileStructure>Record Interleaved</gml:fileStructure>
   <gml:rangeParameters>
    <gml:CompositeValue>
     <gml:valueComponent>
      <SoilType codeSpace=”http://my.big.org/classifications/soils”/>
     </gml:valueComponent>
     <gml:valueComponent>
      <SoilMoisture uom=”http://my.big.org/units/percent”/>
     </gml:valueComponent>
    </gml:CompositeValue>
   </gml:rangeParameters>
  </gml:File>
 </gml:rangeSet>
</SoilData>
```

7.13.5.15 Gridded Coverage

A gridded coverage is a discrete point coverage in which the domain set is a geometric grid of points as shown in Figure 7.12.7.

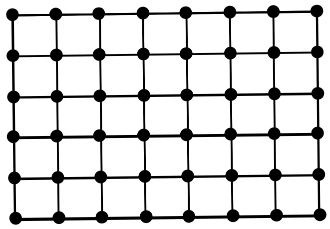


Figure 7.12.7 Gridded Coverage domain is a grid of points

The content model for the gridded coverage is as follows:

```
<element name="GridCoverage" type="gml:GridCoverageType" substitutionGroup="gml:_Coverage"/>

 <complexType name="GridCoverageType">
  <complexContent>
   <restriction base="gml:AbstractCoverageType">
    <sequence>
     <element ref="gml:gridDomain"/>
     <element ref="gml:rangeSet"/>
     <element ref="gml:coverageFunction" minOccurs="0"/>
    </sequence>
    <attribute name="dimension" type="positiveInteger" use="required"/>
   </restriction>
  </complexContent>
 </complexType>
```

Note that this is the same as the MultiPoint coverage except that we have a gml:gridDomain property to describe the domain.

The gml:gridDomain is defined as:

```
 <complexType name="GridDomainType">
  <complexContent>
   <restriction base="gml:DomainSetType">
    <choice minOccurs="0">
     <element ref="gml:Grid"/>
    </choice>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
   </restriction>
  </complexContent>
```



```
  </complexType>
```

The gml:Grid is defined in the grids.xsd discussed in Clause 7.9.3. Note that the simple gridded coverage is not geometrically referenced and hence no geometric positions are assignable to the points in the grid. Such geometric positioning is introduced in the RectifiedGrid Coverage discussed in Clause 7.9.2.4.4.

```
<AverageTempPressure >
  <gml:domainSet>
    <gml:Grid dimension="2">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>0 0</gml:low>
          <gml:high>4 4</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisName>x</gml:axisName>
      <gml:axisName>y</gml:axisName>
    </gml:Grid>
  </gml:domainSet>
  <gml:rangeSet>
    <gml:File>
      <gml:fileName>.../temperature.dat</gml:fileName>
      <gml:fileStructure>Record Interleaved</gml:fileStructure>
      <gml:rangeParameters>
        <gml:CompositeValue>
          <gml:valueComponent>
            <Temperature/>
          </gml:valueComponent>
          <gml:valueComponent>
            <Pressure/>
          </gml:valueComponent>
        </gml:CompositeValue>
      </gml:rangeParameters>
    </gml:File>
  </gml:rangeSet>
</AverageTempPressure>
```

Example: Grid Coverage (uses File encoding for values)

7.13.5.16 RectifiedGrid Coverage

The rectified grid coverage is a discrete point coverage based on a rectified grid.

The rectified grid coverage is similar to the grid coverage of Clause 7.9.2.4.3 except that the points of the grid are geometrically referenced. The rectified grid coverage has a domain that is a RectifiedGrid geometry as defined in the grids.xsd of Clause 7.9.3.

The content model for gml:RectifiedGridCoverage is as follows:

```
<complexType name="RectifiedGridCoverageType">
  <complexContent>
    <restriction base="gml:AbstractCoverageType">
      <sequence>
        <element ref="gml:rectifiedGridDomain"/>
        <element ref="gml:rangeSet"/>
        <element ref="gml:coverageFunction" minOccurs="0"/>
      </sequence>
      <attribute name="dimension" type="positiveInteger" use="required"/>
    </restriction>
  </complexContent>
</complexType>
```

Note that the coverage domain is described by gml:rectifiedGridDomain.

```
<complexType name="RectifiedGridDomainType">
  <complexContent>
    <restriction base="gml:DomainSetType">
      <choice minOccurs="0">
        <element ref="gml:RectifiedGrid"/>
      </choice>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>
```

where the gml:RectifiedGrid geometry is defined in grids.xsd discussed in Clause 7.9.3/

Example: RectifiedGrid Coverage (using DataBlock)

```
<AverageTempPressure >
  <gml:domainSet>
    <gml:RectifiedGrid dimension="2">
      <gml:limits>
        <gml:GridEnvelope>
          <gml:low>1 1</gml:low>
          <gml:high>4 4</gml:high>
        </gml:GridEnvelope>
      </gml:limits>
      <gml:axisName>u</gml:axisName>
      <gml:axisName>v</gml:axisName>
      <gml:origin>
        <gml:Point gml:id="palindrome">
          <gml:coordinates>1.2,3.3,2.1</gml:coordinates>
        </gml:Point>
      </gml:origin>
      <gml:offsetVector>1, 2, 3</gml:offsetVector>
      <gml:offsetVector>2, 1, 0</gml:offsetVector>
    </gml:RectifiedGrid>
  </gml:domainSet>
  <gml:rangeSet>
    <gml:DataBlock>
      <gml:tupleList>3,101.2 5,101.3 7,101.4 11,101.5  13,101.6  17,101.7  19,101.7  23,101.8  29,101.9
31,102.0  37,102.1  41,102.2  43,102.3  47,102.4  53,102.5  59,102.6</gml:tupleList>
      <gml:rangeParameters>
        <gml:CompositeValue>
          <gml:valueComponent>
            <Temperature/>
          </gml:valueComponent>
          <gml:valueComponent>
            <Pressure/>
          </gml:valueComponent>
        </gml:CompositeValue>
      </gml:rangeParameters>
    </gml:DataBlock>
  </gml:rangeSet>
</AverageTempPressure>
```

7.14 Default Styling

7.14.1 General concepts

One of the requirements in developing of GML3.0 (and previous versions) is strict separation of data and presentation. Therefore none of the GML data description constructs have built-in capability to describe the styling information. Rather, the default styling mechanism was created as a separate model that can be “plugged-in” to a GML data set. An example of the use of such pluggable mechanism is data styling information persistence.

The term “default” signifies very relaxed relation to other parts of the GML model. The style information that is assigned to a data set may be used for styling but may also be completely ignored.

The default style schema is directly related to the rest of the GML schemas only in that it imports some very basic constructs from it. None of the other GML schemas depend on the styling one. It also depends on W3C Synchronized Multimedia Integration Language (SMIL) schemas.

The relation of the default style information and GML data instances is achieved through the gml:deafultStyle property defined in defaultStyle.xsd schema. The property may be assigned to the instance by defining such a relationship in application schema. Since GML is feature-based encoding, GML default style always applies to a feature, features or feature collections.

The UML diagram of the styling model is shown in Figure 7.13.1.

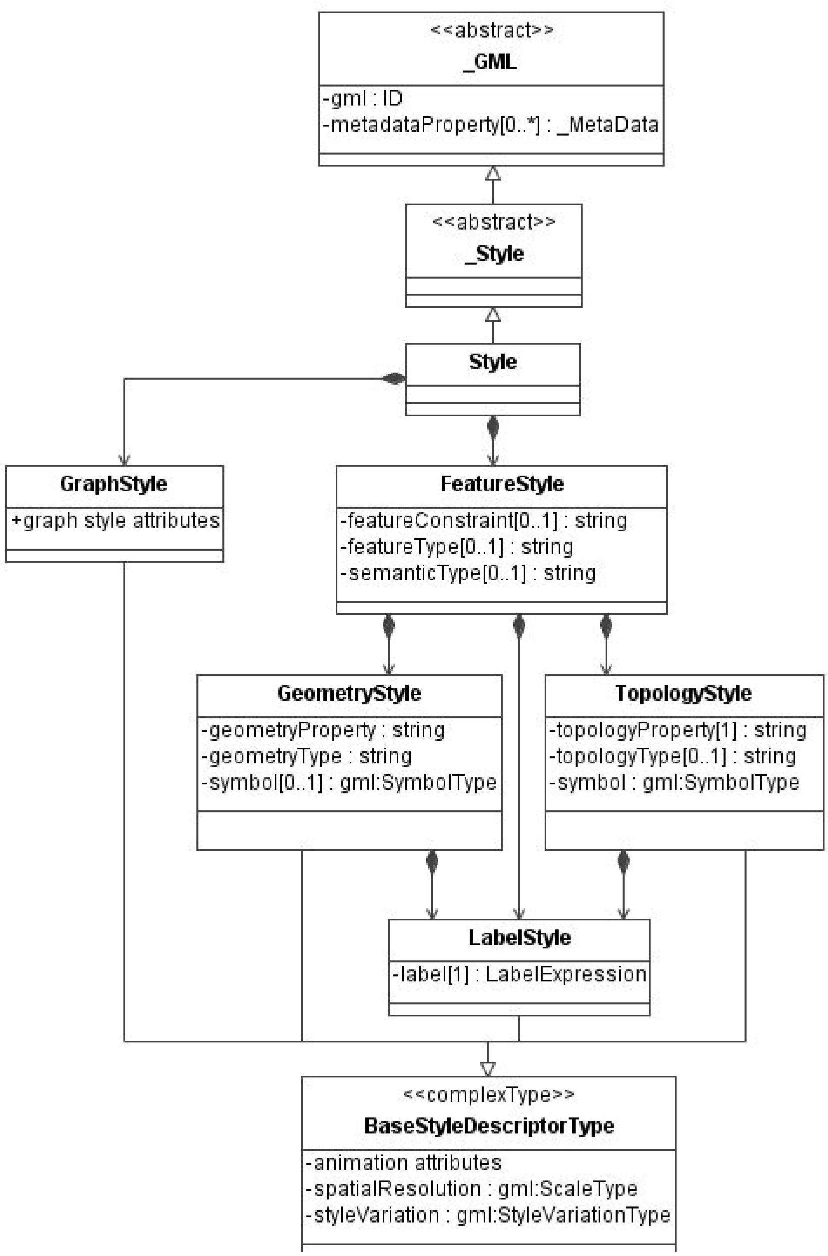


Figure 7.13.1. The GML default styling model

7.14.2 Top-level styling elements

The connection between a GML data set and a styling description is minimal. It is established through only one property, the `gml:defaultStyle`. The value of this property, the `gml:Style` object, contains all styling descriptions. In GML default styling only this property/value pair conforms to GML property/value model, while other styling elements do not.

The *gml:defaultStyle* property

The gml:defaultStyle property is a property defined as a global element in GML namespace and thus can be assigned to any feature or a feature collection defined in an application schema. The definition of the property is shown in the following listing:

```
<!-- ======================================================= -->
<element name="defaultStyle" type="gml:DefaultStylePropertyType"
substitutionGroup="gml:_property"/>
<!-- ======================================================= -->
<complexType name="DefaultStylePropertyType">
  <sequence>
    <element ref="gml:_Style" minOccurs="0"/>
  </sequence>
  <attribute name="about" type="anyURI" use="optional"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!-- ======================================================= -->
```

The `gml:defaultStyle` property is a GML property -- that is specified by the value of the `substitutionGroup` attribute. Thus, the property is implicitly included in any feature definition that contains the `gml:_property`. In other words, `gml:defaultStyle` can appear anywhere in place of `gml:_property`, without explicit specification.

However, it will usually be the case that users want to include this property in a feature explicitly. This is achieved by including it in a definition in an application schema. Following is an example of an `exp:Road` feature definition that includes the `gml:defaultStyle` property.

```
<element name="Road" type="exp:RoadType"
substitutionGroup="gml:_Feature"/>
<complexType name="RoadType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:centerLineOf"/>
        <element ref="gml:defaultStyle"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
```

The `gml:AssociationAttributeGroup` on the property contains the simple XLink attribute group from xlinks.xsd schema and the the `remoteSchema` attribute. Attributes from xslinks.xsd schema allow a user to specify the value of the property remotely instead of inline. The `remoteSchema` attribute may be used to specify the schema of the property's value.

The `about` attribute on the property allows us to assign the style to an arbitrary feature or feature collection regardless of the style's location. The meaning of the attribute is that "The style is about (applies to) feature or features that can be found at the URI that is the attribute value".

The *gml:Style* object

The `gml:Style` object is the default concrete value of the `gml:defaultStyle` property. It is the top-level styling object that encapsulates all other, partial style descriptions. It's definition is presented in the following listing:

```
<!-- ============================================================ -->
<element name="Style" type="gml:StyleType"
substitutionGroup="gml:_Style"/>
<!-- ============================================================ -->
<complexType name="StyleType">
  <complexContent>
    <extension base="gml:AbstractStyleType">
      <sequence>
        <element ref="gml:FeatureStyle" minOccurs="0"
maxOccurs="unbounded"/>
        <element ref="gml:GraphStyle" minOccurs="0"/>
```

```
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ============================================================= -->
```

The `gml:StyleType` which is the content model of the `gml:Style` object extends the `gml:AbstractStyleType`. This base abstract type serves as an abstract base for extensibility purposes, i.e. creating custom style objects, and it does not add any new content to the `gml:AbstractGMLType` from which it derives. This can be seen in the listing below:

```
<!-- ============================================================= -->
<element name="_Style" type="gml:AbstractStyleType" abstract="true"
substitutionGroup="gml:_GML"/>
<!-- ============================================================= -->
<complexType name="AbstractStyleType" abstract="true">
  <complexContent>
    <extension base="gml:AbstractGMLType"/>
  </complexContent>
</complexType>
<!-- ============================================================= -->
```

However, it is not assumed that creating custom style objects will be usual practice since `gml:Style` object provides rich capabilities for describing styles. In case of using this mechanism, usual GML rules have to be observed:

1. The content model of the custom style object has to derive from `gml:AbstractStyleType`.
2. The custom style object has to be substitutable for abstract `gml:_Style`.

The definition of the `gml:Style` object presented previously in the text is itself a proper example of using the extensibility mechanism and it shows how these rules are implemented in the schema.

The function of the styling elements in the `gml:Style` object, namely `gml:FeatureStyle` and `gml:GraphStyle` is to describe styles for two aspects of GML data: individual features and topology graphs that consist of collections of features. Note that elements that describe styles for particular aspects of features, namely, feature style, graph style, geometry style, topology style and label style are often called style descriptors.

7.14.3 Feature style

Feature style descriptor describes the styling information for a set of feature instances. The set is defined by the selection mechanisms that are part of this style descriptor. The style applies to each feature in the set independently - no relations that might exist among features in the set are significant. (The opposite case is graph style where the style applies to a set of features as a whole). The definition of the feature style descriptor is shown in the following listing:

```
<!-- ============================================================ -->
<element name="FeatureStyle" type="gml:FeatureStyleType">
<!-- ============================================================ -->
<complexType name="FeatureStyleType">
  <sequence>
    <element name="featureConstraint" type="string" minOccurs="0"/>
    <element ref="gml:GeometryStyle" minOccurs="0"
maxOccurs="unbounded"/>
    <element ref="gml:TopologyStyle" minOccurs="0"
maxOccurs="unbounded"/>
    <element ref="gml:LabelStyle" minOccurs="0"/>
  </sequence>
  <attribute name="featureType" type="string" use="optional"/>
  <attribute name="baseType" type="string" use="optional"/>
</complexType>
<!-- ============================================================ -->
```

Feature instances to which the style applies are selected using one of the attributes `featureType` or `baseType` and `gml:featureConstraint` element. These two attributes must be used exclusively, with or without the `gml:featureConstraint` element.

<u>The *featureType* attribute</u>

The simplest and most common way of relating features and styles is by using this attribute. It’s value will be the declared name of a feature, instances of which we want to style. For example, if the value is `exp:Road`, the `gml:FeatureStyle` object will simply apply to all Road features. The value of this attribute is always the name of the element from the application schema that declares the feature.

<u>The *baseType* attribute</u>

Another way of selecting the feature instances to which the style applies is to specify, as the value of this attribute, the name of the base type from which feature or features derive. This is always the name of an XMLSchema complex type. Any complex type from the derivation chain can be used; the style applies to any feature instance that ultimatelly derives from it. If we use, for example, `gml:AbstractFeatureType` as the attribute's value, the style applies to all feature instances in a data set.

featureConstraint property

This property is used to further constrain the feature instance set to which the style applies. It is optional and it's value is an XPath expression. If the property does not exist, the style applies to all feature instances selected by `featureType` or `baseType` attribute.

Styling features means styling a particular aspect or aspects of a feature. We can style feature geometry, topology or display arbitrary text string. Feature style contains three style descriptors for respective purposes: `gml:GeometryStyle`, `gml:TopologyStyle` and `gml:LabelStyle`.

7.14.4 Geometry Style

The `gml:GeometryStyle` describes the style for one geometry of a feature. Any number of geometry style descriptors can be assigned to one feature style. This is usually required for features with multiple geometry properties.

Geometry style descriptor is defined as follows:

```
<!-- ============================================================ -->
<element name="GeometryStyle" type="gml:GeometryStyleType"/>
<!-- ============================================================ -->
<complexType name="GeometryStyleType">
  <complexContent>
    <extension base="gml:BaseStyleDescriptorType">
      <sequence>
        <choice>
          <element ref="gml:symbol"/>
          <element name="style" type="string"/>
        </choice>
        <element ref="gml:LabelStyle" minOccurs="0"/>
      </sequence>
      <attribute name="geometryProperty" type="string"/>
      <attribute name="geometryType" type="string"/>
    </extension>
```

```
  </complexContent>
</complexType>
<!-- ============================================================== -->
```

The `geometryProperty` attribute specifies the name of the geometry property of a feature to which this geometry style descriptor applies. It is necessary to specify the geometry type using `geometryType` attribute as well since the application schema of the geometry property may allow different geometries as it's value.

Elements `gml:symbol` and `gml:style` are described in the section 7.14.7 .

7.14.5 Topology Style

The `gml:TopologyStyle` descriptor describes the style for one topology property. Similarly to the `gml:Geometry` style, a feature can have multiple topology properties, thus multiple topology style descriptors can be specified within one feature style.

The definition of topology style is presented in the following listing:

```
<!-- ============================================================== -->
<element name="TopologyStyle" type="gml:TopologyStyleType"/>
<!-- ============================================================== -->
<complexType name="TopologyStyleType">
  <complexContent>
    <extension base="gml:BaseStyleDescriptorType">
      <sequence>
        <choice>
          <element ref="gml:symbol"/>
          <element name="style" type="string"/>
        </choice>
        <element ref="gml:LabelStyle" minOccurs="0"/>
      </sequence>
      <attribute name="topologyProperty" type="string"/>
      <attribute name="topologyType" type="string"/>
    </extension>
  </complexContent>
</complexType>
<!-- ============================================================== -->
```

The `topologyProperty` attribute specifies the name of the topology property of a feature to which this topology style descriptor applies. It is necessary to specify the topology type using `topologyType` attribute as well since the application schema of the topology property may allow different topologies as it's value.

Elements `gml:symbol` and `gml:style` are described in the section 7.14.7.

7.14.6 Label Style

Label style describes the style for the text that is to be displayed along with the graphical representation of a feature. The content of the label is not necessarily defined in the GML data set. More precisely, the content can be static text specified in the style itself and the text from the GML data set.

Label style has two elements: `gml:style` that has the same meaning and use as in other style descriptors and is described in detail in the section 7.14.7, and the `gml:label` that is used to compose the label content. The definition of the `gml:LabelStyle` descriptor is given in the following listing:

```
<!-- =========================================================== -->
<element name="LabelStyle" type="gml:LabelStyleType"/>
<!-- =========================================================== -->
<complexType name="LabelStyleType">
  <complexContent>
    <extension base="gml:BaseStyleDescriptorType">
      <sequence>
        <element name="style" type="string"/>
        <element name="label" type="gml:LabelType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- =========================================================== -->
```

As noted, the label content can be composed of static text and the text extracted from the GML data. The definition of the `gml:label` element that is used to extract the data is given below:

```
<complexType name="LabelType" mixed="true">
  <sequence>
    <element name="LabelExpression" type="string" minOccurs="0"
maxOccurs="unbounded"/>
```

```
  </sequence>
  <attribute ref="gml:transform" use="optional"/>
</complexType>
```

The content model is mixed to allow both text content and unbounded number of `gml:LabelExpression` elements. The value of `gml:LabelExpression` element is an XPath expression that selects the value of some property of the feature.

For example consider this GML data fragment and corresponding `gml:label` style:

```
<exp:City>
  <gml:name>Belgrade</gml:name>
  <exp:size>1,700,000</exp:size>
  <gml:extentOf>
  ……
  </gml:extentOf>
</exp:City>
```

```
<gml:FeatureStyle featureType="exp:City">
  <gml:LabelStyle>
    <gml:style>font-family:Verdana;font-size:18;fill:red</gml:style>
    <gml:label>
      City:
      <gml:LabelExpression>//City/name</gml:LabelExpression>
      , Size:
      <gml:LabelExpression>//City/size</gml:LabelExpression>
    </gml:label>
  </gml:LabelStyle>
</gml:FeatureStyle>
```

This label style will result in the following text being displayed:

City: Belgrade, Size: 1,700,000

7.14.7 Common styling elements

Some common styling elements are used in multiple style descriptors. The `gml:symbol` element is used by geometry and topology style descriptors. The `gml:style` element is declared in and used by geometry, topology and label style descriptors The `spatialResolution, styleVariation` and animation attributes are declared in `gml:BaseStyleDesriptorType`, and inherited by geometry, topology, label and graph style descriptors.

<u>*symbol* element</u>

The symbol element specifies a graphical symbol used to render a geometry or a topology. A symbol is a description of graphical attributes of a graphical object without a particular, implicit meaning. It can be a description of a line, circle, polygon or more complex drawing. Using the symbol element, we can specify a particular symbol in two ways:

1. Remote: Just like any other remote property, the symbol property has the `gml:AssociationAttributeGroup` attributes that allow for specifying a link pointing to a remote object.
2. Inline: The value of the `gml:symbol` property is `gml:_Object`. A symbol can be made substitutable for `gml:_Object` and specified inline. This requires additional definition in an application schema.

This element has two additional attributes: `symbolType` and `transform`. The `symbolType` attribute is enumeration and can take one of two values: `svg` or `other`. Applications will rely on the value of this attribute to decide how to interpret the symbol.

The `transform` attribute allows us to specify a transformation expression that will be applied to the symbol in the rendering phase. It's type is `xsd:string` and the value is specified in the SVG specification (transform attribute).

<u>*style* element</u>

In some cases, when the symbol is composed of very simple graphic primitives or attributes, the `gml:style` element can be used instead. Also, in the case of label style, the symbol doesn't apply at all because no graphic will be drawn on the map, but only text content. The type of this element is `xsd:string` and the SVG grammar is used to express graphic properties. The following example shows the use of the `gml:style` element in the geometry style context.

```
<gml:FeatureStyle featureType="exp:City">
  <gml:GeometryStyle>
    <gml:style>fill:blue;stroke:white</gml:style>
  </gml:GeometryStyle>
</gml:FeatureStyle>
```

styleVariation element

The function of the styleVariation element is manifold:

1. Styling labels.
   Label style does not have a symbol associated with it since the content is not graphical but is given textually. Using this property we can specify it's style attributes.
2. Styling symbol variations.
   One symbol is often used in different cases with slight modifications. It would be cumbersome to create and manage large number of virtually identical symbols; it is easier to create and use only one symbol and express minor differences in it's style using this property.
3. Parametrized styles.
   Parametrized styles are styles whose attributes depend on some property of the feature being styled. For example, a city might be styled differently depending on it's population. The styleVariation property allows for specifying such dependencies.

The definition of this element's content is:

```
<complexType name="StyleVariationType">
  <simpleContent>
    <extension base="string">
      <attribute name="styleProperty" type="string" use="required"/>
      <attribute name="featurePropertyRange" type="string"
use="optional"/>
    </extension>
  </simpleContent>
</complexType>
```

It has two attributes: `styleProperty` and `featurePropertyRange`. The value of the `styleProperty` is an SVG styling attribute name, such as “stroke”, “fill”, etc. It specifies what attribute of the style the property sets or overrides. The value of the `styleVariation` element is the value of the styling attribute specified by the `styleProperty`. The value may be a constant expression or an XPath expression.

The `featurePropertyRange` attribute defines the subset of features to which the variation applies. It's value is an XPath expression.

The following example shows two variations of the symbol style for a City feature. The feature is styled using a circle symbol. The radius of the circle depends on the population of the city, and is also calculated differently depending whether the population of the city is greater or less than 2 million.

```
<gml:FeatureStyle featureType="exp:City">
  <gml:GeometryStyle>
    <gml:styleVariation styleProperty="r" featurePropertyRange=
"population &gt;= 2000000">population div 1000000</gml:styleVariation>
    <gml:styleVariation styleProperty="r" featurePropertyRange=
"population &lt; 2000000">population div 1000000</gml:styleVariation>
    <gml:symbol
xlink:href="http://www.opengis.org/symbols/City.xml#City"/>
  </gml:GeometryStyle>
</gml:FeatureStyle>
```

spatialResolution element

The value of the `spatialResolution` element is defined in the measures.xsd GML schema. The value is derived from the measure type, which is `xsd:double` type with `uom` (units of measure) attribute. In GML styling, the meaning of this element is based on the corresponding definition in the ISO 19115 Metadata DIS, where it is defined as a factor that provides a general understanding of the density of spatial data in the data set. Other than this informal definition, GML does not specify the exact use of this attribute. Application developers can use `spatialResolution` in different ways. For example, it can be used as a map scale denominator (1:50,000, 1:25000, etc.). Applications can also use it's value to determine how to draw features in different scales. For example, a city and it's features are typically drawn in more details on a large scale map, and perhaps only as a single symbol on a small scale map. Or a coastline can be drawn in detail on a large scale map, while a small scale map application can omit some coordinates for better performance.

animation attributes

Animation attributes are used to describe the animation behaviour of the geometry, topology, label or graph. These attributes are defined in the W3C SMIL specification (SMIL 2.0 BasicAnimation Elements):

ATTRIBUTE	USED FOR
animate	Generic attribute animation
animateMotion	Moving an element along the path
animateColor	Animating colour attributes
set	Setting the value of an attribute for a specified duration

7.14.8 Graph Style

GraphStyle descriptor describes style attributes of a graph formed by a set of features. It's definition is shown in the following listing:

```
<complexType name="GraphStyleType">
  <complexContent>
    <extension base="gml:BaseStyleDescriptorType">
      <sequence>
        <element name="planar" type="boolean" minOccurs="0"/>
        <element name="directed" type="boolean" minOccurs="0"/>
        <element name="grid" type="boolean" minOccurs="0"/>
        <element name="minDistance" type="double" minOccurs="0"/>
        <element name="minAngle" type="double" minOccurs="0"/>
        <element name="graphType" type="gml:GraphTypeType"
minOccurs="0"/>
        <element name="drawingType" type="gml:DrawingTypeType"
minOccurs="0"/>
        <element name="lineType" type="gml:LineTypeType"
minOccurs="0"/>
        <element name="aestheticCriteria"
type="gml:AesheticCriteriaType" minOccurs="0" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
```



```
</complexType>
```

Graph style descriptor describes the style for a graph as a whole, not for individual graph elements. It inherits from the base content model common styling properties described in the section 7.14.7.

This descriptor adds to the base content model a group of properties specific to graph styling – they describe the graph in terms of it's specific characteristics. The properties are described in the following table:

Element	Type	Use
planar	xsd:boolean	If true, the graph edges do not cross (planar graph); if false they may cross
directed	xsd:boolean	If true the graph is directed; if false it is not directed
grid	xsd:boolean	If true, the coordinates of vertices, crossings and bends have integer values, otherwise they may have decimal values
minDistance	xsd:double	A recommendation for the minimmum distance between vertices and non-incident edges
minAngle	xsd:double	A recommendation for the minimmum angle between consecutive incident edges (angular resolution)
graphType	xsd:enumeration	The type of the graph. The value may be TREE or BICONNECTED
drawingType	xsd:enumeration	the type of the drawing with respect to the orthogonality of edges. The value may be POLYLINE or ORTHOGONAL
lineType	xsd:enumeration	Determines whether there will be any bent edges. The value may be STRAIGHT or BENT
aestheticCriteria	xsd:enumeration	A recommendation for the general outline of the graph according to a particular aesthetic criteria. The value may be one of the following: MIN_CROSSINGS, MIN_AREA, MIN_BENDS, MAX_BENDS, UNIFORM_BENDS,

		MIN_SLOPES, MIN_EDGE_LENGTH, MAX_EDGE_LENGTH, UNIFORM_EDGE_LENGTH, MAX_ANGULAR_RESOLUTION, MIN_ASPECT_RATIO or MAX_SYMMETRIES

7.15 Modularisation and Dependencies

The base schemas for GML described above have been modularised so that application schemas that do not need the complete set of GML definitions may import only the topical subsets of GML that are required. For example, a GML version 2 application schema migrating to GML version 3 without adding any new definitions could continue to import feature.xsd. It would not import or require parsing of the new GML version 3 definitions for coordinate reference systems, topology, coverages, dynamic features, default styles, and observations. However, it would import and require parsing of all of the basic types that have been added in GML version 3.

The modularisation of GML creates the dependencies among the GML base schemas shown in Figure 7.15.1 below. A dashed arrow in the figure indicates that the schema at the tail of the arrow depends upon the schema at the head of the arrow. A dependency may occur where one schema <include>s another schema in the “gml” namespace. For example, feature.xsd <include>s geometryBasic2d.xsd. A dependency may also occur where one schema <import>s another schema for a namespace other than “gml”. For example, gmlBase.xsd <import>s xlinks.xsd from the “xlink” namespace.

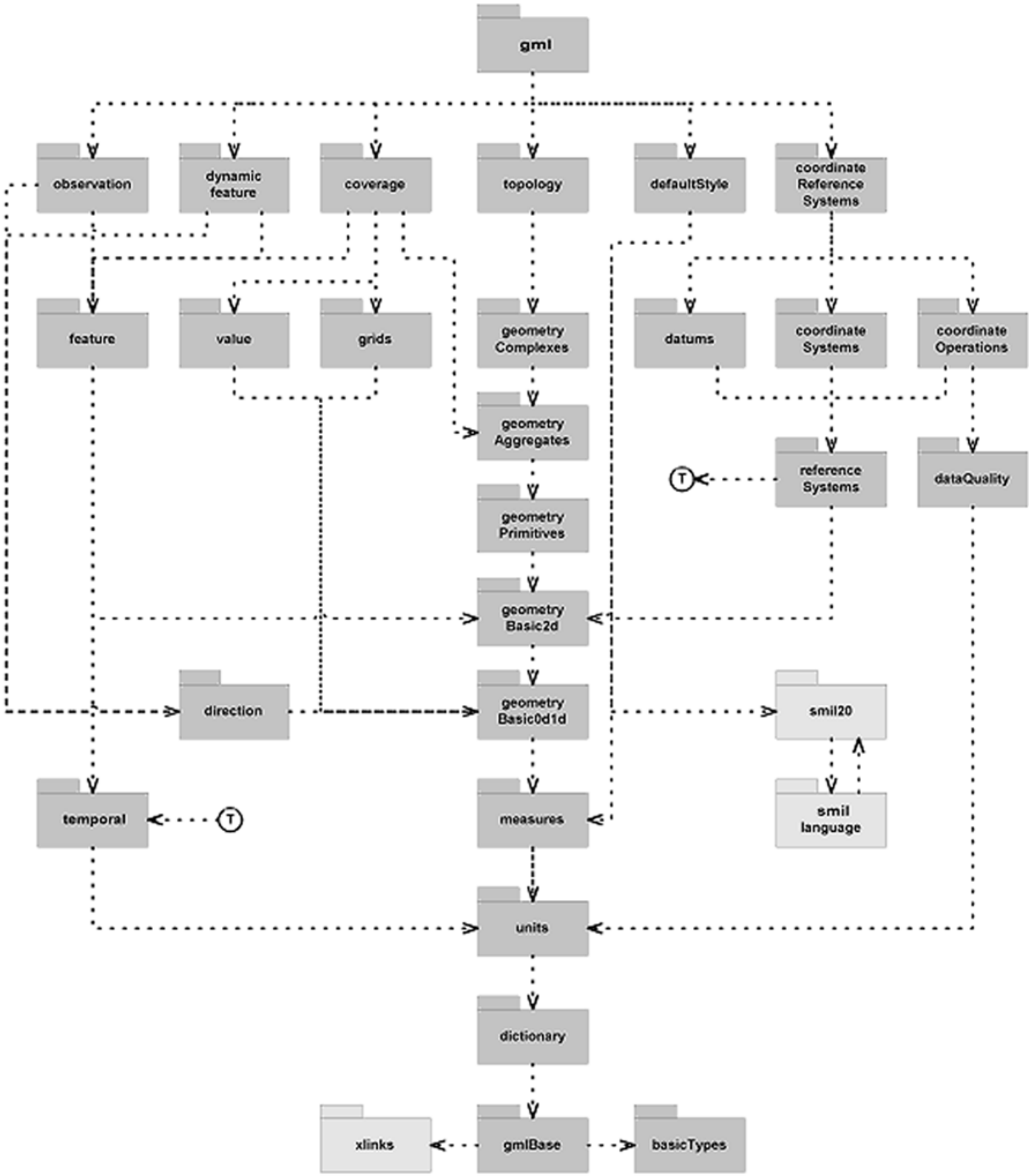


Figure 7.15-1 — Schema Dependencies

There are now six schemas in GML upon which no other GML schemas depend. These top level schemas are the roots of partially overlapping hierarchies of GML schemas:

- observation.xsd
- dynamicFeature.xsd
- coverage.xsd
- topology.xsd
- defaultStyle.xsd
- coordinateReferenceSystems.xsd

An application schema that needs definitions from more than one of these GML topical subset schema hierarchies can <import> gml.xsd and get all of the GML definitions. Or it can contain multiple <import>s for just the appropriate gml schema documents, thereby excluding unwanted GML type definitions. However, as specified in [XMLSchema-2] clause 4.2.3, “it is open to applications to ignore all but the first <import> for a given namespace”.

To work around this problem, an application schema writer may create a custom top-level GML schema by copying the <schema> element from gml.xsd, and <including> just the appropriate gml schema documents. This custom top-level GML schema in the http://www.opengis.net/gml namespace named “gml” is then <import>ed into the application schema, which uses its own application-specific namespace. For example, an application schema for features with topology could <import> a custom top-level GML schema that <include>s just feature.xsd and topology.xsd, thereby importing 17 fewer schemas than would have been imported using gml.xsd.

However, when an application schema will be used in a processing environment that lacks CPU, memory and/or I/O bandwidth, for example, in a mobile hand-held device, an absolutely minimal <import> of GML definitions is often desired. The custom top-level GML schema approach described above might bring in a unacceptably large number of unwanted definitions from each GML schema <include>d in the custom top-level GML schema. The solution is to create a single GML subset schema that contains exactly the required GML type and element definitions.However, creating such a GML subset schema by hand using a text or XML editor to cut and paste definitions is a tedious and error-prone process because it involves analyzing type definition dependencies across the many GML schema documents. An automated approach is recommended instead. An informative sample implementation of a GML schema subset tool is included in Annex F. Subset schemas, however they are produced, are Profiles of GML as described in Clause 7.15.

7.16 Profiles

7.16.1 Profiles of GML and application schemas

GML is a complex specification that is richly expressive. In general, an application need not exploit the entire specification, but may employ a subset of constructs corresponding to specific relevant requirements.

Quoting from technical report “ISO/IEC TR 10000-1:1998 Information technology - Framework and taxonomy of International Standardized Profiles - Part 1: General principles and documentation framework”**,** we use this definition of a profile:

Profile: A set of one or more base standards and/or [profiles], and, where applicable, the identification of chosen classes [(types, attributes and elements)], conforming subsets, options and parameters of those base standards, or [profiles] necessary to accomplish a particular function.

Of course, this was defined for an OSI architecture model, so we must translate ‘class’ to ‘types, attributes and elements’ to apply this definition to XML Schema. There are several ways to implements this, and GML profiles uses a “copy and delete” approach. To create a profile, a developer might copy the applicable schema files from GML and simply delete any global types, elements and local optional particles that she does not need for her application schema.

7.16.2 Definition of Profile

A profile of GML can be defined to enhance interoperability and to curtail ambiguity by allowing only a specific subset of GML. Application schemas can then conform to such a profile in order to take advantage of any interoperability or performance advantages that it offers in comparison with a complete GML. Such profiles can be defined for application schemas that are included in other OGC specifications.

There are cases where reduced functionality is acceptable, or where processing requirements compel use of a logical subset of GML. For example, applications that do not need to handle XLink attributes in any form can adhere to a specific profile that excludes them; the constraint in this case would be to not use links. Other cases might include defining constraints on the level of nesting allowed inside tags (i.e. tree depth), or only allowing features with homogeneous properties as members of a feature collection. In many cases, such constraints can be enforced via new schemas; others may be enforced through procedural agreements within an information community.

7.16.3 Relation to application schema

A profile can be the beginning of an application schema. For example, a location based service profile can limit the types of geometry to that used in LBS applications, and the LBS application schema can then add a “PointCircle,” “PointEllipse” and “PointArc” elements to accommodate the LIF "CIRCLE," "ELLIPSE" and "ARC" elements, which are used to describe error estimates of mobile device location.

The building of such application schemas is thus a two-part process. The profile acts as a restriction of GML to produce types and elements consistent with the complete GML 3 but potentially lacking in some optional particles. The application schema then uses these types as a common base, and uses them in new types and elements by extensions or inclusion.

$$\text{GML 3} \xrightarrow{\text{selection \& restriction}} \text{GML profile} \xrightarrow{\text{extension \& inclusion}} \text{application schema}$$

7.16.4 Rules for elements and types in a profile

Global profiled elements in a GML profile must:

1. Share the same name (and namespace) of a parent element in GML.
2. Include all mandatory particles (subelements and attributes) of the parent element in GML.
3. Include no particle that is not in the parent element in GML.
4. Have the same default values for attributes as the parent element in GML.
5. Have a parallel substitution group hierarchy for named elements in both schemas.

Global types in a GML profile must:

1. Share the same name (and namespace) of a parent type in GML.
2. Include all mandatory particles (subelements and attributes) of the parent type in GML.
3. Include no particle that is not in the parent type in GML.
4. Have the same default values for attributes as the parent type in GML.
5. Have a parallel derivation tree for named types in both schemas.

Instance documents of a profile must be valid against the full GML schema.

Using the "copy and delete" metaphor described above, our mythic developer can:

1. Delete global element and global types.
2. Delete optional subelements from any types or elements
3. Make optional subelements or attributes mandatory in any type or element (if a default value exist, it must be eliminated or the schema validation will report an error — default values are only valid for optional particles)
4. Restrict cardinality of any particle.

None of the above will affect the validity of a document that is designed against the profile, but tested against the full GML schema. Our mythic developer **cannot**:

1. Delete mandatory subelements from any types or elements.

2. Make mandatory particles optional.
3. Relax cardinality restrictions of any particle.
4. Add or change a default or fixed value.

Item 4 is a bit subtler than the others are. Documents valid under the profile would still be valid under the full GML schema, but the interpretation of those documents would change. For example, if a profile specified a default coordinate reference system to be UTM, and the full schema specified a WGS84 geodesic (latitude, longitude) as the default CRS, then the interpretation of the file would change when moving from the profile to the full schema.

7.16.5 Recommendations for application schemas using GML profiles

In order that the profile within an application schema can be later extended to include other profiled GML elements, the following recommendations are made:

1. Global elements that are not in a GML profile but are in an application schema using a GML profile should not have the same name as any element in the GML schemas.
2. Global types that are not in a GML profile but are in an application schema using a GML profile should not have the same name as any type in the GML schemas.

If a type or element in an application schema is found to be of universal use, then the above conventions will aid the application schema from migrating that type or element from its own namespace to that of GML.

The following recommendations are made simply as a bookkeeping convenience to those trying to understand the role of the profile in the application schema:

1. Profiled elements and types should be included in a file structure that parallels that of GML. The exact naming convention of the parallelism is left to the application schema author.
2. A reference to the appropriate GML schema file should be made in a comment near the beginning of the file.
3. The profile should use the GML namespace (http://www.opengis.net/gml)

7.16.6 Summary of Rules for GML profiles

In summary, the rules for a profile:

1. A profile of GML is a logical restriction of a subset of GML.
2. A profile must not change the name, definition, or data type of mandatory GML elements or attributes.
3. The relevant schema or schemas that define a profile must use in the core ‘gml’ namespace http://www.opengis.net/gml.

4. An application schema may extend and use types from the profile, but must do so in its own namespace, and not use http://www.opengis.net/gml.

The functional test of these rules is:

> In any instance document for an application schema using a GML profile will be valid against the same application schema if the GML profile is replaced by the complete GML schema. Further, the interpretation of that document would be the same regardless of which of the two schemas were used.

8 Rules for Application Schemes

8.1 GML Documents

A GML document conforming to GML 2.x consists of a feature collection. Its root element must have a content model derived directly or indirectly from gml:AbstractFeatureCollectionType. A GML 2.x document thus has a standard structure as follows:

```
<abc:FeatureCollection>
  <abc:featureMember>
    < .. some feature .. >
  </abc:featureMember>
  <abc:featureMember xlink:href=”http:// …”/>
  <abc:featureMember>
    < .. some feature .. >
  </abc:featureMember>
  <abc:featureMember>
  < .. other properties of the feature collection .. >
</abc:FeatureCollection>
```

where: <abc:FeatureCollection> is an element whose content model derives from gml:AbstractFeatureCollectionType.

Several document types are permitted using GML 3, including:

- Feature Collections (as in GML 2.x)
- Coverages (Coverages are Features so is a Feature Collection)
- Observation Collections (Observations are Features so is a Feature Collection)
- Coordinate Reference System Dictionaries
- Dictionaries
- Topological Complexes
- Units of Measure Dictionaries

Each of these document types is then supported by a corresponding application schema type. Both the document type and the associated application schema are described in this clause.

8.2 GML Application Schemas

A GML Application Schema is an XML Schema, conforming to the rules outlined in this clause that describes one or more types of geographic objects, components of geographic objects, or meta data, including dictionaries and definitions, used in the definition of geographic objects.. A GML Application Schema defines a vocabulary for a particular domain of discourse by defining and describing the terms of that vocabulary (see ISO TC/211 19109).

All GML Application Schemas are constructed, using the rules of this clause, from one or more of the core GML schemas defined in Clause 7.

In GML 2.x, GML Application Schemas were restricted to the development of vocabularies of geographic features and all application schemas were based on the feature.xsd.

In GML 3., this concept is extended and several different types of application schemas can be constructed including Feature Schemas, Coverage Schemas, Observation Schemas, Topology Schemas, Coordinate System Schemas, and Value Schemas. The rules for the construction of each of these types of schemas are discussed in the following clauses.

GML also allows the derivation of many other kinds of elements such as new units of measure, new geometry properties and new geometries. While these elements can be packaged into separate schemas they are viewed as subordinate to the schema categories of this clause. Any conformant GML application schema MUST be one of the schema types listed below, or be a schema document that complies to the rules from the respective schema clause. It is thus permissible to create a GML application schema that defines Features, Coverages and Values, so long as this schema satisfies the rules of Clauses 8.3, 8.4 and 8.6.

Note that elements in the content of complex types that are defined with local names in an application schema will prevent derivation by restriction in another namespace. Such complex types are appropriate for elements intended for use “as is” in their own namespace, and should be declared to be final=”restriction”. Elements in the content of complex types defined by reference to global elements support derivation by restriction in another namespace, allowing restriction of cardinality, and/or replacement by a member of a substitution group. Such complex types designed for derivation by restriction are appropriate “library types” for elements in substitution groups that cross namespaces.

Also note that the required import of a gml schema in clauses 8.*.2 below may be provided indirectly via the import of another schema in the namespace for gml that includes the required gml schema. For example, the import of gml.xsd would satisfy any of these schema import requirements. In addition, the required import of a gml schema may be provided by the import of an equivalent subset schema as described in clause 7.14 and Annex F, or by the import of an equivalent schema from a GML profile that defines the complex type of the document root element. These are all equivalent schemas with respect to satisfying the schema import requirements.

8.3 Schemas defining Features and Feature Collections

Features and Feature collections are the primary view of geospatial information supported by GML, and are particularly useful in modeling real world geography or in defining message types for geographic web services. A Feature in GML models a real world object or concept and provides an element (feature type) and an associated set of properties describing that type also encoded as elements.

Schemas supporting features and feature collections were introduced in GML 2.x. Feature application schemas define geographic features and feature collections for a specific application domain or community . These application schemas must obey the rules described in the following clauses.

8.3.1 Target Namespace

The application schema must declare a target namespace. This is the namespace in which the terms (features, feature collections) of the vocabulary "live". This must not be the GML namespace (http://www.opengis.net/gml). It is conventional for the namespace identifier to be a URL controlled by the application schema author's organization. A target namespace is declared in the application schema using the targetNamespace attribute of the schema element from XML Schema.

8.3.2 Import feature.xsd

The application schema must import the necessary components from GML 3. In order to define feature types for the application domain, it is necessary to import feature.xsd or equivalent schema either directly, for example as follows:

```
<xs:import namespace="http://www.opengis.net/gml" schemaLocation="../ feature.xsd"/>
```

or indirectly, by importing one of the GML schema documents which includes feature.xsd transitively.

Note that the <import> element specifies that the components described in feature.xsd are associated with the GML namespace http://www.opengis.net/gml. This namespace identifier must match the target namespace specified in the schema being imported in order to ensure XML Schema validity.

The path (schemaLocation) to feature.xsd can be to a local copy of the document, or may be a URI reference to feature.xsd in some remote repository, such as the repository http://schemas.opengis.net/ on the OGC web site

8.3.3 Features must derive from gml:AbstractFeatureType

All geographic features in the application schema must be declared as global elements in the schema, i.e. they must be immediate child elements of the XML Schema <schema> element. The content model for such global elements must derive either directly or indirectly from gml:AbstractFeatureType.

8.3.4 Feature Collections must derive from gml:AbstractFeatureCollectionType

All geographic feature collections in the application schema must be declared as global elements in the schema, that is they must be child elements of the XML Schema <schema> element. The content model for such global elements must derive either directly or indirectly from gml:AbstractFeatureCollectionType.

8.3.5 Properties representing associations

Elements representing properties of features may be declared as global elements in an application schema, or they may be declared locally within feature content models (type definitions). The type for a property element may be derived from gml:AssociationType or gml:ArrayAssociationType, or may follow the pattern of gml:AssociationType.

If the value of the property is expected to be available elsewhere, it is necessary for the property element in an instance to provide a pointer to the value, so the property must support association by-reference. This is accomplished using xlink attributes, and in the GML application schema the XML type for the property must include gml:AssociationAttributeGroup.

If the value of the property is expected to be represented in place, the content model (XML schema type) for the property must support this. The property element must either have XML Schema simpleContent of the appropriate simpleType, or it must have complexContent comprising a **single** child element of the appropriate type.

If the value of the property is expected to be available either elsewhere, or represented in place, then the type for the property element must support both methods. The type for the property element must have the gml:AssociationAttributeGroup, in which all members are optional, and the child element must have minOccurs="0" so that in an instance document the property element may be empty if it carries an xlink. Note that the value of a property of an ArrayAssociationType must be represented in place.

If the last pattern is used, and it is desired to prohibit the possibility of both xlink attributes and content, or neither, then this constraint should be recorded as a normative directive in an <annotation> element on the element declaration in the application schema. The directive may be expressed as prose, or it may be expressed using a formal notation such as Schematron.

8.3.6 Application Features are Features

A feature defined in an application schema must conform to the rules respecting GML features as described in Clause 7.2. These include in particular:

1. The name of feature element is the semantic type of the feature.
2. The children of a feature are always properties that describe the feature, and such properties can only be encoded as child elements. Properties cannot be encoded as XML attributes.

8.3.7 GML Feature Collection Document

A GML application may use a gml:FeatureCollection to contain features defined in an application schema when there is no requirement to restrict the inclusion of other types of features. A GML Application Schema may define specialised feature collection types and global elements to serve as the root elements for documents describing Feature Collections specific to the application domain where there is a requirement to exclude features not defined in that domain.

8.4 Schemas defining Coverages

Coverages are an alternative view of geospatial information. This view focuses on the variation of a property or properties across a domain, so is particularly useful in analysis.

This clause defines the rules for the construction of application schemas for coverages. Note that coverages are features and hence the rules of Clause 7.1 apply also to coverages.

8.4.1 Target Namespace

The application schema must declare a target namespace. This is the namespace in which the terms (features, feature collections) of the vocabulary live. This must not be the GML namespace (http://www.opengis.net/gml).It is conventional for the namespace identifier to be a URL based on a domain controlled by the application schema author's organization. A target namespace is declared in the application schema using the targetNamespace attribute of the schema element from XML Schema.

8.4.2 Import coverage.xsd

The application schema must import the necessary components from GML 3. In order to define coverages for the application domain, it is necessary to import coverage.xsd either directly, for example as follows:

```
<xs:import namespace="http://www.opengis.net/gml" schemaLocation="../coverage.xsd"/>
```

or indirectly, by importing one of the GML schema documents which includes coverage.xsd transitively.

Note that the <import> element specifies that the components described in coverage.xsd are associated with the GML namespace http://www.opengis.net/gml. This namespace identifier must match the targetnamespace specified in the schema being imported in order to ensure XML Schema validity.

The path (schemaLocation) to coverage.xsd can be to a local copy of the document, or may be a URI reference to feature.xsd in some remote repository, such as the repository http://schemas.opengis.net/ on the OGC web site.

8.4.3 Coverages must derive from gml:AbstractCoverageType

All geographic coverages in the application schema must be declared as global elements in the schema, that is they must be child elements of the XML Schema <schema> element. The content model for such global elements must derive by extension either directly or indirectly from gml:AbstractCoverageType. Note that gml:AbstractCoverageType itself derives from gml:AbstractFeatureType and hence the condition of Clause 7.2.3. is satisfied.

The coverage.xsd provides the specific coverage types (MultiPointCoverage, MultiSurfaceCoverage, GriddedCoverage, RectifiedGridCoverage) and application coverages can derive from any of these as well.

8.4.4 Range Parameters Must be Derived from gml:ValueType

The coverage schema must define or import the definitions for all Range Parameters. Each such Range Parameter must be substitutable for gml:_Value as defined in the valueObjects.xsd schema. Note that this allows the Range Parameter to take on a wide range of types. Note further that the value,xsd provides several abstract sub-types that are substitutable for gml:_Value, including gml:_ScalarValue and gml:_ValueList. Concrete scalar and value list types, and substitution group head elements, are also provided (substitutable for gml:_ScalarValue or gml:_ValueList) and include:

- gml:Category (gml:CodeType)
- gml:CategoryList (gml:CodeOrNullListType)
- gml:Quantity (gml:MeasureType)
- gml:QuantityList (gml:MeasureOrNullListType)
- gml:Count (gml:CountType)
- gml:Boolean (gml:BooleanType)

Please consult the valueObjects.xsd schema described in Clause 7.10.4 if you are writing a Coverage Application Schema.

Typical examples of the use of the value types in the development of a GML coverage can be found in the examples clause for coverages Clause 7.13.5, and are summarized in Table 8.4.1.

Coverage	Range Parameter	Definition in GML
Temperature Distribution (weather)	temperature	Would be derived from gml:MeasureType and made substitutable for gml:measure defined in the measures.xsd.
Soil type distribution (agronomy)	Soil type	Would be derived from gml:Category and made substitutable for gml_Category. Weak reference to an enumeration of soil types.
Multi-spectral optical image (remote sensing)	Reflectance in each spectral band.	Would be derived from gml:QuantityType or gml:QuantityListType and made substitutable for gml:Quantity or gml:QuantityList as appropriate.
Distribution of West Nile Virus cases. (epidemiology)	CaseCount	Would be derived from gml:ObservableCountType and made substitutable for gml:Count.

Table 8.4.1 Range Parameters for Coverage Schemas

8.4.5 Coverage Document

A coverage document is defined by a corresponding coverage schema. The root element of this document must be a coverage defined in this schema or may be a feature collection whose members are coverages as described in Clause 7.2.6.

8.5 Schemas defining Observations

Recall from Clause 7.x that an observation in GML is a kind of feature that represents an act of observing or observation event. . This may be associated with a measurement obtained from some type of instrument, or may just be a photograph acquired by a travelling tourist.

This clause describes how to create an observation application schema.

An observation application schema defines one or more types of observations according to the following rules.

8.5.1 Target Namespace

The application schema must declare a target namespace. This is the namespace in which the kinds of observations defined in the application schema are said to "live". This cannot be the GML namespace (http://www.opengis.net/gml) and should be based on a domain controlled by the application schema author's organization. A target namespace is declared in the application schema using the targetNamespace attribute of the schema element from XML Schema.

8.5.2 Import observation.xsd

The application schema must import the GML 3. observation.xsd schema with the correct namespace assignment, e.g.

```
<xs:import namespace="http://www.opengis.net/gml" schemaLocation="../ observation.xsd"/>
```

The path (schemaLocation) to the observation.xsd can be a local reference, or a URI reference to the observation.xsd on the OGC web site.

8.5.3 Observations must derive from gml:ObservationType

All observation types (kinds of observations) defined in the application schema must be declared as global elements in the schema, that is they must be child elements of the XML Schema <schema> element. The content model for such global elements must derive by extension either directly or indirectly from gml:ObservationType.

8.5.4 Observation Collections must derive from gml:CollectionType

All observation collections in the application schema must be declared as global elements in the schema, that is they must be child elements of the XML Schema <schema> element. The content model for such global elements must derive by extension either directly or indirectly from gml:CollectionType.

8.5.5 Observations are Features

An observation defined in an application schema must conform to the rules respecting GML features as described in Clause 7.2. These include in particular:

1. Observation semantic type information can ONLY be carried by elements. The following is not valid GML.

   ```
   <abc:Measurement type =”WaterQuality”> … </abc:Measurement>
   ```

2. The children of a observation are always properties that describe the observation, and such properties can only be encoded as child elements. Properties cannot be encoded as XML attributes.

3. No child of an observation can itself be a feature or other GML object (i.e. one derived from gml:AbstractGMLType). Thus no child of a observation can be feature, geometry, topology, coordinate reference system etc.

8.5.6 Observation Collection Document

Corresponding to a Observation Collection document there must be a GML Application Schema that defines the single root element of that Observation Collection document. Note that this does not imply that this Observation Collection is defined by a single application schema. The features referenced from the Observation Collection element may be contained in any number of other schemas and these may define observations only, observation collections or any combination of the same.

An observation collection could be used, for example, to encode a set of measurements from one or more sensor devices.

8.6 Schemas defining Dictionaries and Definitions

8.6.1 Overview

A common requirement is to collect a set of definitions together into a dictionary, in order that a term may be referred to many times, while its (potentially lengthy) definition is only recorded once. The kind of information that might form the content of a dictionary are units of measure, coordinate reference systems, observable-types or measurands, parties including individuals or organizations.

In order to support this, some generic components for dictionaries and definitions are provided in GML 3. The basic Definition is a simple element which supports a prose description. This may be used directly for simple non-parameterised definitions, or may serve as the basis for specialized definition elements.

The basic Dictionary is a bag of Definitions. It may be used as a container for an arbitrary set of Definitions or elements that are in the Definition substitution group. It may also serve as the basis for a specialized Dictionary restricted to contain only certain types of definition.

One set of specialized definitions is built in to GML 3, for units of measure, and serves as an example of how to derive specialized definition components.

This clause describes how to create an application schema for definitions.

An application schema for definitions defines one or more types of definitions according to the following rules.

8.6.2 Target Namespace

The application schema must declare a target namespace. This is the namespace in which the terms (definitions, dictionaries) for this application live. This must not be the GML namespace (http://www.opengis.net/gml). It is conventional for the namespace identifier to be a URL based on a domain controlled by the application schema author's organization. A target namespace is declared in the application schema using the targetNamespace attribute of the schema element from XML Schema.

8.6.3 Import dictionary.xsd

The application schema must import the necessary components from GML 3. In order to define feature types for the application domain, it is necessary to import dictionary.xsd either directly, for example as follows:

```
<xs:import namespace="http://www.opengis.net/gml" schemaLocation="../dictionary.xsd"/>
```

or indirectly, by importing one of the GML schema documents which includes gmlBase.xsd transitively.

Note that the <import> element specifies that the components described in gmlBase.xsd are associated with the GML namespace http://www.opengis.net/gml. This namespace identifier must match the targetnamespace specified in the schema being imported in order to ensure XML Schema validity.

The path (schemaLocation) to gmlBase.xsd can be to a local copy of the document, or may be a URI reference to gmlBase.xsd in some remote repository, such as the repository http://schemas.opengis.net/ on the OGC web site.

8.6.4 Definitions must derive from gml:DefinitionType

All definitions in the application schema must be declared as global elements in the schema, i.e. they must be immediate child elements of the XML Schema <schema> element. The content model for such global elements must derive either directly or indirectly from gml:DefinitionType.

8.6.5 Dictionaries must derive from gml:DictionaryType

A dictionary in the application schema must be declared as a global element in the schema, that is it must be a child element of the XML Schema <schema> element. The content model for such global elements must derive either directly or indirectly from gml:DictionaryType.

8.7 Schemas defining Coordinate Reference Systems

Follow the directions in 02-036r6 Annex A or its successor.

8.8 Schemas defining Values

GML allows for user defined value types. Such values types can be used to express the property types of features and other types of GML objects. The basic root types for user-defined values are defined in basicTypes.xsd. An alternative form for the expression of values is contained in values.xsd. This is used mainly to provide values for the gml:resultOf parameter for an observation.

8.8.1 Target Namespace

The application schema must declare a target namespace. This is the namespace in which the terms values of the vocabulary are said to “live”. This cannot be the GML namespace (http://www.opengis.net/gml) and should be based on a domain controlled by the application schema author's organization. A target namespace is declared in the application schema using the targetNamespace attribute of the schema element from XML Schema.

8.8.2 Import valueObjects.xsd or basicTypes.xsd

The application schema must import the GML 3., valueObjects.xsd or basicTypes.xsd schema with the correct namespace assignment, e.g.

```
<xs:import namespace="http://www.opengis.net/gml" schemaLocation="../ valueObjects.xsd"/>
```

The path (schemaLocation) to the valueObjects.xsd (basicTypes.xsd) can be a local reference or a URI reference to the valueObjects.xsd (basicTypes.xsd) on the OGC web site.

8.8.3 Construction of New Value Types

New value types can be created by derivation (typically by restriction) from any of the root types shown in Table 8.8.1.

Table 8.8.1

Content Model	Description
MeasureType	A numerical quantity with a unit of measure (UOM)
CategoryType	A classification
CountType	A count of occurrences, incidences etc.

Some standard value types can be found in the measures.xsd schema.

8.9 Summary of GML3 components for application schemas

GML 3 provides a large number of components that may be used in application schemas. These include the following general classes:

1. Concrete elements and attributes in the GML namespace, which can appear directly in conformant instance documents, and may be used directly in content models in GML application schemas. E.g. gml:location, gml:name, gml:description, gml:Observation, gml:id, gml:Dictionary, gml:FeatureCollection.

2. Concrete and abstract elements in the gml namespace may act as the head of a substitution group. An element declared in an application schema may be assigned to a substitution group whose head is a GML element. The element declared in the application schema may then appear in instance documents in place of the substitution group head and be conformant to the content model that

refers to the substitution group head. Note that in order to be a valid member of a substitution group, the type of the element must be validly derived from the type of the element which is the head of the substitution group. All abstract elements in the GML schemas are only useful acting as the heads of substitution groups. E.g. gml:_GML, gml:_Feature,_gml:_Geometry, gml:_TimeObject

3. Concrete types in the gml namespace, which may be used directly as the content model of a concrete element in an application schema. The element declared in the application schema will be in a different namespace, and may be used in an instance document. E.g. gml:EnvelopeType may be used unmodified as the content model for an element xmml:Interval.

4. Concrete and abstract types in the gml namespace which act as the basis for the definition of more specialised types that define the content models of elements in an application schema. E.g. gml:AbstractGeometryType, gml:ObservationType, gml:DefinitionType.

The last group includes the root types for the GML syntax which instantiate the Feature-property model. The most important of these are gml:AbstractGMLType, gml:AbstractFeatureType and gml:AssociationType.

All feature types defined in GML application schemas *must* be (ultimately) *derived* from AbstractFeatureType, and thereby inherit as optional properties a unique label *id*, a general metadata container *metaDataProperty*, plus named metadata properties *name*, *description*, *boundedBy*, and *location*.

Other important top-level types include CollectionType, ArrayType, AbstractGeometryType, AbstractCoverageType, ObservationType, and various propertyTypes.

ANNEXES

Annex A
(normative)

Abstract Test Suite

A.1 Conformance Class A. Conformance of the XML Implementation of GML data

A.1.1 Existence of an applicable GML application schema.

a) Test Purpose: To verify the existence of a GML application schema applicable to the GML data set.

b) Test Method: Ensure that the application schema applicable to a particular GML data set exists.

c) Reference:

d) Test Type: Basic Test

A.1.2 Conformance of the applicable application schema

a) Test Purpose: To verify that the applicable GML application schema is conformant to this specification.

b) Test Method: Test whether the application schema satisfies all the tests specified in the Conformance Class B.

c) Reference:

d) Test Type: Capability Test.

A.1.3 Conformance of the data set

a) Test Purpose: To verify the validity of the GML data against the conformant GML application schema.

b) Test Method: Validate GML Data set against corresponding GML application schema tested in the steps A1.1-A1.2. The process may be using an appropriate software tool for validation or be a manual process that checks all relevant definitions from the GML and application schemas.

c) Reference:

d) Test Type: Capability Test

A.2 Conformance Class B. Conformance of the XML Implementation of GML Application Schema

A.2.1 XMLSchema valid

a) Test Purpose: To verify the conformance of the GML application schema to the XML and XMLSchema specifications.

b) Test Method: Validate GML application schema using appropriate process for validation. The process may be using an appropriate software tool for validation or be a manual process that checks all relevant rules from the XML 1.0 and XMLSchema 1.0 specifications.

c) Reference:

d) Test Type: Capability Test

A.2.2 GML model valid

a) Test Purpose: To verify the validity against GML model.

b) Test Method: Ensure that the schema conforms to Clause 8 of this specification (Rules for creating GML Application Schemas) and 7.13 (Profiles).

c) Reference:

d) Test Type: Basic Test

A.3 Conformance Class C. Conformance of the GML Interface Implementation for GML Data

A.3.1 Serialization capability test

a) Test Purpose: To verify the existence of the serialization operation of the implementation.

b) Test Method: Verify that the implementation implements the serialize operation that produces GML in XML format.

c) Reference:

d) Test Type: Basic Test

A.3.2 Serialization validity test

a) Test Purpose: To verify that the result of the implementation serialization is conformant with this specification.

b) Test Method: Verify that the result of the serialization of the implementation conforms to this specification according to the Conformance Class A (A.1)

c) Reference:

d) Test Type: Capability Test

A.3.3 Transforming to and from XML format

a) Test Purpose: To verify that the creation of implementation's objects from GML source in XML format, if that capability exists, performs valid function.

b) Test Method: If it is possible to create implementation's objects from GML source in XML format, verify that successive actions of object creation and serialization produce (test A.3.2) the result that is identical to the source (no loss of information).

c) Reference:

d) Test Type: Capability Test

A.4 Conformance Class D. Conformance of the GML Interface Implementation for GML Application Schema

A.4.1 Serialization capability test

a) Test Purpose: To verify the existence of the serialization operation of the implementation.

b) Test Method: Verify that the implementation implements the serialize operation that produces GML Application Schema in XML Schema format.

c) Reference:

d) Test Type: Basic Test

A.4.2 Serialization validity test

a) Test Purpose: To verify that the result of the implementation serialization is conformant with this specification.

b) Test Method: Verify that the result of the serialization of the implementation conforms to this specification according to the Conformance Class B (A.2)

c) Reference:

d) Test Type: Capability Test

A.4.3 Transforming to and from XML format

a) Test Purpose: To verify that the creation of implementation schema objects from GML Application Schema source in XML format, if that capability exists, performs valid function.

b) Test Method: If it is possible to create implementation's schema objects from GML Application Schema source in XML format, verify that successful actions of object creation and serialization produce (test A.4.2) the result that is identical to the source (without loss of information).

c) Reference:

d) Test Type: Capability Test

Annex B
(normative)

Conformance

B.1 GML Object Rules

This clause summarizes the rules for the detection of GML objects. These rules specify the necessary and sufficient conditions for an element.

B.1.1 GML Objects

An element x is a GML object (of some kind) iff its content model derives ultimately from gml:AbstractGMLType, so that it may have identity provided by a gml:id attribute.

An element x is a more specific kind of GML object iff its content model derives ultimately from a more specific GML complex type derived from AbstractGMLType. For example, an element x is a GML feature iff its content model derives ultimately from gml:AbstractFeatureType. An element x is a GML geometry iff its content model derives ultimately from gml:AbstractGeometryType. And so on.

B.1.2 GML Properties

An element p is a property of a GML object x iff p is a child of x in the XML sense.

B.2 Application Schemas

Writers of application schemas must conform to the rules stated in clauses 7 and 8.

B.3 Conformance Classes

B.3.1 Conformance Requirements

Conformance with this specification shall be checked using all the relevant tests specified in Annex A. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance, are specified in ISO 19105: Geographic information — Conformance and Testing.

This OpenGIS® specification defines four conformance classes: A, B, C and D. Any product claiming conformance with one of these classes shall pass all the requirements described in the corresponding abstract test suite specified in the Annex A.

1. **Class A: Conformance of the XML Implementation of GML Data Set.**
   This class applies to GML data encoded using XML.

2. **Class B: Conformance of the XML Implementation of GML Application Schemas.**
   This class applies to GML application schemas encoded using XML Schema schema description language.
3. **Class C: Conformance of Interface Implementations for GML Data.**
   This class applies to implementations of software interfaces that consume and/or produce GML data.
4. **Class D: Conformance of Interface Implementations for GML Application Schemas.**
   This class applies to implementations of software interfaces that consume and/or produce GML application schemas.

Annex C
(normative)

GML Schemas

gml.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:gml:v3.00">gml.xsd</appinfo>
    <documentation>
		Copyright (c) 2002 OGC, All Rights Reserved.
		Top level GML schema
	</documentation>
  </annotation>
  <!-- ====================================================================== -->
  <include schemaLocation="dynamicFeature.xsd"/>
  <include schemaLocation="topology.xsd"/>
  <include schemaLocation="coverage.xsd"/>
  <include schemaLocation="coordinateReferenceSystems.xsd"/>
  <include schemaLocation="observation.xsd"/>
  <include schemaLocation="defaultStyle.xsd"/>
  <!-- ====================================================================== -->
</schema>
```

gmlBase.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:gmlBase:v3.00">
      <sch:title>Schematron validation</sch:title>
      <sch:ns prefix="gml" uri="http://www.opengis.net/gml"/>
      <sch:ns prefix="xlink" uri="http://www.w3.org/1999/xlink"/>
      <sch:pattern name="Check either href or content not both">
        <sch:rule abstract="true" id="hrefOrContent">
          <sch:report test="@xlink:href and (*|text())">
            Property element may not carry both a reference to an object and contain an object.</sch:report>
          <sch:assert test="@xlink:href | (*|text())">
            Property element must either carry a reference to an object or contain an object.</sch:assert>
        </sch:rule>
      </sch:pattern>
    </appinfo>
    <documentation>
		GML base schema for GML 3.0
		Components to support the GML encoding model.
```



```
              The abstract Schematron rules can be used by any schema that includes gmlBase.
    </documentation>
  </annotation>
  <!-- ================================================================
       includes and imports
       ================================================================ -->
  <include schemaLocation="basicTypes.xsd"/>
  <import namespace="http://www.w3.org/1999/xlink" schemaLocation="../xlink/xlinks.xsd"/>
  <!-- ============================================================== -->
  <!-- =================== 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>
  <!-- ============================================================== -->
  <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 may have an ID and are thus referenceable.  </documentation>
    </annotation>
    <sequence>
      <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
      <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>
    <attribute ref="gml:id" use="optional"/>
  </complexType>
  <!-- ============================================================== -->
  <!-- =========== Concrete "Collection" supertype ======================== -->
  <element name="Bag" type="gml:BagType" substitutionGroup="gml:_GML">
    <annotation>
      <documentation>Generic GML element to contain a heterogeneous collection of GML
_Objects</documentation>
    </annotation>
  </element>
  <!-- ============================================================== -->
  <complexType name="BagType">
```



```
    <annotation>
      <documentation>A non-abstract generic collection type that can be used as a document element for a 
collection of any GML types - Geometries, Topologies, Features ...

“FeatureCollections” may only contain Features.  “GeometryCollections” may only contain Geometry’s.  
“Bags” are less constrained – they must contain objects that are substitutable for gml:_Object.  This may mix 
several levels, including Features, Definitions, Dictionaries, Geometries etc.  

The content model would ideally be 
   member 0..*
   members 0..1
   member 0..*
for maximum flexibility in building a collection from both homogeneous and distinct components: 
included "member" elements each contain a single Object
an included "members" element contains a set of Objects 

However, this is non-deterministic, thus prohibited by XSD.  

</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGMLType">
        <sequence>
          <element ref="gml:member" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:members" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <!-- =========== Concrete "Array" supertype ========================= -->
  <element name="Array" type="gml:ArrayType" substitutionGroup="gml:_GML">
    <annotation>
      <documentation>Generic GML element to contain a homogeneous array of GML 
_Objects</documentation>
    </annotation>
  </element>
  <!-- =========================================================================== -->
  <complexType name="ArrayType">
    <annotation>
      <documentation>A non-abstract generic collection type that can be used as a document element for a 
homogeneous collection of any GML types - Geometries, Topologies, Features ...</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGMLType">
        <sequence>
          <element ref="gml:members" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <!-- =========== Abstract Metadata supertype ========================= -->
  <element name="_MetaData" type="gml:AbstractMetaDataType" abstract="true" 
substitutionGroup="gml:_Object">
    <annotation>
      <documentation>Abstract element which acts as the head of a substitution group for packages of 
MetaData properties.  </documentation>
    </annotation>
  </element>
```

```
  <!-- ============================================================ -->
  <complexType name="AbstractMetaDataType" abstract="true" mixed="true">
    <annotation>
      <documentation> An abstract base type for complex metadata types.</documentation>
    </annotation>
    <attribute ref="gml:id" use="optional"/>
  </complexType>
  <!-- ============================================================ -->
  <!-- =========== Container for Generic Metadata ======================== -->
  <element name="GenericMetaData" type="gml:GenericMetaDataType"
substitutionGroup="gml:_MetaData">
    <annotation>
      <documentation>Concrete element in the _MetaData substitution group, which permits any well-formed
XML content.  Intended to act as a container for metadata defined in external schemas, for which it is not
possible to add the concrete components to the GML _MetaData substitution group directly.
</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <complexType name="GenericMetaDataType" mixed="true">
    <complexContent mixed="true">
      <extension base="gml:AbstractMetaDataType">
        <sequence>
          <any processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <!-- ==================== Properties ============================== -->
  <!-- ============================================================ -->
  <!-- ===== Global element at the head of the "property" substitution group ============ -->
  <!-- element name="_property" type="anyType" abstract="true" -->
  <element name="_property" abstract="true">
    <annotation>
      <documentation>This abstract element may act as the head of a substitutionGroup hierararchy which
may contain either simpleContent or complexContent elements.  It may be used to assert the model position
of "property" elements declared in other GML schemas.  </documentation>
    </annotation>
  </element>
  <!-- ================= Base Property Types ============================== -->
  <!-- ================================================================== -->
  <!-- ==== property types for unspecified association - by Value or by Reference ==== -->
  <!-- ====== single Objects - by Value or by Reference ======== -->
  <element name="_association" type="gml:AssociationType" abstract="true"/>
  <!-- ============================================================ -->
  <element name="_strictAssociation" type="gml:AssociationType" abstract="true">
    <annotation>
      <appinfo>
        <sch:pattern name="refAndContent co-occurence prohibited">
          <sch:rule context="gml:_strictAssociation">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
      <documentation>must carry a reference to an object or contain an object but not both</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <element name="member" type="gml:AssociationType"/>
```

```xml
  <!-- ============================================================== -->
  <complexType name="AssociationType">
    <annotation>
      <documentation>  A pattern or base for derived types used to specify complex types corresponding to an  unspecified UML association - either composition or aggregation.  Restricts the cardinality of Objects contained in the association to a maximum of one.  An instance of this type can contain an element representing an Object, or serve as a pointer to a remote Object.

Descendents of this type can be restricted in an application schema to
* allow only specified classes as valid participants in the aggregation
* allow only association by reference (i.e. empty the content model) or by value (i.e. remove the xlinks).

When used for association by reference, the value of the gml:remoteSchema attribute can be used to locate a schema fragment that constrains the target instance.

In many cases it is desirable to impose the constraint prohibiting the occurence of both reference and value in the same instance, as that would be ambiguous.  This is accomplished by adding a directive in the annotation element of the element declaration.  This directive can be in the form of normative prose, or can use a Schematron pattern to automatically constrain co-occurrence - see the declaration for _strictAssociation below.

If co-occurence is not prohibited, then both a link and content may be present.  If this occurs in an instance, then the rule for interpretation is that the instance found by traversing the href provides the normative value of the property, and should be used when possible.  The value(s) included as content may be used if the remote instance cannot be resolved.  This may be considered to be a "cached" version of the value(s).
</documentation>
    </annotation>
    <sequence>
      <element ref="gml:_Object" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ============================================================== -->
  <element name="_reference" type="gml:ReferenceType" abstract="true"/>
  <!-- ============================================================== -->
  <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>
    <sequence/>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ============================================================== -->
  <!-- ========= multiple objects  - by Value or by Reference ================== -->
  <element name="members" type="gml:ArrayAssociationType"/>
  <!-- ============================================================== -->
  <complexType name="ArrayAssociationType">
    <annotation>
      <documentation>  A base for derived types used to specify complex types containing an array of objects, by unspecified UML association - either composition or aggregation.  An instance of this type contains elements representing Objects.

Ideally this type would be derived by extension of AssociationType.
However, this leads to a non-deterministic content model, since both the base and the extension have minOccurs="0", and is thus prohibited in XML Schema. </documentation>
    </annotation>
    <sequence>
      <element ref="gml:_Object" minOccurs="0" maxOccurs="unbounded"/>
```

```
    </sequence>
  </complexType>
  <!-- ============================================================== -->
  <!-- =========== Abstract "property" supertype ========================= -->
  <element name="metaDataProperty" type="gml:MetaDataPropertyType">
    <annotation>
      <documentation>Contains or refers to a metadata package that contains metadata properties.
</documentation>
    </annotation>
  </element>
  <!-- ============================================================== -->
  <complexType name="MetaDataPropertyType">
    <annotation>
      <documentation> Base type for complex metadata property types.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:_MetaData" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
    <attribute name="about" type="anyURI" use="optional"/>
  </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>
  <!-- ============================================================== -->
  <attribute name="remoteSchema" type="anyURI">
    <annotation>
      <documentation>Reference to an XML Schema fragment that specifies the content model of the
property's value. This is in conformance with the XML Schema Section 4.14 Referencing Schemas from
Elsewhere. </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", and the remoteSchema
attribute, which is also optional.  These attributes can be attached to any element, thus allowing it to act as a
pointer. The 'remoteSchema' attribute allows an element  that carries link attributes to indicate that the
element is declared  in a remote schema rather than by the schema that constrains the current document
instance. </documentation>
    </annotation>
    <attributeGroup ref="xlink:simpleLink"/>
    <attribute ref="gml:remoteSchema" use="optional"/>
  </attributeGroup>
  <!-- ============================================================== -->
  <element name="name" type="gml:CodeType">
    <annotation>
      <documentation>Identifier 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="gml:StringOrRefType">
    <annotation>
      <documentation>Contains a simple text description of the object, or refers to an external description. </documentation>
    </annotation>
  </element>
  <!-- ========================================================= -->
  <complexType name="StringOrRefType">
    <annotation>
      <documentation>
This type is available wherever there is a need for a "text" type property. It is of string type, so the text can be included inline, but the value can also be referenced remotely via xlinks from the AssociationAttributeGroup. If the remote reference is present, then the value obtained by traversing the link should be used, and the string content of the element can be used for an annotation.  </documentation>
    </annotation>
    <simpleContent>
      <extension base="string">
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </extension>
    </simpleContent>
  </complexType>
  <!-- ========================================================= -->
</schema>
```

basicTypes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema" xmlns:sch="http://www.ascc.net/xml/schematron" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:basicTypes:v3.00">basicTypes.xsd</appinfo>
    <documentation>
    Generic simpleContent components for use in GML
    </documentation>
  </annotation>
  <!-- ============================================================== -->
  <simpleType name="NullEnumeration">
    <annotation>
      <documentation> Some common reasons for a null value:

        innapplicable - the object does not have a value
        missing - The correct value is not readily available to the sender of this data.
                   Furthermore, a correct value may not exist.
        template - the value will be available later
        unknown - The correct value is not known to, and not computable by, the sender of this data.
                   However, a correct value probably exists.
        withheld - the value is not divulged

        Specific communities may agree to assign more strict semantics when these terms are used in a particular context.
      </documentation>
```



```
    </annotation>
    <restriction base="string">
      <enumeration value="inapplicable"/>
      <enumeration value="missing"/>
      <enumeration value="template"/>
      <enumeration value="unknown"/>
      <enumeration value="withheld"/>
    </restriction>
  </simpleType>
  <!-- ============================================================== -->
  <simpleType name="NullType">
    <annotation>
      <documentation>Utility type for null elements.  The value may be selected from one of the enumerated
tokens, or may be a URI in which case this should identify a resource which describes the reason for the
null. </documentation>
    </annotation>
    <union memberTypes="gml:NullEnumeration anyURI"/>
  </simpleType>
  <!-- ============================================================== -->
  <element name="Null" type="gml:NullType"/>
  <!-- ======================================================== -->
  <simpleType name="booleanOrNull">
    <annotation>
      <documentation>Union of the XML Schema boolean type and the GML Nulltype.  An element which
uses this type may have content which is either a boolean {0,1,true,false} or a value from
Nulltype</documentation>
    </annotation>
    <union memberTypes="gml:NullEnumeration boolean anyURI"/>
  </simpleType>
  <!-- ============================================================== -->
  <simpleType name="booleanOrNullList">
    <annotation>
      <documentation>XML List based on the union type defined above.  An element declared with this type
contains a space-separated list of boolean values {0,1,true,false} with null values interspersed as
needed</documentation>
    </annotation>
    <list itemType="gml:booleanOrNull"/>
  </simpleType>
  <!-- ============================================================== -->
  <simpleType name="booleanList">
    <annotation>
      <documentation>XML List based on XML Schema boolean type.  An element of this type contains a
space-separated list of boolean values {0,1,true,false}</documentation>
    </annotation>
    <list itemType="boolean"/>
  </simpleType>
  <!-- ============================================================== -->
  <simpleType name="stringOrNull">
    <annotation>
      <documentation>Union of the XML Schema string type and the GML Nulltype.  An element which uses
this type may have content which is either a string or a value from Nulltype.  Note that a "string" may contain
whitespace.  </documentation>
    </annotation>
    <union memberTypes="gml:NullEnumeration string anyURI"/>
  </simpleType>
  <!-- ============================================================== -->
  <simpleType name="NameOrNull">
    <annotation>
```

```
      <documentation>Union of the XML Schema Name type and the GML Nulltype.  An element which uses
this type may have content which is either a Name or a value from Nulltype.  Note that a "Name" may not
contain whitespace.  </documentation>
    </annotation>
    <union memberTypes="gml:NullEnumeration Name anyURI"/>
  </simpleType>
  <!-- ============================================================== -->
  <simpleType name="NameOrNullList">
    <annotation>
      <documentation>XML List based on the union type defined above.  An element declared with this type
contains a space-separated list of Name values with null values interspersed as needed</documentation>
    </annotation>
    <list itemType="gml:NameOrNull"/>
  </simpleType>
  <!-- ============================================================== -->
  <simpleType name="NameList">
    <annotation>
      <documentation>XML List based on XML Schema Name type.  An element of this type contains a
space-separated list of Name values</documentation>
    </annotation>
    <list itemType="Name"/>
  </simpleType>
  <!-- ============================================================== -->
  <simpleType name="doubleOrNull">
    <annotation>
      <documentation>Union of the XML Schema double type and the GML Nulltype.  An element which uses
this type may have content which is either a double or a value from Nulltype</documentation>
    </annotation>
    <union memberTypes="gml:NullEnumeration double anyURI"/>
  </simpleType>
  <!-- ============================================================== -->
  <simpleType name="doubleOrNullList">
    <annotation>
      <documentation>XML List based on the union type defined above.  An element declared with this type
contains a space-separated list of double values with null values interspersed as needed</documentation>
    </annotation>
    <list itemType="gml:doubleOrNull"/>
  </simpleType>
  <!-- ============================================================== -->
  <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>
  <!-- ============================================================== -->
  <simpleType name="integerOrNull">
    <annotation>
      <documentation>Union of the XML Schema integer type and the GML Nulltype.  An element which uses
this type may have content which is either an integer or a value from Nulltype</documentation>
    </annotation>
    <union memberTypes="gml:NullEnumeration integer anyURI"/>
  </simpleType>
  <!-- ============================================================== -->
  <simpleType name="integerOrNullList">
    <annotation>
      <documentation>XML List based on the union type defined above.  An element declared with this type
contains a space-separated list of integer values with null values interspersed as needed</documentation>
    </annotation>
```

```
    <list itemType="gml:integerOrNull"/>
  </simpleType>
  <!-- ============================================================= -->
  <simpleType name="integerList">
    <annotation>
      <documentation>XML List based on XML Schema integer type.  An element of this type contains a
space-separated list of integer values</documentation>
    </annotation>
    <list itemType="integer"/>
  </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="CodeListType">
    <annotation>
      <documentation>List of values on a uniform nominal scale.  List of text tokens.
      In a list context a token should not include any spaces, so xsd:Name is used instead of xsd:string.
      If a codeSpace attribute is present, then its value is a reference to
      a Reference System for the value, a dictionary or code list.</documentation>
    </annotation>
    <simpleContent>
      <extension base="gml:NameList">
        <attribute name="codeSpace" type="anyURI" use="optional"/>
      </extension>
    </simpleContent>
  </complexType>
  <!-- ============================================================= -->
  <complexType name="CodeOrNullListType">
    <annotation>
      <documentation>List of values on a uniform nominal scale.  List of text tokens.
      In a list context a token should not include any spaces, so xsd:Name is used instead of xsd:string.
      A member of the list may be a typed null.
      If a codeSpace attribute is present, then its value is a reference to
      a Reference System for the value, a dictionary or code list.</documentation>
    </annotation>
    <simpleContent>
      <extension base="gml:NameOrNullList">
        <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>
<!-- ============================================================ -->
<complexType name="MeasureListType">
  <annotation>
    <documentation>List of numbers with a uniform 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="gml:doubleList">
      <attribute name="uom" type="anyURI" use="required"/>
    </extension>
  </simpleContent>
</complexType>
<!-- ============================================================ -->
<complexType name="MeasureOrNullListType">
  <annotation>
    <documentation>List of numbers with a uniform scale.
    A member of the list may be a typed null.
    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="gml:doubleOrNullList">
      <attribute name="uom" type="anyURI" use="required"/>
    </extension>
  </simpleContent>
</complexType>
<!-- ============================================================ -->
<complexType name="CoordinatesType">
  <annotation>
    <documentation>Tables or arrays of tuples.
      May be used for text-encoding of values from a table.
      Actually just a string, but allows the user to indicate which characters are used as separators.
      The value of the 'cs' attribute is the separator for coordinate values,
      and the value of the 'ts' attribute gives the tuple separator (a single space by default);
      the default values may be changed to reflect local usage.
      Defaults to CSV within a tuple, space between tuples.
      However, any string content will be schema-valid.  </documentation>
  </annotation>
  <simpleContent>
    <extension base="string">
      <attribute name="decimal" type="string" default="."/>
      <attribute name="cs" type="string" default=","/>
      <attribute name="ts" type="string" default="&#x20;"/>
    </extension>
  </simpleContent>
</complexType>
<!-- ============================================================ -->
<simpleType name="SignType">
  <annotation>
    <documentation>Utility type used in various places
    - e.g. to indicate the direction of topological objects;
    "+" for forwards, or "-" for backwards.</documentation>
```

```
    </annotation>
    <restriction base="string">
      <enumeration value="-"/>
      <enumeration value="+"/>
    </restriction>
  </simpleType>
  <!-- ============================================================ -->
</schema>
```

dictionary.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:dictionary:v3.00"/>
    <documentation>
		Dictionary schema for GML 3.0
		Components to support the lists of definitions.
    </documentation>
  </annotation>
  <!-- ========================================================
      includes and imports
     ======================================================== -->
  <include schemaLocation="gmlBase.xsd"/>
  <!-- ====================================================== -->
  <!-- ====================================================== -->
  <!-- === Dictionary and Definition components === -->
  <!-- ====================================================== -->
  <element name="Definition" type="gml:DefinitionType" substitutionGroup="gml:_GML"/>
  <!-- ====================================================== -->
  <complexType name="DefinitionType">
    <annotation>
      <documentation>A definition, which can be included in or referenced by a dictionary. In this extended
type, the inherited "description" optional element can hold the definition whenever only text is needed. The
inherited "name" elements can provide one or more brief terms for which this is the definition. The inherited
"metaDataProperty" elements can be used to reference or include more information about this definition.
The gml:id attribute is required - it must be possible to reference this definition using this handle.
</documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:AbstractGMLType">
        <sequence>
          <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:description" minOccurs="0"/>
          <element ref="gml:name" maxOccurs="unbounded"/>
        </sequence>
        <attribute ref="gml:id" use="required"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ====================================================== -->
  <element name="Dictionary" type="gml:DictionaryType" substitutionGroup="gml:Definition"/>
  <element name="DefinitionCollection" type="gml:DictionaryType" substitutionGroup="gml:Definition"/>
  <!-- ====================================================== -->
  <complexType name="DictionaryType">
    <annotation>
```

```
    <documentation>A non-abstract bag that is specialized for use as a dictionary which contains a set of
definitions. These definitions are referenced from other places, in the same and different XML documents. In
this restricted type, the inherited optional "description" element can be used for a description of this
dictionary. The inherited optional "name" element can be used for the name(s) of this dictionary. The
inherited "metaDataProperty" elements can be used to reference or contain more information about this
dictionary. The inherited required gml:id attribute allows the dictionary to be referenced using this handle.
</documentation>
   </annotation>
   <complexContent>
    <extension base="gml:DefinitionType">
     <choice minOccurs="0" maxOccurs="unbounded">
      <element ref="gml:dictionaryEntry">
       <annotation>
        <documentation>An entry in this dictionary. The content of an entry can itself be a lower level
dictionary or definition collection. This element follows the standard GML property model, so the value may
be provided directly or by reference. Note that if the value is provided by reference, this definition does not
carry a handle (gml:id) in this context, so does not allow external references to this specific entry in this
context. When used in this way the referenced definition will usually be in a dictionary in the same XML
document. </documentation>
       </annotation>
      </element>
      <element ref="gml:indirectEntry">
       <annotation>
        <documentation>An identified reference to a remote entry in this dictionary, to be used when this
entry should be identified to allow external references to this specific entry. </documentation>
       </annotation>
      </element>
     </choice>
    </extension>
   </complexContent>
  </complexType>
  <!-- ====================================================== -->
  <element name="dictionaryEntry" type="gml:DictionaryEntryType"/>
  <element name="definitionMember" type="gml:DictionaryEntryType"
substitutionGroup="gml:dictionaryEntry"/>
  <!-- ====================================================== -->
  <complexType name="DictionaryEntryType">
   <annotation>
    <documentation>An entry in a dictionary of definitions. An instance of this type contains or refers to a
definition object.

The number of definitions contained in this dictionaryEntry is restricted to one, but a DefinitionCollection or
Dictionary that contains multiple definitions can be substituted if needed. Specialized descendents of this
dictionaryEntry might be restricted in an application schema to allow only including specified types of
definitions as valid entries in a dictionary. </documentation>
   </annotation>
   <sequence>
    <element ref="gml:Definition" minOccurs="0">
     <annotation>
      <documentation>This element in a dictionary entry contains the actual definition. </documentation>
     </annotation>
    </element>
   </sequence>
   <attributeGroup ref="gml:AssociationAttributeGroup">
    <annotation>
     <documentation>A non-identified reference to a remote entry in this dictionary, to be used when this
entry need not be identified to allow external references to this specific entry. The remote entry referenced
will usually be in a dictionary in the same XML document. This element will usually be used in dictionaries
that are inside of another dictionary. </documentation>
    </annotation>
```

```
    </attributeGroup>
  </complexType>
  <!-- ============================================================ -->
  <element name="indirectEntry" type="gml:IndirectEntryType"/>
  <!-- ============================================================ -->
  <complexType name="IndirectEntryType">
    <annotation>
      <documentation>An entry in a dictionary of definitions that contains a GML object which references a remote definition object. This entry is expected to be convenient in allowing multiple elements in one XML document to contain short (abbreviated XPointer) references, which are resolved to an external definition provided in a Dictionary element in the same XML document. Specialized descendents of this dictionaryEntry might be restricted in an application schema to allow only including specified types of definitions as valid entries in a dictionary. </documentation>
    </annotation>
    <sequence>
      <element ref="gml:DefinitionProxy"/>
    </sequence>
  </complexType>
  <!-- ============================================================ -->
  <element name="DefinitionProxy" type="gml:DefinitionProxyType" substitutionGroup="gml:Definition"/>
  <!-- ============================================================ -->
  <complexType name="DefinitionProxyType">
    <annotation>
      <documentation>A proxy entry in a dictionary of definitions. An element of this type contains a reference to a remote definition object. This entry is expected to be convenient in allowing multiple elements in one XML document to contain short (abbreviated XPointer) references, which are resolved to an external definition provided in a Dictionary element in the same XML document. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:DefinitionType">
        <sequence>
          <element ref="gml:definitionRef">
            <annotation>
              <documentation>A reference to a remote entry in this dictionary, used when this dictionary entry is identified to allow external references to this specific entry. The remote entry referenced can be in a dictionary in the same or different XML document. </documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="definitionRef" type="gml:ReferenceType"/>
  <!-- ===================================================================== -->
</schema>
```

units.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="3.00"
xml:lang="en">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-units:v3.00"/>
    <documentation>
      <name>units.xsd</name>
      <version>3.0</version>
```

```
      <scope>How to encode units of measure (or uom) for numerical values. </scope>
      <description>Builds on gmlBase.xsd to encode units of measure (or uom), including definitions of units
of measure and dictionaries of such definitions. GML 3.0 candidate schema, primary editor: Arliss Whiteside.
Last updated 2002/11/13. </description>
      <copyright>Copyright (c) 2001-2002 OpenGIS, All Rights Reserved.</copyright>
      <conformance>Parts of this schema are based on Subclause 6.5.7 of ISO/CD 19103 Geographic
information - Conceptual schema language, on Subclause A.5.2.2.3 of ISO/CD 19118 Geographic
information - Encoding, and on most of OpenGIS Recommendation Paper OGC 02-007r4 Units of Measure
Use and Definition Recommendations. </conformance>
    </documentation>
  </annotation>
  <!-- ==============================================================
       includes and imports
      ============================================================== -->
  <include schemaLocation="dictionary.xsd"/>
  <!-- ==============================================================
       elements and types
      ============================================================== -->
  <element name="unitOfMeasure" type="gml:UnitOfMeasureType"/>
  <!-- ============================================================== -->
  <complexType name="UnitOfMeasureType">
    <annotation>
      <documentation>Reference to a unit of measure definition that applies to all the numerical values
described by the element containing this element. Notice that a complexType which needs to include the
uom attribute can do so by extending this complexType. Alternately, this complexType can be used as a
pattern for a new complexType. </documentation>
    </annotation>
    <sequence/>
    <attribute name="uom" type="anyURI" use="required">
      <annotation>
        <documentation>Reference to a unit of measure definition, usually within the same XML document but
possibly outside the XML document which contains this reference. For a reference within the same XML
document, the "#" symbol should be used, followed by a text abbreviation of the unit name. However, the "#"
symbol may be optional, and still may be interpreted as a reference. </documentation>
      </annotation>
    </attribute>
  </complexType>
  <!-- ============================================================== -->
  <element name="UnitDefinition" type="gml:UnitDefinitionType" substitutionGroup="gml:Definition"/>
  <!-- ============================================================== -->
  <complexType name="UnitDefinitionType">
    <annotation>
      <documentation>Definition of a unit of measure (or uom). The definition includes a quantityType
property, which indicates the phenomenon to which the units apply, and a catalogSymbol, which gives the
short symbol used for this unit. This element is used when the relationship of this unit to other units or units
systems is unknown.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:DefinitionType">
        <sequence>
          <element ref="gml:quantityType"/>
          <element ref="gml:catalogSymbol" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================== -->
  <element name="BaseUnit" type="gml:BaseUnitType" substitutionGroup="gml:UnitDefinition"/>
  <!-- ============================================================== -->
  <complexType name="BaseUnitType">
```

```
    <annotation>
      <documentation>Definition of a unit of measure which is a base unit from the system of units.  A base unit cannot be derived by combination of other base units within this system.  Sometimes known as "fundamental unit".  </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:UnitDefinitionType">
        <sequence>
          <element name="unitsSystem" type="gml:ReferenceType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="DerivedUnit" type="gml:DerivedUnitType" substitutionGroup="gml:UnitDefinition"/>
  <!-- =============================================================== -->
  <complexType name="DerivedUnitType">
    <annotation>
      <documentation>Definition of a unit of measure which is defined through algebraic combination of more primitive units, which are usually base units from a particular system of units. Derived units based directly on base units are usually preferred for quantities other than the base units or fundamental quantities within a system.  If a derived unit is not the preferred unit, the ConventionalUnit element should be used instead. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:UnitDefinitionType">
        <sequence>
          <element ref="gml:unitDerivation"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="ConventionalUnit" type="gml:ConventionalUnitType" substitutionGroup="gml:UnitDefinition"/>
  <!-- ============================================================= -->
  <complexType name="ConventionalUnitType">
    <annotation>
      <documentation>Definition of a unit of measure which is related to a preferred unit for this quantity type through a conversion formula.  A method for deriving this unit by algebraic combination of more primitive units, may also be provided.  </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:UnitDefinitionType">
        <sequence>
          <choice>
            <element ref="gml:conversionToPreferredUnit"/>
            <element ref="gml:roughConversionToPreferredUnit"/>
          </choice>
          <element ref="gml:unitDerivation" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="quantityType" type="gml:StringOrRefType">
    <annotation>
      <documentation>Informal description of the phenomenon or type of quantity that is measured or observed. For example, "length", "angle", "time", "pressure", or "temperature". When the quantity is the
```

```
result of an observation or measurement, this term is known as Observable Type or Measurand.
</documentation>
    </annotation>
  </element>
  <!-- =========================================================== -->
  <element name="catalogSymbol" type="gml:CodeType">
    <annotation>
      <documentation>For global understanding of a unit of measure, it is often possible to reference an item
in a catalog of units, using a symbol in that catalog. The "codeSpace" attribute in "CodeType" identifies a
namespace for the catalog symbol value, and might reference the catalog. The "string" value in "CodeType"
contains the value of a symbol that is unique within this catalog namespace. This symbol often appears
explicitly in the catalog, but it could be a combination of symbols using a specified algebra of units. For
example, the symbol "cm" might indicate that it is the "m" symbol combined with the "c" prefix.
</documentation>
    </annotation>
  </element>
  <!-- =========================================================== -->
  <element name="unitDerivation" type="gml:UnitDerivationType"/>
  <!-- =========================================================== -->
  <complexType name="UnitDerivationType">
    <annotation>
      <documentation>Definition of the relationship of a derived unit of measure to the primitive units to which
that unit is related. Contains a set of one or more UnitTerms which are combined as a product, each of
which references a unit of measure defined elsewhere, and has an exponent. </documentation>
    </annotation>
    <sequence>
      <element ref="gml:unitTerm" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <!-- =========================================================== -->
  <element name="unitTerm" type="gml:UnitTermType"/>
  <!-- =========================================================== -->
  <complexType name="UnitTermType">
    <annotation>
      <documentation>Definition of one unit term for a derived unit of measure. This unit term references
another unit of measure (uom) and provides an integer exponent applied to that unit in defining the
compound unit. The exponent can be positive or negative, but not zero. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:UnitOfMeasureType">
        <attribute name="exponent" type="integer"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- =========================================================== -->
  <element name="conversionToPreferredUnit" type="gml:ConversionToPreferredUnitType">
    <annotation>
      <documentation>This element is included when this unit has an accurate conversion to the preferred
unit for this quantity type. </documentation>
    </annotation>
  </element>
  <!-- =========================================================== -->
  <element name="roughConversionToPreferredUnit" type="gml:ConversionToPreferredUnitType">
    <annotation>
      <documentation>This element is included when the correct definition of this unit is unknown, but this unit
has a rough or inaccurate conversion to the preferred unit for this quantity type. </documentation>
    </annotation>
  </element>
  <!-- =========================================================== -->
  <complexType name="ConversionToPreferredUnitType">
```



```
    <annotation>
      <documentation>Relation of a unit to the preferred unit for this quantity type, specified by an arithmetic conversion (scaling and/or offset). A preferred unit is either a base unit or a derived unit selected for all units of one quantity type. The mandatory attribute "uom" shall reference the preferred unit that this conversion applies to. The conversion is specified by one of two alternative elements: "factor" or "formula".
</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:UnitOfMeasureType">
        <choice>
          <element ref="gml:factor"/>
          <element ref="gml:formula"/>
        </choice>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="factor" type="double">
    <annotation>
      <documentation>Specification of the scale factor by which a value using this unit of measure can be multiplied to obtain the corresponding value using the preferred unit of measure. </documentation>
    </annotation>
  </element>
  <!-- ============================================================= -->
  <element name="formula" type="gml:FormulaType"/>
  <!-- ============================================================= -->
  <complexType name="FormulaType">
    <annotation>
      <documentation>Paremeters of a simple formula by which a value using this unit of measure can be converted to the corresponding value using the preferred unit of measure. The formula element contains elements a, b, c and d, whose values use the XML Schema type "double". These values are used in the formula y = (a + bx) / (c + dx), where x is a value using this unit, and y is the corresponding value using the preferred unit. The elements a and d are optional, and if values are not provided, those parameters are considered to be zero. If values are not provided for both a and d, the formula is equivalent to a fraction with numerator and denominator parameters. </documentation>
    </annotation>
    <sequence>
      <element name="a" type="double" minOccurs="0"/>
      <element name="b" type="double"/>
      <element name="c" type="double"/>
      <element name="d" type="double" minOccurs="0"/>
    </sequence>
  </complexType>
  <!-- ============================================================= -->
</schema>
```

measures.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified" version="3.00" xml:lang="en">
  <xsd:annotation>
    <xsd:appinfo source="urn:opengis:specification:gml:schema-measures:v3.00"/>
    <xsd:documentation>
      <name>measures.xsd</name>
      <version>3.0</version>
      <scope>How to encode measures, each with associated unit of measure (uom). </scope>
```

```
      <description>Extends the units.xsd and basicTypes.xsd schemas with types for recording measures
using specific types of units, especially the measures and units needed for coordinate reference systems
and coordinate operations. The specific unit types encoded are length, angle, scale factor, time, area,
volume, velocity, and grid length. This schema allows angle values to be recorded as single numbers or in
degree-minute-second format. GML 3.0 candidate schema, primary editor: Arliss Whiteside. Last updated
2002/11/13. </description>
      <copyright>Copyright (c) 2001-2002 OpenGIS, All Rights Reserved</copyright>
      <conformance>Parts of this schema are based on Subclause 6.5.7 of ISO/CD 19103 Geographic
information - Conceptual schema language, on Subclause A.5.2.2.3 of ISO/CD 19118 Geographic
information - Encoding, and on Subclause 4.7 of OpenGIS Recommendation Paper OGC 02-007r4 Units of
Measure Use and Definition Recommendations. </conformance>
    </xsd:documentation>
  </xsd:annotation>
  <!-- ==============================================================
      includes and imports
      ============================================================== -->
  <xsd:include schemaLocation="units.xsd"/>
  <!-- ==============================================================
      elements and types
      ============================================================== -->
  <!-- This schema uses the gml:MeasureType defined in basicTypes.xsd with the modified meaning:
             <documentation>Value of a quantity, with its units. This element uses the XML Schema primitive
data type "double" because it supports both decimal and scientific notation, and thus offers flexibility and
precision. However, there is no requirement to store values using any particular format, and applications
receiving elements of this type may choose to coerce the data to any other type as convenient. The XML
attribute uom references the units or scale by which the amount should be multiplied. For a reference within
the same XML document, the abbreviated XPointer prefix "#" symbol should be used, followed by a text
abbreviation of the unit name. However, the "#" symbol may be optional, and still may be interpreted as a
reference. </documentation> -->
  <!-- ============================================================ -->
  <xsd:element name="measure" type="gml:MeasureType"/>
  <!-- ============================================================ -->
  <xsd:complexType name="LengthType">
    <xsd:annotation>
      <xsd:documentation>Value of a length (or distance) quantity, with its units. Uses the MeasureType with
the restriction that the unit of measure referenced by uom must be suitable for a length, such as metres or
feet. </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="gml:MeasureType"/>
    </xsd:simpleContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:complexType name="ScaleType">
    <xsd:annotation>
      <xsd:documentation>Value of a scale factor (or ratio) that has no physical unit. Uses the MeasureType
with the restriction that the unit of measure referenced by uom must be suitable for a scale factor, such as
percent, permil, or parts-per-million. </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="gml:MeasureType"/>
    </xsd:simpleContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:complexType name="TimeType">
    <xsd:annotation>
      <xsd:documentation>Value of a time or temporal quantity, with its units. Uses the MeasureType with the
restriction that the unit of measure referenced by uom must be suitable for a time value, such as seconds or
weeks. </xsd:documentation>
    </xsd:annotation>
```

```
   <xsd:simpleContent>
     <xsd:extension base="gml:MeasureType"/>
   </xsd:simpleContent>
 </xsd:complexType>
 <!-- ============================================================== -->
 <xsd:complexType name="GridLengthType">
   <xsd:annotation>
     <xsd:documentation>Value of a length (or distance) quantity in a grid, where the grid spacing does not
have any associated physical units, or does not have a constant physical spacing. This grid length will often
be used in a digital image grid, where the base units are likely to be pixel spacings. Uses the MeasureType
with the restriction that the unit of measure referenced by uom must be suitable for length along the axes of
a grid, such as pixel spacings or grid spacings. </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleContent>
     <xsd:extension base="gml:MeasureType"/>
   </xsd:simpleContent>
 </xsd:complexType>
 <!-- ============================================================== -->
 <xsd:complexType name="AreaType">
   <xsd:annotation>
     <xsd:documentation>Value of a spatial area quantity, with its units. Uses the MeasureType with the
restriction that the unit of measure referenced by uom must be suitable for an area, such as square metres
or square miles. </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleContent>
     <xsd:extension base="gml:MeasureType"/>
   </xsd:simpleContent>
 </xsd:complexType>
 <!-- ============================================================== -->
 <xsd:complexType name="VolumeType">
   <xsd:annotation>
     <xsd:documentation>Value of a spatial volume quantity, with its units. Uses the MeasureType with the
restriction that the unit of measure referenced by uom must be suitable for a volume, such as cubic metres
or cubic feet. </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleContent>
     <xsd:extension base="gml:MeasureType"/>
   </xsd:simpleContent>
 </xsd:complexType>
 <!-- ============================================================== -->
 <xsd:complexType name="VelocityType">
   <xsd:annotation>
     <xsd:documentation>Value of a velocity, with its units. Uses the MeasureType with the restriction that
the unit of measure referenced by uom must be suitable for a velocity, such as metres per second or miles
per hour. </xsd:documentation>
   </xsd:annotation>
   <xsd:simpleContent>
     <xsd:extension base="gml:MeasureType"/>
   </xsd:simpleContent>
 </xsd:complexType>
 <!-- ============================================================== -->
 <xsd:complexType name="AngleChoiceType">
   <xsd:annotation>
     <xsd:documentation>Value of an angle quantity provided in either degree-minute-second format or
single value format. </xsd:documentation>
   </xsd:annotation>
   <xsd:choice>
     <xsd:element ref="gml:angle"/>
     <xsd:element ref="gml:dmsAngle"/>
   </xsd:choice>
```

```
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="angle" type="gml:AngleType"/>
  <!-- ============================================================== -->
  <xsd:complexType name="AngleType">
    <xsd:annotation>
      <xsd:documentation>Value of an angle quantity recorded as a single number, with its units. Uses the
MeasureType with the restriction that the unit of measure referenced by uom must be suitable for an angle,
such as degrees or radians. </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="gml:MeasureType"/>
    </xsd:simpleContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="dmsAngle" type="gml:DMSAngleType"/>
  <!-- ============================================================== -->
  <xsd:complexType name="DMSAngleType">
    <xsd:annotation>
      <xsd:documentation>Angle value provided in degree-minute-second or degree-minute format.
</xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element ref="gml:degrees"/>
      <xsd:choice minOccurs="0">
        <xsd:element ref="gml:decimalMinutes"/>
        <xsd:sequence>
          <xsd:element ref="gml:minutes"/>
          <xsd:element ref="gml:seconds" minOccurs="0"/>
        </xsd:sequence>
      </xsd:choice>
    </xsd:sequence>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="degrees" type="gml:DegreesType"/>
  <!-- ============================================================== -->
  <xsd:complexType name="DegreesType">
    <xsd:annotation>
      <xsd:documentation>Integer number of degrees, plus the angle direction. This element can be used for
geographic Latitude and Longitude. For Latitude, the XML attribute direction can take the values "N" or "S",
meaning North or South of the equator. For Longitude, direction can take the values "E" or "W", meaning
East or West of the prime meridian. This element can also be used for other angles. In that case, the
direction can take the values "+" or "-" (of SignType), in the specified rotational direction from a specified
reference direction. </xsd:documentation>
    </xsd:annotation>
    <xsd:simpleContent>
      <xsd:extension base="gml:DegreeValueType">
        <xsd:attribute name="direction">
          <xsd:simpleType>
            <xsd:union>
              <xsd:simpleType>
                <xsd:restriction base="string">
                  <xsd:enumeration value="N"/>
                  <xsd:enumeration value="E"/>
                  <xsd:enumeration value="S"/>
                  <xsd:enumeration value="W"/>
                </xsd:restriction>
              </xsd:simpleType>
              <xsd:simpleType>
                <xsd:restriction base="gml:SignType"/>
```

```
          </xsd:simpleType>
        </xsd:union>
      </xsd:simpleType>
    </xsd:attribute>
   </xsd:extension>
  </xsd:simpleContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:simpleType name="DegreeValueType">
  <xsd:annotation>
   <xsd:documentation>Integer number of degrees in a degree-minute-second or degree-minute angular
value, without indication of direction. </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="nonNegativeInteger">
   <xsd:maxInclusive value="359"/>
  </xsd:restriction>
 </xsd:simpleType>
 <!-- ============================================================ -->
 <xsd:element name="decimalMinutes" type="gml:DecimalMinutesType"/>
 <!-- ============================================================ -->
 <xsd:simpleType name="DecimalMinutesType">
  <xsd:annotation>
   <xsd:documentation>Decimal number of arc-minutes in a degree-minute angular value.
</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="decimal">
   <xsd:minInclusive value="0.00"/>
   <xsd:maxExclusive value="60.00"/>
  </xsd:restriction>
 </xsd:simpleType>
 <!-- ============================================================ -->
 <xsd:element name="minutes" type="gml:ArcMinutesType"/>
 <!-- ============================================================ -->
 <xsd:simpleType name="ArcMinutesType">
  <xsd:annotation>
   <xsd:documentation>Integer number of arc-minutes in a degree-minute-second angular value.
</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="nonNegativeInteger">
   <xsd:maxInclusive value="59"/>
  </xsd:restriction>
 </xsd:simpleType>
 <!-- ============================================================ -->
 <xsd:element name="seconds" type="gml:ArcSecondsType"/>
 <!-- ============================================================ -->
 <xsd:simpleType name="ArcSecondsType">
  <xsd:annotation>
   <xsd:documentation>Number of arc-seconds in a degree-minute-second angular value.
</xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="decimal">
   <xsd:minInclusive value="0.00"/>
   <xsd:maxExclusive value="60.00"/>
  </xsd:restriction>
 </xsd:simpleType>
 <!-- ============================================================ -->
</xsd:schema>
```


temporal.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:temporal:v3.00"/>
    <documentation xml:lang="en">
    The temporal schema for GML 3.0 provides constructs for handling time-varying spatial
    data. This schema reflects a partial yet consistent implementation of the model described
    in ISO 19108 (topological complexes and feature succession are not included).
    Copyright (c) 2002 OGC, All Rights Reserved.
    </documentation>
  </annotation>
  <!-- ================================================================== -->
  <include schemaLocation="units.xsd"/>
  <!-- ================================================================== -->
  <element name="_TimeObject" type="gml:AbstractTimeType" abstract="true"
substitutionGroup="gml:_GML">
    <annotation>
      <documentation xml:lang="en">
      This abstract element acts as the head of the substitution group for temporal primitives and complexes.
      </documentation>
    </annotation>
  </element>
  <!-- ==================================== -->
  <complexType name="AbstractTimeType" abstract="true">
    <annotation>
      <documentation xml:lang="en">
       A temporal object must be associated with a temporal reference system via URI.
       The Gregorian calendar with UTC is the default reference system, following ISO
       8601. Other reference systems in common use include the GPS calendar and the
       Julian calendar.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGMLType">
        <attribute name="frame" type="anyURI" use="optional" default="#ISO-8601"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- ================================================================== -->
  <element name="_TimePrimitive" type="gml:TimePrimitiveType" abstract="true"
substitutionGroup="gml:_TimeObject">
    <annotation>
      <documentation xml:lang="en">
      This abstract element acts as the head of the substitution group for temporal primitives.
      </documentation>
    </annotation>
  </element>
  <!-- ==================================== -->
  <complexType name="TimePrimitiveType" abstract="true">
    <annotation>
      <documentation xml:lang="en">
      The abstract supertype for temporal and topological primitives.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractTimeType"/>
    </complexContent>
```

```xml
</complexType>
<!-- ====================================================================== -->
<element name="TimeInstant" type="gml:TimeInstantType" substitutionGroup="gml:_TimePrimitive"/>
<!-- ====================================== -->
<complexType name="TimeInstantType" final="#all">
  <complexContent>
    <extension base="gml:TimePrimitiveType">
      <sequence>
        <element ref="gml:timePosition"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ====================================================================== -->
<element name="begin" type="gml:TimeInstantPropertyType"/>
<element name="end" type="gml:TimeInstantPropertyType"/>
<!-- ====================================== -->
<complexType name="TimeInstantPropertyType">
  <annotation>
    <documentation xml:lang="en">
     An association with a gml:TimeInstant element at the end.
    </documentation>
  </annotation>
  <sequence>
    <element ref="gml:TimeInstant" minOccurs="0"/>
  </sequence>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!-- ====================================================================== -->
<element name="TimePeriod" type="gml:TimePeriodType" substitutionGroup="gml:_TimePrimitive"/>
<!-- ====================================== -->
<complexType name="TimePeriodType">
  <complexContent>
    <extension base="gml:TimePrimitiveType">
      <sequence>
        <element ref="gml:begin"/>
        <element ref="gml:end"/>
        <element ref="gml:_duration" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ====================================================================== -->
<element name="_duration" type="gml:TimeDurationType" abstract="true">
  <annotation>
    <documentation xml:lang="en">
    This abstract element serves as the head of the substitution group for elements used
    to indicate temporal length or distance (duration, interval).
   </documentation>
  </annotation>
</element>
<!-- ====================================== -->
<simpleType name="TimeDurationType">
  <annotation>
    <documentation xml:lang="en">
    Base type for describing temporal length or distance. The value space is further
    constrained by subtypes that conform to the ISO 8601 or ISO 11404 standards.
    </documentation>
  </annotation>
  <union memberTypes="duration decimal"/>
```

```
</simpleType>
<!-- ====================================================================== -->
<element name="duration" type="duration" substitutionGroup="gml:_duration">
  <annotation>
    <documentation xml:lang="en">
    This element is an instance of the primitive xsd:duration simple type to
    enable use of the ISO 8601 syntax for temporal length (e.g. P5DT4H30M).
    It is a valid subtype of TimeDurationType according to section 3.14.6,
    rule 2.2.4 in XML Schema, Part 1.
    </documentation>
  </annotation>
</element>
<!-- ====================================================================== -->
<simpleType name="TemporalPositionType">
  <annotation>
    <documentation xml:lang="en">
    Here we have collapsed the hierarchy of subtypes for temporal position in 19108
    by defining a union of simple types for indicating temporal position relative to a
    specific reference system.
    Date and time may be indicated with varying degrees of precision:
    year, year-month, date, or dateTime (all ISO 8601 format). Note
    that the dateTime type does not allow right-truncation (i.e. omitting
    seconds). An ordinal era may be referenced via URI, and a decimal value
    can be used to indicate the distance from the scale origin (e.g. UNIX time,
    GPS calendar).
    </documentation>
  </annotation>
  <union memberTypes="dateTime date gYearMonth gYear anyURI decimal"/>
</simpleType>
<!-- ====================================================================== -->
<element name="timePosition" type="gml:TimePositionType">
  <annotation>
    <documentation>Direct representation of a temporal position. </documentation>
  </annotation>
</element>
<!-- ==================================== -->
<complexType name="TimePositionType" final="#all">
  <annotation>
    <documentation xml:lang="en">
    Indeterminate time values are also allowed, as described in ISO 19108. The indeterminatePosition
    attribute can be used alone or it can qualify a specific value for temporal position (e.g. before
    2002-12, after 1019624400). For time values that identify position within a calendar, the
    calendarEraName attribute provides the name of the calendar era to which the date is
    referenced (e.g. the Meiji era of the Japanese calendar).
    </documentation>
  </annotation>
  <simpleContent>
    <extension base="gml:TemporalPositionType">
      <attribute name="indeterminatePosition" type="gml:TimeIndeterminateValueType" use="optional"/>
      <attribute name="calendarEraName" type="string" use="optional"/>
      <attribute name="frame" type="anyURI" use="optional" default="#ISO-8601"/>
    </extension>
  </simpleContent>
</complexType>
<!-- ====================================================================== -->
<simpleType name="TimeIndeterminateValueType">
  <annotation>
    <documentation xml:lang="en">
      This enumerated data type specifies values for indeterminate positions.
    </documentation>
```

```
    </annotation>
    <restriction base="string">
      <enumeration value="after"/>
      <enumeration value="before"/>
      <enumeration value="now"/>
      <enumeration value="unknown"/>
    </restriction>
  </simpleType>
  <!-- ====================================================================== -->
  <element name="interval" type="gml:TimeIntervalLengthType" substitutionGroup="gml:_duration">
    <annotation>
      <documentation>
            This element is a valid subtype of TimeDurationType according to section 3.14.6,
            rule 2.2.4 in XML Schema, Part 1.
            </documentation>
    </annotation>
  </element>
  <!-- ====================================== -->
  <complexType name="TimeIntervalLengthType" final="#all">
    <annotation>
      <documentation xml:lang="en">
      This type extends the built-in xsd:decimal simple type to allow floating-point
      values for temporal length. According to  the ISO 11404 model you have to use
      positiveInteger together with appropriate values for radix and factor. The
      resolution of the time interval is to one radix ^(-factor) of the specified
      time unit (e.g. unit="second", radix="10", factor="3" specifices a resolution
      of milliseconds). It is a subtype of TimeDurationType.
      </documentation>
    </annotation>
    <simpleContent>
      <extension base="decimal">
        <attribute name="unit" type="gml:TimeUnitType" use="required"/>
        <attribute name="radix" type="positiveInteger" use="optional"/>
        <attribute name="factor" type="integer" use="optional"/>
      </extension>
    </simpleContent>
  </complexType>
  <!-- ====================================================================== -->
  <simpleType name="TimeUnitType">
    <annotation>
      <documentation xml:lang="en">
        This enumerated data type indicates standard units for measuring time.
      </documentation>
    </annotation>
    <restriction base="string">
      <enumeration value="year"/>
      <enumeration value="month"/>
      <enumeration value="day"/>
      <enumeration value="hour"/>
      <enumeration value="minute"/>
      <enumeration value="second"/>
    </restriction>
  </simpleType>
  <!-- ====================================================================== -->
  <element name="timePrimitiveProperty" type="gml:TimePrimitivePropertyType"/>
  <element name="timeStamp" type="gml:TimePrimitivePropertyType"
substitutionGroup="gml:timePrimitiveProperty"/>
  <!-- ====================================== -->
  <complexType name="TimePrimitivePropertyType">
    <annotation>
```



```
      <documentation xml:lang="en">
        A timestamp property associates an element with a temporal primitive.
      </documentation>
    </annotation>
    <sequence>
      <element ref="gml:_TimePrimitive" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- =============== types for defining temporal reference systems ============ -->
  <element name="TimeReferenceSystem" type="gml:TimeReferenceSystemType"
substitutionGroup="gml:_GML"/>
  <!-- =================================== -->
  <complexType name="TimeReferenceSystemType">
    <annotation>
      <documentation xml:lang="en">
        A value in the time domain is measured relative to a temporal reference system. Common
        types of reference systems include calendars, ordinal temporal reference systems, and
        temporal coordinate systems (time elapsed since some epoch, e.g. UNIX time).
      </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:DefinitionType">
        <sequence>
          <element name="domainOfValidity" type="string"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================== -->
  <element name="TimeCoordinateSystem" type="gml:TimeCoordinateSystemType"
substitutionGroup="gml:TimeReferenceSystem"/>
  <!-- =================================== -->
  <complexType name="TimeCoordinateSystemType" final="#all">
    <annotation>
      <documentation xml:lang="en">
      A temporal coordinate system is based on a continuous interval scale defined in terms of a single time
interval.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:TimeReferenceSystemType">
        <sequence>
          <element name="origin" type="gml:TimeInstantPropertyType"/>
          <element name="interval" type="gml:UnitOfMeasureType"/>
          <element name="incrementDirection" type="gml:SignType" default="+" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================== -->
  <element name="TimeOrdinalReferenceSystem" type="gml:TimeOrdinalReferenceSystemType"
substitutionGroup="gml:TimeReferenceSystem"/>
  <!-- =================================== -->
  <complexType name="TimeOrdinalReferenceSystemType" final="#all">
    <annotation>
      <documentation xml:lang="en">
      In an ordinal reference system the order of events in time can be well
      established, but the magnitude of the intervals between them can not be
      accurately determined (e.g. a stratigraphic sequence).
```

```
      </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:TimeReferenceSystemType">
        <sequence>
          <element name="component" type="gml:TimeOrdinalReferenceSystemMemberType"
maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- =================================================================== -->
  <complexType name="TimeOrdinalReferenceSystemMemberType">
    <sequence>
      <element ref="gml:TimeOrdinalEra"/>
    </sequence>
  </complexType>
  <!-- =================================================================== -->
  <element name="TimeOrdinalEra" type="gml:TimeOrdinalEraType" substitutionGroup="gml:_GML"/>
  <!-- ====================================== -->
  <complexType name="TimeOrdinalEraType">
    <annotation>
      <documentation xml:lang="en">
      Ordinal temporal reference systems are often hierarchically structured
      such that an ordinal era at a given level of the hierarchy includes a
      sequence of shorter, coterminous ordinal eras.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGMLType">
        <sequence>
          <element ref="gml:begin" minOccurs="0"/>
          <element ref="gml:end" minOccurs="0"/>
          <element name="member" type="gml:TimeOrdinalReferenceSystemMemberType" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- =================================================================== -->
</schema>
```

geometryBasic0d1d.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-
xsd:geometryBasic0d1d:v3.00">geometryBasic0d1d.xsd</appinfo>
    <documentation>
		Copyright (c) 2001-2002 OGC, All Rights Reserved.
    </documentation>
  </annotation>
  <!-- ============================================================== -->
  <include schemaLocation="measures.xsd">
    <annotation>
```

```
      <documentation>This includes not only measures.xsd, but also units.xsd, gmlBase.xsd and basicTypes.xsd.</documentation>
    </annotation>
  </include>
  <!-- ============================================================== -->
  <!-- ===========  abstract supertype for geometry objects =================== -->
  <!-- ============================================================== -->
  <element name="_Geometry" type="gml:AbstractGeometryType" abstract="true" substitutionGroup="gml:_GML">
    <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="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="gid" type="string" use="optional">
          <annotation>
            <documentation>This attribute is included for backward compatibility with GML 2 and is deprecated with GML 3. This identifer is superceded by "gml:id" inherited from AbstractGMLType. The attribute "gid" should not be used anymore and may be deleted in future versions of GML without further notice.</documentation>
          </annotation>
        </attribute>
        <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 (Note: These "WKCRS"-ids still have to be specified).  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 aggregate.</documentation>
          </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================== -->
  <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>
      <element ref="gml:_Geometry" minOccurs="0"/>
    </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="GeometryArrayPropertyType">
    <annotation>
      <documentation>A container for an array of geometry elements. The elements are always contained in
the array property, referencing geometry elements or arrays of geometry elements is not
supported.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:_Geometry" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </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>
  <!-- ============================================================== -->
  <complexType name="GeometricPrimitivePropertyType">
    <annotation>
      <documentation>A property that has a geometric primitive 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>
      <element ref="gml:_GeometricPrimitive" minOccurs="0"/>
    </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 (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>
          <choice>
            <annotation>
              <documentation>GML supports two different ways to specify the direct poisiton of a point. 1. The "pos" element is of type DirectPositionType. 2. The "coordinates" element is of type CoordinatesType. The number of direct positions in the coordinates list must be one.</documentation>
            </annotation>
            <element ref="gml:pos"/>
            <element ref="gml:coordinates"/>
            <element ref="gml:coord">
              <annotation>
                <documentation>Deprecated with GML version 3.0. Use "pos" instead. The "coord" element is included for backwards compatibility with GML 2.</documentation>
              </annotation>
            </element>
          </choice>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="pointProperty" type="gml:PointPropertyType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:pointProperty">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
      <documentation>This property element either references a point via the XLink-attributes or contains the point element. pointProperty is the predefined property which can be used by GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for Point.</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <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>
      <element ref="gml:Point" minOccurs="0"/>
    </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="pointArrayProperty" type="gml:PointArrayPropertyType"/>
  <!-- =========================================================== -->
  <complexType name="PointArrayPropertyType">
    <annotation>
      <documentation>A container for an array of points. The elements are always contained in the array
property, referencing geometry elements or arrays of geometry elements is not supported.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:Point" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <!-- =========================================================== -->
  <!-- primitive geometry objects (1-dimensional) -->
  <!-- ============================================================= -->
  <element name="_Curve" type="gml:AbstractCurveType" abstract="true"
substitutionGroup="gml:_GeometricPrimitive">
    <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>
  <!-- ============================================================= -->
  <element name="curveProperty" type="gml:CurvePropertyType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:curveProperty">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
```

```
      <documentation>This property element either references a curve via the XLink-attributes or contains the
curve element. curveProperty is the predefined property which can be used by GML Application Schemas
whenever a GML Feature has a property with a value that is substitutable for _Curve.</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <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>
      <element ref="gml:_Curve" minOccurs="0"/>
    </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="curveArrayProperty" type="gml:CurveArrayPropertyType"/>
  <!-- ========================================================= -->
  <complexType name="CurveArrayPropertyType">
    <annotation>
      <documentation>A container for an array of curves. The elements are always contained in the array
property, referencing geometry elements or arrays of geometry elements is not supported.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:_Curve" minOccurs="0" maxOccurs="unbounded"/>
    </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, GM_LineString of ISO 19107 is implemented by
LineStringSegment.              </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCurveType">
        <sequence>
          <choice>
            <annotation>
              <documentation>GML supports two different ways to specify the control points of a line string. 1. A
sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are
control points that are only part of this curve, "pointRep" elements contain a point that may be referenced
from other geometry elements or reference another point defined outside of this curve (reuse of existing
```

```
points). 2. The "coordinates" element allows for 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 curve only. The
number of direct positions in the coordinates list must be at least two.</documentation>
          </annotation>
          <choice minOccurs="2" maxOccurs="unbounded">
            <element ref="gml:pos"/>
            <element ref="gml:pointRep"/>
            <element ref="gml:coord">
              <annotation>
                <documentation>Deprecated with GML version 3.0. Use "pos" instead. The "coord" element is
included for backwards compatibility with GML 2.</documentation>
              </annotation>
            </element>
          </choice>
          <element ref="gml:coordinates"/>
        </choice>
      </sequence>
    </extension>
  </complexContent>
 </complexType>
 <!-- ============================================================ -->
 <!-- positions -->
 <!-- ========================================================= -->
 <element name="pos" type="gml:DirectPositionType"/>
 <!-- ============================================================ -->
 <complexType name="DirectPositionType">
   <annotation>
     <documentation>DirectPosition instances hold the coordinates for a position within some coordinate
reference system (CRS). Since DirectPositions, as data types, will often be included in larger objects (such
as geometry elements) that have references to CRS, the "srsName" attribute will in general be missing, if
this particular DirectPosition is included in a larger element with such a reference to a CRS. In this case, the
CRS is implicitly assumed to take on the value of the containing object's CRS.</documentation>
   </annotation>
   <simpleContent>
     <extension base="gml:doubleList">
       <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 (Note: These "WKCRS"-ids
still have to be specified).  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
the attribute will be specified at the direct position level only in rare cases.</documentation>
         </annotation>
       </attribute>
       <attribute name="dimension" type="positiveInteger" use="optional">
         <annotation>
           <documentation>The attribute "dimension" is the length of coordinate sequence (the number of
entries in the list). This is determined by the coordinate reference system.</documentation>
         </annotation>
       </attribute>
     </extension>
   </simpleContent>
 </complexType>
 <!-- ============================================================ -->
 <element name="vector" type="gml:VectorType"/>
 <!-- ============================================================ -->
 <complexType name="VectorType">
   <annotation>
```

```
      <documentation>A Vector is an ordered set of numbers called coordinates that represent a position in a
coordinate reference system (CRS). For some application the components of the position may be adjusted
to yield a unit vector.</documentation>
    </annotation>
    <simpleContent>
      <restriction base="gml:DirectPositionType"/>
    </simpleContent>
  </complexType>
  <!-- =========================================================== -->
  <element name="pointRep" type="gml:PointPropertyType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:pointRep">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
    </annotation>
  </element>
  <!-- =========================================================== -->
  <element name="coordinates" type="gml:CoordinatesType"/>
  <!-- ========================================================= -->
  <!-- Envelope -->
  <!-- ========================================================= -->
  <element name="Envelope" type="gml:EnvelopeType" substitutionGroup="gml:_Geometry"/>
  <!-- ========================================================= -->
  <complexType name="EnvelopeType">
    <annotation>
      <documentation>Envelope defines an extent using a pair of positions defining opposite corners in
arbitrary dimensions.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGeometryType">
        <sequence>
          <choice>
            <element ref="gml:coord" minOccurs="2" maxOccurs="2">
              <annotation>
                <appinfo>deprecated</appinfo>
                <documentation>deprecated with GML version 3.0</documentation>
              </annotation>
            </element>
            <element ref="gml:pos" minOccurs="2" maxOccurs="2"/>
            <element ref="gml:coordinates"/>
          </choice>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ========================================================= -->
  <!-- ========================================================= -->
  <!-- ========================================================= -->
  <!--         The following types and elements are deprecated and should not be used !                   -->
  <element name="coord" type="gml:CoordType">
    <annotation>
      <documentation>Deprecated with GML 3.0 and included for backwards compatibility with GML 2. Use
the "pos" element instead.</documentation>
    </annotation>
  </element>
  <complexType name="CoordType">
```



```
    <annotation>
      <documentation>Represents a coordinate tuple in one, two, or three dimensions. Deprecated with GML 3.0 and replaced by DirectPositionType.</documentation>
    </annotation>
    <sequence>
      <element name="X" type="decimal"/>
      <element name="Y" type="decimal" minOccurs="0"/>
      <element name="Z" type="decimal" minOccurs="0"/>
    </sequence>
  </complexType>
  <!-- ============================================================== -->
  <element name="lineStringProperty" type="gml:LineStringPropertyType">
    <annotation>
      <documentation>Deprecated with GML 3.0 and included only for backwards compatibility with GML 2.0. Use "curveProperty" instead. This property element either references a line string via the XLink-attributes or contains the line string element.</documentation>
    </annotation>
  </element>
  <!-- ============================================================== -->
  <complexType name="LineStringPropertyType">
    <annotation>
      <documentation>This type is deprecated with GML 3 and shall not be used. It is included for backwards compatibility with GML 2. Use CurvePropertyType instead. A property that has a line string 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>
      <element ref="gml:LineString" minOccurs="0"/>
    </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>
  <!-- ============================================================== -->
</schema>
```

geometryBasic2d.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml" xmlns:sch="http://www.ascc.net/xml/schematron" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:geometryBasic2d:v3.00">geometryBasic2d.xsd</appinfo>
    <documentation>
		Copyright (c) 2001-2002 OGC, All Rights Reserved.
```

```
    </documentation>
  </annotation>
  <include schemaLocation="geometryBasic0d1d.xsd"/>
  <!-- ============================================================= -->
  <!-- primitive geometry objects (2-dimensional) -->
  <!-- ============================================================= -->
  <element name="_Surface" type="gml:AbstractSurfaceType" abstract="true"
substitutionGroup="gml:_GeometricPrimitive">
    <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>
  <!-- ============================================================= -->
  <element name="surfaceProperty" type="gml:SurfacePropertyType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:surfaceProperty">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
      <documentation>This property element either references a surface via the XLink-attributes or contains
the surface element. surfaceProperty is the predefined property which can be used by GML Application
Schemas whenever a GML Feature has a property with a value that is substitutable for
_Surface.</documentation>
    </annotation>
  </element>
  <!-- ============================================================= -->
  <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>
      <element ref="gml:_Surface" minOccurs="0"/>
    </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="surfaceArrayProperty" type="gml:SurfaceArrayPropertyType"/>
  <!-- =========================================================== -->
  <complexType name="SurfaceArrayPropertyType">
    <annotation>
      <documentation>A container for an array of surfaces. The elements are always contained in the array
property, referencing geometry elements or arrays of geometry elements is not supported.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:_Surface" minOccurs="0" maxOccurs="unbounded"/>
    </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, GM_Polygon of ISO 19107 is implemented by PolygonPatch.</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="_Ring" type="gml:AbstractRingType" abstract="true"
substitutionGroup="gml:_Geometry">
    <annotation>
      <documentation>The "_Ring" element is the abstract head of the substituition group for all closed
boundaries of a surface patch.</documentation>
    </annotation>
  </element>
  <!-- =========================================================== -->
  <complexType name="AbstractRingType" abstract="true">
    <annotation>
      <documentation>
            An abstraction of a ring to support surface boundaries of different complexity.
         </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGeometryType"/>
    </complexContent>
  </complexType>
  <!-- =========================================================== -->
  <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>
  <element name="outerBoundaryIs" type="gml:AbstractRingPropertyType" substitutionGroup="gml:exterior">
   <annotation>
    <documentation>
                 Deprecated with GML 3.0, included only for backwards compatibility with GML 2. Use "exterior" instead.
              </documentation>
   </annotation>
  </element>
  <element name="innerBoundaryIs" type="gml:AbstractRingPropertyType" substitutionGroup="gml:interior">
   <annotation>
    <documentation>
                 Deprecated with GML 3.0, included only for backwards compatibility with GML 2. Use "interior" instead.
              </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:_Ring"/>
   </sequence>
  </complexType>
  <!-- ============================================================== -->
  <element name="LinearRing" type="gml:LinearRingType" substitutionGroup="gml:_Ring"/>
  <!-- ============================================================== -->
  <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:AbstractRingType">
     <sequence>
      <choice>
       <annotation>
        <documentation>GML supports two different ways to specify the control points of a linear ring.
1. A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements are control points that are only part of this ring, "pointRep" elements contain a point that may be referenced from other geometry elements or reference another point defined outside of this ring (reuse of existing points).
```

```
2. The "coordinates" element allows for 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 coordinate list must be at least four.</documentation>
      </annotation>
      <choice minOccurs="4" maxOccurs="unbounded">
        <element ref="gml:pos"/>
        <element ref="gml:pointRep"/>
      </choice>
      <element ref="gml:coordinates"/>
      <element ref="gml:coord" minOccurs="4" maxOccurs="unbounded">
        <annotation>
          <documentation>Deprecated with GML version 3.0 and included for backwards compatibility with GML 2. Use "pos" elements instead.</documentation>
        </annotation>
      </element>
     </choice>
    </sequence>
   </extension>
  </complexContent>
 </complexType>
 <!-- ============================================================== -->
 <complexType name="LinearRingPropertyType">
  <annotation>
   <documentation>
        Encapsulates a ring to represent properties in features or geometry collections.
      </documentation>
  </annotation>
  <choice>
   <element ref="gml:LinearRing"/>
  </choice>
 </complexType>
 <!-- ============================================================== -->
 <!--

   The following types and elements are deprecated and should not be used !

   -->
 <!-- ============================================================== -->
 <element name="polygonProperty" type="gml:PolygonPropertyType">
  <annotation>
   <documentation>Deprecated with GML 3.0 and included only for backwards compatibility with GML 2.0. Use "surfaceProperty" instead.
This property element either references a polygon via the XLink-attributes or contains the polygon element.</documentation>
  </annotation>
 </element>
 <!-- ============================================================== -->
 <complexType name="PolygonPropertyType">
  <annotation>
   <documentation>This type is deprecated with GML 3 and shall not be used. It is included for backwards compatibility with GML 2. Use SurfacePropertyType instead.
A property that has a polygon 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>
   <element ref="gml:Polygon" minOccurs="0"/>
  </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>
  <!-- ============================================================== -->
</schema>
```

geometryPrimitives.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-
xsd:geometryPrimitives:v3.00">geometryPrimitives.xsd</appinfo>
    <documentation>
		Copyright (c) 2001-2002 OGC, All Rights Reserved.
    </documentation>
  </annotation>
  <!-- ============================================================== -->
  <include schemaLocation="geometryBasic2d.xsd"/>
  <!-- ============================================================== -->
  <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. Each curve segment within a curve
may be defined using a different interpolation method. 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>
  <!-- ============================================================== -->
```

```
  <element name="baseCurve" type="gml:CurvePropertyType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:baseCurve">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
      <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>
  <!-- ============================================================== -->
  <element name="OrientableCurve" type="gml:OrientableCurveType" substitutionGroup="gml:_Curve"/>
  <!-- ============================================================== -->
  <complexType name="OrientableCurveType">
    <annotation>
      <documentation>
                  OrientableCurve consists of a curve and an orientation. If the orientation is "+", then the
OrientableCurve is identical to the baseCurve. If the orientation is "-", then the OrientableCurve is related to
another _Curve with a parameterization that reverses the sense of the curve traversal.
               </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCurveType">
        <sequence>
          <element ref="gml:baseCurve">
            <annotation>
              <documentation>References or contains the base curve (positive orientation).
NOTE: This definition allows for a nested structure, i.e. an OrientableCurve may use another
OrientableCurve as its base curve.</documentation>
            </annotation>
          </element>
        </sequence>
        <attribute name="orientation" type="gml:SignType" default="+">
          <annotation>
            <documentation>If the orientation is "+", then the OrientableCurve is identical to the baseCurve. If
the orientation is "-", then the OrientableCurve is related to another _Curve with a parameterization that
reverses the sense of the curve traversal. "+" is the default value.</documentation>
          </annotation>
        </attribute>
      </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>
    <attribute name="numDerivativesAtStart" type="integer" use="optional" default="0">
      <annotation>
        <documentation>The attribute "numDerivativesAtStart" specifies the type of continuity between this
curve segment and its predecessor. If this is the first curve segment in the curve, one of these values, as
appropriate, is ignored. The default value of "0" means simple continuity, which is a mandatory minimum
level of continuity. This level is referred to as "C 0 " in mathematical texts. A value of 1 means that the
function and its first derivative are continuous at the appropriate end point: "C 1 " continuity. A value of "n"
for any integer means the function and its first n derivatives are continuous: "C n " continuity.
NOTE: Use of these values is only appropriate when the basic curve definition is an underdetermined
system. For example, line string segments cannot support continuity above C 0 , since there is no spare
control parameter to adjust the incoming angle at the end points of the segment. Spline functions on the
other hand often have extra degrees of freedom on end segments that allow them to adjust the values of the
derivatives to support C 1 or higher continuity.</documentation>
      </annotation>
    </attribute>
    <attribute name="numDerivativesAtEnd" type="integer" use="optional" default="0">
      <annotation>
        <documentation>The attribute "numDerivativesAtEnd" specifies the type of continuity between this
curve segment and its successor. If this is the last curve segment in the curve, one of these values, as
appropriate, is ignored. The default value of "0" means simple continuity, which is a mandatory minimum
level of continuity. This level is referred to as "C 0 " in mathematical texts. A value of 1 means that the
function and its first derivative are continuous at the appropriate end point: "C 1 " continuity. A value of "n"
for any integer means the function and its first n derivatives are continuous: "C n " continuity.
NOTE: Use of these values is only appropriate when the basic curve definition is an underdetermined
system. For example, line string segments cannot support continuity above C 0 , since there is no spare
control parameter to adjust the incoming angle at the end points of the segment. Spline functions on the
other hand often have extra degrees of freedom on end segments that allow them to adjust the values of the
derivatives to support C 1 or higher continuity.</documentation>
      </annotation>
    </attribute>
    <attribute name="numDerivativeInterior" type="integer" use="optional" default="0">
      <annotation>
        <documentation>The attribute "numDerivativesInterior" specifies the type of continuity that is
guaranteed interior to the curve. The default value of "0" means simple continuity, which is a mandatory
minimum level of continuity. This level is referred to as "C 0 " in mathematical texts. A value of 1 means that
the function and its first derivative are continuous at the appropriate end point: "C 1 " continuity. A value of
"n" for any integer means the function and its first n derivatives are continuous: "C n " continuity.
NOTE: Use of these values is only appropriate when the basic curve definition is an underdetermined
system. For example, line string segments cannot support continuity above C 0 , since there is no spare
control parameter to adjust the incoming angle at the end points of the segment. Spline functions on the
other hand often have extra degrees of freedom on end segments that allow them to adjust the values of the
derivatives to support C 1 or higher continuity.</documentation>
      </annotation>
    </attribute>
  </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>
```

```xml
      <element ref="gml:_CurveSegment" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <!-- =========== 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>
          <choice>
            <annotation>
              <documentation>GML supports two different ways to specify the control points of a curve segment.
1. A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements
are control points that are only part of this curve segment, "pointRep" elements contain a point that may be
referenced from other geometry elements or reference another point defined outside of this curve segment
(reuse of existing points).
2. The "coordinates" element allows for 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 curve segment only. The
number of direct positions in the coordinate list must be at least two.</documentation>
            </annotation>
            <choice minOccurs="2" maxOccurs="unbounded">
              <element ref="gml:pos"/>
              <element ref="gml:pointRep"/>
            </choice>
            <element ref="gml:coordinates"/>
          </choice>
        </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>
  <!-- =========== global element in "_CurveSegment" substitution group ================= -->
  <element name="ArcString" type="gml:ArcStringType" substitutionGroup="gml:_CurveSegment"/>
  <!-- ============================================================ -->
  <complexType name="ArcStringType">
    <annotation>
      <documentation>
                  An ArcString is a curve segment that uses three-point circular arc interpolation.
               </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCurveSegmentType">
        <sequence>
          <choice>
```



```
          <annotation>
            <documentation>GML supports two different ways to specify the control points of a curve segment.
1. A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements
are control points that are only part of this curve segment, "pointRep" elements contain a point that may be
referenced from other geometry elements or reference another point defined outside of this curve segment
(reuse of existing points).
2. The "coordinates" element allows for 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 curve segment only. The
number of direct positions in the coordinate list must be at least three.</documentation>
          </annotation>
          <choice minOccurs="3" maxOccurs="unbounded">
            <element ref="gml:pos"/>
            <element ref="gml:pointRep"/>
          </choice>
          <element ref="gml:coordinates"/>
        </choice>
      </sequence>
      <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="circularArc3Points">
        <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 an
ArcString the interpolation is fixed as "circularArc3Points".</documentation>
        </annotation>
      </attribute>
      <attribute name="numArc" type="integer" use="optional">
        <annotation>
          <documentation>The number of arcs in the arc string can be explicitly stated in this attribute. The
number of control points in the arc string must be 2 * numArc + 1.</documentation>
        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>
<!-- =========== global element in "_CurveSegment" substitution group ================ -->
<element name="Arc" type="gml:ArcType" substitutionGroup="gml:ArcString"/>
<!-- ========================================================== -->
<complexType name="ArcType">
  <annotation>
    <documentation>
            An Arc is an arc string with only one arc unit, i.e. three control points.
        </documentation>
  </annotation>
  <complexContent>
    <restriction base="gml:ArcStringType">
      <sequence>
        <choice>
          <annotation>
            <documentation>GML supports two different ways to specify the control points of a curve segment.
1. A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements
are control points that are only part of this curve segment, "pointRep" elements contain a point that may be
referenced from other geometry elements or reference another point defined outside of this curve segment
(reuse of existing points).
2. The "coordinates" element allows for 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 curve segment only. The
number of direct positions in the coordinate list must be three.</documentation>
          </annotation>
          <choice minOccurs="3" maxOccurs="3">
            <element ref="gml:pos"/>
            <element ref="gml:pointRep"/>
```

```
          </choice>
          <element ref="gml:coordinates"/>
        </choice>
      </sequence>
      <attribute name="numArc" type="integer" use="optional" fixed="1">
        <annotation>
          <documentation>An arc is an arc string consiting of a single arc, the attribute is fixed to
"1".</documentation>
        </annotation>
      </attribute>
    </restriction>
  </complexContent>
</complexType>
<!-- =========== global element in "_CurveSegment" substitution group ================= -->
<element name="Circle" type="gml:CircleType" substitutionGroup="gml:Arc"/>
<!-- ============================================================ -->
<complexType name="CircleType">
  <annotation>
    <documentation>
          A Circle is an arc whose first and last control points coincide to form a full circle. The "start"
and "end" bearing are equal and shall be the bearing for the first controlPoint listed. Note: This still requires
at least 3 distinct non-co-linear points to be unambiguously defined. The arc is simply extended until the first
point is encountered.
        </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:ArcType"/>
  </complexContent>
</complexType>
<!-- =========== global element in "_CurveSegment" substitution group ================= -->
<element name="ArcStringByBulge" type="gml:ArcStringByBulgeType"
substitutionGroup="gml:_CurveSegment"/>
<!-- ============================================================ -->
<complexType name="ArcStringByBulgeType">
  <annotation>
    <documentation>
          This variant of the arc computes the mid points of the arcs instead of storing the coordinates
directly. The control point sequence consists of the start and end points of each arc plus the bulge.
        </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractCurveSegmentType">
      <sequence>
        <choice>
          <annotation>
            <documentation>GML supports two different ways to specify the control points of a curve segment.
1. A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements
are control points that are only part of this curve segment, "pointRep" elements contain a point that may be
referenced from other geometry elements or reference another point defined outside of this curve segment
(reuse of existing points).
2. The "coordinates" element allows for 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 curve segment only. The
number of direct positions in the coordinate list must be at least two.</documentation>
          </annotation>
          <choice minOccurs="2" maxOccurs="unbounded">
            <element ref="gml:pos"/>
            <element ref="gml:pointRep"/>
          </choice>
          <element ref="gml:coordinates"/>
        </choice>
```

```
      <element name="bulge" type="double" maxOccurs="unbounded">
        <annotation>
          <documentation>The bulge controls the offset of each arc's midpoint. The "bulge" is the real
number multiplier for the normal that determines the offset direction of the midpoint of each arc. The length
of the bulge sequence is exactly 1 less than the length of the control point array, since a bulge is needed for
each pair of adjacent points in the control point array. The bulge is not given by a distance, since it is simply
a multiplier for the normal.
The midpoint of the resulting arc is given by: midPoint = ((startPoint + endPoint)/2.0) +
bulge*normal</documentation>
        </annotation>
      </element>
      <element name="normal" type="gml:VectorType" maxOccurs="unbounded">
        <annotation>
          <documentation>The attribute "normal" is a vector normal (perpendicular) to the chord of the arc,
the line joining the first and last
point of the arc. In a 2D coordinate system, there are only two possible directions for the normal, and it is
often given as a signed real, indicating its length, with a positive sign indicating a left turn angle from the
chord line, and a negative sign indicating a right turn from the chord. In 3D, the normal determines the plane
of the arc, along with the start and endPoint of the arc.
The normal is usually a unit vector, but this is not absolutely necessary. If the normal is a zero vector, the
geometric object becomes equivalent to the straight line between the two end points. The length of the
normal sequence is exactly the same as for the bulge sequence, 1 less than the control point sequence
length.</documentation>
        </annotation>
      </element>
    </sequence>
    <attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="circularArc2PointWithBulge">
      <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 an
ArcStringByBulge the interpolation is fixed as "circularArc2PointWithBulge".</documentation>
      </annotation>
    </attribute>
    <attribute name="numArc" type="integer" use="optional">
      <annotation>
        <documentation>The number of arcs in the arc string can be explicitly stated in this attribute. The
number of control points in the arc string must be numArc + 1.</documentation>
      </annotation>
    </attribute>
   </extension>
  </complexContent>
 </complexType>
 <!-- =========== global element in "_CurveSegment" substitution group ================ -->
 <element name="ArcByBulge" type="gml:ArcByBulgeType" substitutionGroup="gml:ArcStringByBulge"/>
 <!-- ============================================================ -->
 <complexType name="ArcByBulgeType">
  <annotation>
   <documentation>
          An ArcByBulge is an arc string with only one arc unit, i.e. two control points and one bulge.
       </documentation>
  </annotation>
  <complexContent>
   <restriction base="gml:ArcStringByBulgeType">
    <sequence>
     <choice>
      <annotation>
       <documentation>GML supports two different ways to specify the control points of a curve segment.
```

```
1. A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements
are control points that are only part of this curve segment, "pointRep" elements contain a point that may be
referenced from other geometry elements or reference another point defined outside of this curve segment
(reuse of existing points).
2. The "coordinates" element allows for 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 curve segment only. The
number of direct positions in the coordinate list must be two.</documentation>
          </annotation>
          <choice minOccurs="2" maxOccurs="2">
            <element ref="gml:pos"/>
            <element ref="gml:pointRep"/>
          </choice>
          <element ref="gml:coordinates"/>
        </choice>
        <element name="bulge" type="double">
          <annotation>
            <documentation>The bulge controls the offset of each arc's midpoint. The "bulge" is the real
number multiplier for the normal that determines the offset direction of the midpoint of each arc. The length
of the bulge sequence is exactly 1 less than the length of the control point array, since a bulge is needed for
each pair of adjacent points in the control point array. The bulge is not given by a distance, since it is simply
a multiplier for the normal.
The midpoint of the resulting arc is given by: midPoint = ((startPoint + endPoint)/2.0) +
bulge*normal</documentation>
          </annotation>
        </element>
        <element name="normal" type="gml:VectorType">
          <annotation>
            <documentation>The attribute "normal" is a vector normal (perpendicular) to the chord of the arc,
the line joining the first and last
point of the arc. In a 2D coordinate system, there are only two possible directions for the normal, and it is
often given as a signed real, indicating its length, with a positive sign indicating a left turn angle from the
chord line, and a negative sign indicating a right turn from the chord. In 3D, the normal determines the plane
of the arc, along with the start and endPoint of the arc.
The normal is usually a unit vector, but this is not absolutely necessary. If the normal is a zero vector, the
geometric object becomes equivalent to the straight line between the two end points. The length of the
normal sequence is exactly the same as for the bulge sequence, 1 less than the control point sequence
length.</documentation>
          </annotation>
        </element>
      </sequence>
      <attribute name="numArc" type="integer" use="optional" fixed="1">
        <annotation>
          <documentation>An arc is an arc string consiting of a single arc, the attribute is fixed to
"1".</documentation>
        </annotation>
      </attribute>
    </restriction>
  </complexContent>
</complexType>
<!-- =========== global element in "_CurveSegment" substitution group ================ -->
<element name="ArcByCenterPoint" type="gml:ArcByCenterPointType"
substitutionGroup="gml:_CurveSegment"/>
<!-- =========================================================== -->
<complexType name="ArcByCenterPointType">
  <annotation>
    <documentation>
                This variant of the arc requires that the points on the arc have to be computed instead of
storing the coordinates directly. The control point is the center point of the arc plus the radius and the
bearing at start and end. This represenation can be used only in 2D.
          </documentation>
```

```
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCurveSegmentType">
        <sequence>
          <choice>
            <annotation>
              <documentation>GML supports two different ways to specify the control points of a curve segment.
1. A "pos" (DirectPositionType) or "pointRep" (PointPropertyType) element. The "pos" element contains a
center point that is only part of this curve segment, a "pointRep" element contains a point that may be
referenced from other geometry elements or reference another point defined outside of this curve segment
(reuse of existing points).
2. The "coordinates" element can be used to specifiy the coordinates of the center point, too. The number of
direct positions in the coordinate list must be one.</documentation>
            </annotation>
            <choice>
              <element ref="gml:pos"/>
              <element ref="gml:pointRep"/>
            </choice>
            <element ref="gml:coordinates"/>
          </choice>
          <element name="radius" type="gml:LengthType">
            <annotation>
              <documentation>The radius of the arc.</documentation>
            </annotation>
          </element>
          <element name="startAngle" type="gml:AngleType" minOccurs="0">
            <annotation>
              <documentation>The bearing of the arc at the start.</documentation>
            </annotation>
          </element>
          <element name="endAngle" type="gml:AngleType" minOccurs="0">
            <annotation>
              <documentation>The bearing of the arc at the end.</documentation>
            </annotation>
          </element>
        </sequence>
        <attribute name="interpolation" type="gml:CurveInterpolationType"
fixed="circularArcCenterPointWithRadius">
          <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 an
ArcByCenterPoint the interpolation is fixed as "circularArcCenterPointWithRadius".</documentation>
          </annotation>
        </attribute>
        <attribute name="numArc" type="integer" use="required" fixed="1">
          <annotation>
            <documentation>Since this type describes always a single arc, the attribute is fixed to
"1".</documentation>
          </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
  <!-- =========== global element in "_CurveSegment" substitution group ================ -->
  <element name="CircleByCenterPoint" type="gml:CircleByCenterPointType"
substitutionGroup="gml:ArcByCenterPoint"/>
  <!-- ============================================================ -->
  <complexType name="CircleByCenterPointType">
    <annotation>
```

```
      <documentation>A CircleByCenterPoint is an ArcByCenterPoint with identical start and end angle to
form a full circle. Again, this represenation can be used only in 2D.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:ArcByCenterPointType"/>
    </complexContent>
  </complexType>
  <!-- =========== global element in "_CurveSegment" substitution group ================ -->
  <element name="CubicSpline" type="gml:CubicSplineType" substitutionGroup="gml:_CurveSegment"/>
  <!-- ============================================================ -->
  <complexType name="CubicSplineType">
    <annotation>
      <documentation>
			Cubic splines are similar to line strings in that they are a sequence of segments each with its
own defining function. A cubic spline uses the control points and a set of derivative parameters to define a
piecewise 3rd degree polynomial interpolation. Unlike line-strings, the parameterization by arc length is not
necessarily still a polynomial.
			The function describing the curve must be C2, that is, have a continuous 1st and 2nd
derivative at all points, and pass through the controlPoints in the order given. Between the control points, the
curve segment is defined by a cubic polynomial. At each control point, the polynomial changes in such a
manner that the 1st and 2nd derivative vectors are the same from either side. The control parameters record
must contain vectorAtStart, and vectorAtEnd which are the unit tangent vectors at controlPoint[1] and
controlPoint[n] where n = controlPoint.count.
			Note: only the direction of the vectors is relevant, not their length.
		</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCurveSegmentType">
        <sequence>
          <choice>
            <annotation>
              <documentation>GML supports two different ways to specify the control points of a curve segment.
1. A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements
are control points that are only part of this curve segment, "pointRep" elements contain a point that may be
referenced from other geometry elements or reference another point defined outside of this curve segment
(reuse of existing points).
2. The "coordinates" element allows for 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 curve segment only. The
number of direct positions in the coordinate list must be at least three.</documentation>
            </annotation>
            <choice minOccurs="3" maxOccurs="unbounded">
              <element ref="gml:pos"/>
              <element ref="gml:pointRep"/>
            </choice>
            <element ref="gml:coordinates"/>
          </choice>
          <element name="vectorAtStart" type="gml:VectorType">
            <annotation>
              <documentation>"vectorAtStart" is the unit tangent vector at the start point of the
spline.</documentation>
            </annotation>
          </element>
          <element name="vectorAtEnd" type="gml:VectorType">
            <annotation>
              <documentation>"vectorAtEnd" is the unit tangent vector at the end point of the
spline.</documentation>
            </annotation>
          </element>
        </sequence>
        <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="cubicSpline">
```

```
        <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
CubicSpline the interpolation is fixed as "cubicSpline".</documentation>
        </annotation>
      </attribute>
      <attribute name="degree" type="integer" fixed="3">
        <annotation>
          <documentation>The degree for a cubic spline is "3".</documentation>
        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>
<!-- ============================================================ -->
<complexType name="KnotType">
  <annotation>
    <documentation>A knot is a breakpoint on a piecewise spline curve.</documentation>
  </annotation>
  <sequence>
    <element name="value" type="double">
      <annotation>
        <documentation>The property "value" is the value of the parameter at the knot of the spline. The
sequence of knots shall be a non-decreasing sequence. That is, each knot's value in the sequence shall be
equal to or greater than the previous knot's value. The use of equal consecutive knots is normally handled
using the multiplicity.</documentation>
      </annotation>
    </element>
    <element name="multiplicity" type="nonNegativeInteger">
      <annotation>
        <documentation>The property "multiplicity" is the multiplicity of this knot used in the definition of the
spline (with the same weight).</documentation>
      </annotation>
    </element>
    <element name="weight" type="double">
      <annotation>
        <documentation>The property "weight" is the value of the averaging weight used for this knot of the
spline.</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
<!-- ============================================================ -->
<complexType name="KnotPropertyType">
  <annotation>
    <documentation>
                Encapsulates a knot to use it in a geometric type.
            </documentation>
  </annotation>
  <sequence>
    <element name="Knot" type="gml:KnotType"/>
  </sequence>
</complexType>
<!-- =========== global element in "_CurveSegment" substitution group ================ -->
<element name="BSpline" type="gml:BSplineType" substitutionGroup="gml:_CurveSegment"/>
<!-- ============================================================ -->
<complexType name="BSplineType">
  <annotation>
```

```xml
   <documentation>A B-Spline is a piecewise parametric polynomial or rational curve described in terms of
control points and basis functions. Knots are breakpoints on the curve that connect its pieces. They are
given as a non-decreasing sequence of real numbers. If the weights in the knots are equal then it is a
polynomial spline. The degree is the algebraic degree of the basis functions. </documentation>
  </annotation>
  <complexContent>
   <extension base="gml:AbstractCurveSegmentType">
    <sequence>
     <choice>
      <annotation>
       <documentation>GML supports two different ways to specify the control points of a curve segment.
1. A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements
are control points that are only part of this curve segment, "pointRep" elements contain a point that may be
referenced from other geometry elements or reference another point defined outside of this curve segment
(reuse of existing points).
2. The "coordinates" element allows for 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 curve segment
only.</documentation>
      </annotation>
      <choice minOccurs="0" maxOccurs="unbounded">
       <element ref="gml:pos"/>
       <element ref="gml:pointRep"/>
      </choice>
      <element ref="gml:coordinates"/>
     </choice>
     <element name="degree" type="nonNegativeInteger">
      <annotation>
       <documentation>The attribute "degree" shall be the degree of the polynomial used for interpolation
in this spline.</documentation>
      </annotation>
     </element>
     <element name="knot" type="gml:KnotPropertyType" minOccurs="2" maxOccurs="unbounded">
      <annotation>
       <documentation>The property "knot" shall be the sequence of distinct knots used to define the
spline basis functions.</documentation>
      </annotation>
     </element>
    </sequence>
    <attribute name="interpolation" type="gml:CurveInterpolationType" default="polynomialSpline">
     <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
BSpline the interpolation can be either "polynomialSpline" or "rationalSpline", default is
"polynomialSpline".</documentation>
     </annotation>
    </attribute>
    <attribute name="isPolynomial" type="boolean" use="optional">
     <annotation>
      <documentation>The attribute “isPolynomial” is set to “true” if this is a polynomial
spline.</documentation>
     </annotation>
    </attribute>
    <attribute name="knotType" type="gml:KnotTypesType" use="optional">
     <annotation>
      <documentation>The attribute "knotType" gives the type of knot distribution used in defining this
spline. This is for information only
and is set according to the different construction-functions.</documentation>
     </annotation>
    </attribute>
```

```
      </extension>
    </complexContent>
  </complexType>
  <!-- =========== global element in "_CurveSegment" substitution group ================ -->
  <element name="Bezier" type="gml:BezierType" substitutionGroup="gml:BSpline"/>
  <!-- ============================================================ -->
  <complexType name="BezierType">
    <annotation>
      <documentation>Bezier curves are polynomial splines that use Bezier or Bernstein polynomials for
interpolation purposes. It is a special case of the B-Spline curve with two knots.</documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:BSplineType">
        <sequence>
          <choice>
            <annotation>
              <documentation>GML supports two different ways to specify the control points of a curve segment.
1. A sequence of "pos" (DirectPositionType) or "pointRep" (PointPropertyType) elements. "pos" elements
are control points that are only part of this curve segment, "pointRep" elements contain a point that may be
referenced from other geometry elements or reference another point defined outside of this curve segment
(reuse of existing points).
2. The "coordinates" element allows for 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 curve segment
only.</documentation>
            </annotation>
            <choice minOccurs="0" maxOccurs="unbounded">
              <element ref="gml:pos"/>
              <element ref="gml:pointRep"/>
            </choice>
            <element ref="gml:coordinates"/>
          </choice>
          <element name="degree" type="nonNegativeInteger">
            <annotation>
              <documentation>The attribute "degree" shall be the degree of the polynomial used for interpolation
in this spline.</documentation>
            </annotation>
          </element>
          <element name="knot" type="gml:KnotPropertyType" minOccurs="2" maxOccurs="2">
            <annotation>
              <documentation>The property "knot" shall be the sequence of distinct knots used to define the
spline basis functions.</documentation>
            </annotation>
          </element>
        </sequence>
        <attribute name="interpolation" type="gml:CurveInterpolationType" fixed="polynomialSpline">
          <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 Bezier
the interpolation is fixed as "polynomialSpline".</documentation>
          </annotation>
        </attribute>
        <attribute name="isPolynomial" type="boolean" fixed="true">
          <annotation>
            <documentation>The attribute “isPolynomial” is set to “true” as this is a polynomial
spline.</documentation>
          </annotation>
        </attribute>
        <attribute name="knotType" type="gml:KnotTypesType" use="prohibited">
          <annotation>
```

```
      <documentation>The property "knotType" is not relevant for Bezier curve 
segments.</documentation>
     </annotation>
    </attribute>
   </restriction>
  </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>
 <!-- ============================================================== -->
 <element name="baseSurface" type="gml:SurfacePropertyType">
  <annotation>
   <appinfo>
    <sch:pattern>
     <sch:rule context="gml:baseSurface">
      <sch:extends rule="hrefOrContent"/>
     </sch:rule>
    </sch:pattern>
   </appinfo>
   <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>
 <!-- ============================================================== -->
 <element name="OrientableSurface" type="gml:OrientableSurfaceType" 
substitutionGroup="gml:_Surface"/>
 <!-- ============================================================== -->
 <complexType name="OrientableSurfaceType">
  <annotation>
   <documentation>
			OrientableSurface consists of a surface and an orientation. If the orientation is "+", then the 
OrientableSurface is identical to the baseSurface. If the orientation is "-", then the OrientableSurface is a 
```

```
reference to a Surface with an up-normal that reverses the direction for this OrientableSurface, the sense of
"the top of the surface".
                </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractSurfaceType">
        <sequence>
          <element ref="gml:baseSurface">
            <annotation>
              <documentation>References or contains the base surface (positive orientation).</documentation>
            </annotation>
          </element>
        </sequence>
        <attribute name="orientation" type="gml:SignType" default="+">
          <annotation>
            <documentation>If the orientation is "+", then the OrientableSurface is identical to the baseSurface.
If the orientation is "-", then the OrientableSurface is a reference to a Surface with an up-normal that
reverses the direction for this OrientableSurface, the sense of "the top of the surface". "+" is the default
value.</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>
  </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="Triangle" type="gml:TriangleType" substitutionGroup="gml:_SurfacePatch"/>
  <!-- ============================================================ -->
  <complexType name="TriangleType">
    <annotation>
      <documentation>Represents a triangle as a surface with an outer boundary consisting of a linear ring.
Note that this is a polygon (subtype) with no inner boundaries. The number of points in the linear ring must
be four.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractSurfacePatchType">
        <sequence>
          <element ref="gml:exterior">
            <annotation>
              <documentation>
                                  Constraint: The Ring shall be a LinearRing and must form a triangle, the first
and the last position must be co-incident.
                            </documentation>
            </annotation>
          </element>
        </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="Rectangle" type="gml:RectangleType" substitutionGroup="gml:_SurfacePatch"/>
```



```
 <!-- ============================================================== -->
 <complexType name="RectangleType">
  <annotation>
    <documentation>Represents a rectangle as a surface with an outer boundary consisting of a linear ring.
Note that this is a polygon (subtype) with no inner boundaries. The number of points in the linear ring must
be five.</documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractSurfacePatchType">
      <sequence>
        <element ref="gml:exterior">
          <annotation>
            <documentation>
                                    Constraint: The Ring shall be a LinearRing and must form a rectangle; the first
and the last position must be co-incident.
                              </documentation>
          </annotation>
        </element>
      </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="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>
 <!-- ============================================================== -->
 <element name="Ring" type="gml:RingType" substitutionGroup="gml:_Ring"/>
 <!-- ============================================================== -->
 <complexType name="RingType">
  <annotation>
    <documentation>A Ring is used to represent a single connected component of a surface boundary. It
consists of a sequence of curves connected in a cycle (an object whose boundary is empty).
A Ring is structurally similar to a composite curve in that the endPoint of each curve in the sequence is the
startPoint of the next curve in the Sequence. Since the sequence is circular, there is no exception to this
rule. Each ring, like all boundaries, is a cycle and each ring is simple.
NOTE: Even though each Ring is simple, the boundary need not be simple. The easiest case of this is
where one of the interior rings of a surface is tangent to its exterior ring.</documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractRingType">
      <sequence>
        <element ref="gml:curveMember" maxOccurs="unbounded">
          <annotation>
            <documentation>This element references or contains one curve in the composite curve. The curves
are contiguous, the collection of curves is ordered.
NOTE: This definition allows for a nested structure, i.e. a CompositeCurve may use, for example, another
CompositeCurve as a curve member.</documentation>
          </annotation>
```

```
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ============================================================ -->
<complexType name="RingPropertyType">
  <annotation>
    <documentation>
                  Encapsulates a ring to represent properties in features or geometry collections.
            </documentation>
  </annotation>
  <sequence>
    <element ref="gml:Ring"/>
  </sequence>
</complexType>
<!-- ============================================================ -->
<!-- primitive geometry objects (3-dimensional) -->
<!-- ============================================================ -->
<!-- ============================================================ -->
<element name="_Solid" type="gml:AbstractSolidType" abstract="true"
substitutionGroup="gml:_GeometricPrimitive">
  <annotation>
    <documentation>The "_Solid" element is the abstract head of the substituition group for all (continuous)
solid elements.</documentation>
  </annotation>
</element>
<!-- ============================================================ -->
<complexType name="AbstractSolidType">
  <annotation>
    <documentation>
                An abstraction of a solid to support the different levels of complexity. A solid is always
contiguous.
            </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractGeometricPrimitiveType"/>
  </complexContent>
</complexType>
<!-- ============================================================ -->
<element name="solidProperty" type="gml:SolidPropertyType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:solidProperty">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
    <documentation>This property element either references a solid via the XLink-attributes or contains the
solid element. solidProperty is the predefined property which can be used by GML Application Schemas
whenever a GML Feature has a property with a value that is substitutable for _Solid.</documentation>
  </annotation>
</element>
<!-- ============================================================ -->
<complexType name="SolidPropertyType">
  <annotation>
    <documentation>A property that has a solid 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>
      <element ref="gml:_Solid" minOccurs="0"/>
    </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="solidArrayProperty" type="gml:SolidArrayPropertyType"/>
  <!-- ============================================================ -->
  <complexType name="SolidArrayPropertyType">
    <annotation>
      <documentation>A container for an array of solids. The elements are always contained in the array property, referencing geometry elements or arrays of geometry elements is not supported.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:_Solid" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <!-- ============================================================ -->
  <element name="Solid" type="gml:SolidType" substitutionGroup="gml:_Solid"/>
  <!-- ============================================================ -->
  <complexType name="SolidType">
    <annotation>
      <documentation>A solid is the basis for 3-dimensional geometry. The extent of a solid is defined by the boundary surfaces (shells). A shell is represented by a composite surface, where every  shell is used to represent a single connected component of the boundary of a solid. It consists of a composite surface (a list of orientable surfaces) connected in a topological cycle (an object whose boundary is empty). Unlike a Ring, a Shell's elements have no natural sort order. Like Rings, Shells are simple.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractSolidType">
        <sequence>
          <element name="exterior" type="gml:SurfacePropertyType" minOccurs="0">
            <annotation>
              <appinfo>
                <sch:pattern>
                  <sch:rule context="gml:exterior">
                    <sch:extends rule="hrefOrContent"/>
                  </sch:rule>
                </sch:pattern>
              </appinfo>
              <documentation>Boundaries of solids are similar to surface boundaries. In normal 3-dimensional Euclidean space, one (composite) surface is distinguished as the exterior. In the more general case, this is not always possible.</documentation>
            </annotation>
          </element>
```

```
        <element name="interior" type="gml:SurfacePropertyType" minOccurs="0" maxOccurs="unbounded">
          <annotation>
            <appinfo>
              <sch:pattern>
                <sch:rule context="gml:interior">
                  <sch:extends rule="hrefOrContent"/>
                </sch:rule>
              </sch:pattern>
            </appinfo>
            <documentation>Boundaries of solids are similar to surface boundaries.</documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ============================================================== -->
<!-- predefined simple types (enumerations, simple typed arrays) -->
<!-- ============================================================== -->
<simpleType name="CurveInterpolationType">
  <annotation>
    <documentation>CurveInterpolationType 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="linear"/>
    <enumeration value="geodesic"/>
    <enumeration value="circularArc3Points"/>
    <enumeration value="circularArc2PointWithBulge"/>
    <enumeration value="circularArcCenterPointWithRadius"/>
    <enumeration value="elliptical"/>
    <enumeration value="clothoid"/>
    <enumeration value="conic"/>
    <enumeration value="polynomialSpline"/>
    <enumeration value="cubicSpline"/>
    <enumeration value="rationalSpline"/>
  </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="none"/>
    <enumeration value="planar"/>
    <enumeration value="spherical"/>
    <enumeration value="elliptical"/>
    <enumeration value="conic"/>
    <enumeration value="tin"/>
    <enumeration value="parametricCurve"/>
    <enumeration value="polynomialSpline"/>
    <enumeration value="rationalSpline"/>
    <enumeration value="triangulatedSpline"/>
  </restriction>
</simpleType>
<!-- ============================================================== -->
```



```
  <simpleType name="KnotTypesType">
    <annotation>
      <documentation>Defines allowed values for the knots` type. Uniform knots implies that all knots are of
multiplicity 1 and they differ by a positive constant from the preceding knot. Knots are quasi-uniform iff they
are of multiplicity (degree + 1) at the ends, of multiplicity 1 elsewhere, and they differ by a positive constant
from the preceding knot.</documentation>
    </annotation>
    <restriction base="string">
      <enumeration value="uniform"/>
      <enumeration value="quasiUniform"/>
      <enumeration value="piecewiseBezier"/>
    </restriction>
  </simpleType>
  <!-- ============================================================ -->
</schema>
```

geometryAggregates.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-
xsd:geometryAggregates:v3.00">geometryAggregates.xsd</appinfo>
    <documentation>
		Copyright (c) 2001-2002 OGC, All Rights Reserved.
    </documentation>
  </annotation>
  <include schemaLocation="geometryPrimitives.xsd"/>
  <!-- ============================================================ -->
  <!-- aggregate geometry objects -->
  <!-- ============================================================ -->
  <!-- ============================================================ -->
  <element name="_GeometricAggregate" type="gml:AbstractGeometricAggregateType" abstract="true"
substitutionGroup="gml:_Geometry">
    <annotation>
      <documentation>The "_GeometricAggregate" element is the abstract head of the substituition group for
all geometric aggremates.</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>
  <!-- ============================================================ -->
  <element name="MultiGeometry" type="gml:MultiGeometryType"
substitutionGroup="gml:_GeometricAggregate"/>
  <!-- ======================================================= -->
  <complexType name="MultiGeometryType">
    <annotation>
      <documentation>
```

```
                    A geometry collection must include one or more geometries, referenced through
geometryMember elements.
               </documentation>
   </annotation>
   <complexContent>
     <extension base="gml:AbstractGeometricAggregateType">
       <sequence>
         <annotation>
           <documentation>The members of the geometric aggregate can be specified either using the
"standard" property or the array property style. It is also valid to use both the "standard" and the array
property style in the same collection.
NOTE: Array properties cannot reference remote geometry elements.</documentation>
         </annotation>
         <element ref="gml:geometryMember" minOccurs="0" maxOccurs="unbounded"/>
         <element ref="gml:geometryMembers" minOccurs="0"/>
       </sequence>
     </extension>
   </complexContent>
 </complexType>
 <!-- ============================================================== -->
 <element name="multiGeometryProperty" type="gml:MultiGeometryPropertyType">
   <annotation>
     <appinfo>
       <sch:pattern>
         <sch:rule context="gml:multiGeometryProperty">
           <sch:extends rule="hrefOrContent"/>
         </sch:rule>
       </sch:pattern>
     </appinfo>
     <documentation>This property element either references a geometric aggregate via the XLink-attributes
or contains the "multi geometry" element. multiGeometryProperty is the predefined property which can be
used by GML Application Schemas whenever a GML Feature has a property with a value that is
substitutable for _GeometricAggregate.</documentation>
   </annotation>
 </element>
 <!-- ============================================================== -->
 <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>
     <element ref="gml:_GeometricAggregate" minOccurs="0"/>
   </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>
          <annotation>
            <documentation>The members of the geometric aggregate can be specified either using the
"standard" property or the array property style. It is also valid to use both the "standard" and the array
property style in the same collection.
NOTE: Array properties cannot reference remote geometry elements.</documentation>
          </annotation>
          <element ref="gml:pointMember" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:pointMembers" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- =========================================================== -->
  <element name="multiPointProperty" type="gml:MultiPointPropertyType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:multiGeometryProperty">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
      <documentation>This property element either references a point aggregate via the XLink-attributes or
contains the "multi point" element. multiPointProperty is the predefined property which can be used by GML
Application Schemas whenever a GML Feature has a property with a value that is substitutable for
MultiPoint.</documentation>
    </annotation>
  </element>
  <!-- =========================================================== -->
  <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>
      <element ref="gml:MultiPoint" minOccurs="0"/>
    </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>
      <annotation>
       <documentation>The members of the geometric aggregate can be specified either using the
"standard" property or the array property style. It is also valid to use both the "standard" and the array
property style in the same collection.
NOTE: Array properties cannot reference remote geometry elements.</documentation>
      </annotation>
      <element ref="gml:curveMember" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="gml:curveMembers" minOccurs="0"/>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="multiCurveProperty" type="gml:MultiCurvePropertyType">
   <annotation>
    <appinfo>
     <sch:pattern>
      <sch:rule context="gml:multiCurveProperty">
       <sch:extends rule="hrefOrContent"/>
      </sch:rule>
     </sch:pattern>
    </appinfo>
    <documentation>This property element either references a curve aggregate via the XLink-attributes or
contains the "multi curve" element. multiCurveProperty is the predefined property which can be used by
GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for
MultiCurve.</documentation>
   </annotation>
  </element>
  <!-- ============================================================ -->
  <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>
    <element ref="gml:MultiCurve" minOccurs="0"/>
   </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>
				<annotation>
					<documentation>The members of the geometric aggregate can be specified either using the
"standard" property or the array property style. It is also valid to use both the "standard" and the array
property style in the same collection.
NOTE: Array properties cannot reference remote geometry elements.</documentation>
				</annotation>
				<element ref="gml:surfaceMember" minOccurs="0" maxOccurs="unbounded"/>
				<element ref="gml:surfaceMembers" minOccurs="0"/>
			</sequence>
		</extension>
	</complexContent>
</complexType>
<!-- =========================================================== -->
<element name="multiSurfaceProperty" type="gml:MultiSurfacePropertyType">
	<annotation>
		<appinfo>
			<sch:pattern>
				<sch:rule context="gml:multiSurfaceProperty">
					<sch:extends rule="hrefOrContent"/>
				</sch:rule>
			</sch:pattern>
		</appinfo>
		<documentation>This property element either references a surface aggregate via the XLink-attributes or
contains the "multi surface" element. multiSurfaceProperty is the predefined property which can be used by
GML Application Schemas whenever a GML Feature has a property with a value that is substitutable for
MultiSurface.</documentation>
	</annotation>
</element>
<!-- =========================================================== -->
<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>
		<element ref="gml:MultiSurface" minOccurs="0"/>
```

```
    </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="MultiSolid" type="gml:MultiSolidType" substitutionGroup="gml:_GeometricAggregate"/>
  <!-- ========================================================== -->
  <complexType name="MultiSolidType">
    <annotation>
      <documentation>
                        A MultiSolid is defined by one or more Solids, referenced through solidMember elements.
                  </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGeometricAggregateType">
        <sequence>
          <annotation>
            <documentation>The members of the geometric aggregate can be specified either using the
"standard" property or the array property style. It is also valid to use both the "standard" and the array
property style in the same collection.
NOTE: Array properties cannot reference remote geometry elements.</documentation>
          </annotation>
          <element ref="gml:solidMember" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:solidMembers" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- =========================================================== -->
  <element name="multiSolidProperty" type="gml:MultiSolidPropertyType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:multiSolidProperty">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
      <documentation>This property element either references a solid aggregate via the XLink-attributes or
contains the "multi solid" element. multiSolidProperty is the predefined property which can be used by GML
Application Schemas whenever a GML Feature has a property with a value that is substitutable for
MultiSolid.</documentation>
    </annotation>
  </element>
  <!-- =========================================================== -->
  <complexType name="MultiSolidPropertyType">
    <annotation>
      <documentation>A property that has a collection of solids 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>
     <element ref="gml:MultiSolid" minOccurs="0"/>
   </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>
 <!-- ============================================================ -->
 <!--

     The following types and elements are deprecated and should not be used !
     For backward compatibility with GML2 only

     -->
 <!-- ============================================================ -->
 <element name="MultiPolygon" type="gml:MultiPolygonType"
substitutionGroup="gml:_GeometricAggregate">
   <annotation>
     <documentation>Deprecated with GML 3.0 and included for backwards compatibility with GML 2. Use
the "MultiSurface" element instead.</documentation>
   </annotation>
 </element>
 <!-- ============================================================ -->
 <element name="MultiLineString" type="gml:MultiLineStringType"
substitutionGroup="gml:_GeometricAggregate">
   <annotation>
     <documentation>Deprecated with GML 3.0 and included for backwards compatibility with GML 2. Use
the "MultiCurve" element instead.</documentation>
   </annotation>
 </element>
 <!-- ============================================================ -->
 <complexType name="MultiLineStringType">
   <annotation>
     <documentation>
                    A MultiLineString is defined by one or more LineStrings, referenced through
lineStringMember elements. Deprecated with GML version 3.0. Use MultiCurveType instead.
               </documentation>
   </annotation>
   <complexContent>
     <extension base="gml:AbstractGeometricAggregateType">
       <sequence>
         <element ref="gml:lineStringMember" minOccurs="0" maxOccurs="unbounded"/>
       </sequence>
     </extension>
   </complexContent>
 </complexType>
 <!-- ============================================================ -->
```



```xml
  <complexType name="MultiLineStringPropertyType">
    <annotation>
      <documentation>This type is deprecated with GML 3 and shall not be used. It is included for backwards
compatibility with GML 2. Use MultiCurvePropertyType instead.
A property that has a collection of line strings 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>
      <element ref="gml:MultiLineString" minOccurs="0"/>
    </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="MultiPolygonType">
    <annotation>
      <documentation>
                        A MultiPolygon is defined by one or more Polygons, referenced through polygonMember
elements. Deprecated with GML version 3.0. Use MultiSurfaceType instead.
                  </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGeometricAggregateType">
        <sequence>
          <element ref="gml:polygonMember" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================== -->
  <complexType name="MultiPolygonPropertyType">
    <annotation>
      <documentation>This type is deprecated with GML 3 and shall not be used. It is included for backwards
compatibility with GML 2. Use MultiSurfacePropertyType instead.

A property that has a collection of polygons 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>
      <element ref="gml:MultiPolygon" minOccurs="0"/>
    </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="geometryMember" type="gml:GeometryPropertyType">
    <annotation>
      <documentation>This property element either references a geometry element via the XLink-attributes or
contains the geometry element.</documentation>
    </annotation>
  </element>
  <element name="geometryMembers" type="gml:GeometryArrayPropertyType">
    <annotation>
      <documentation>This property element contains a list of geometry elements. The order of the elements
is significant and shall be preserved when processing the array.</documentation>
    </annotation>
  </element>
  <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="pointMembers" type="gml:PointArrayPropertyType">
    <annotation>
      <documentation>This property element contains a list of points. The order of the elements is significant
and shall be preserved when processing the array.</documentation>
    </annotation>
  </element>
  <element name="curveMembers" type="gml:CurveArrayPropertyType">
    <annotation>
      <documentation>This property element contains a list of curves. The order of the elements is significant
and shall be preserved when processing the array.</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>
  <element name="surfaceMembers" type="gml:SurfaceArrayPropertyType">
    <annotation>
      <documentation>This property element contains a list of surfaces. The order of the elements is
significant and shall be preserved when processing the array.</documentation>
    </annotation>
  </element>
  <element name="solidMember" type="gml:SolidPropertyType">
    <annotation>
      <documentation>This property element either references a solid via the XLink-attributes or contains the
solid element. A solid element is any element which is substitutable for "_Solid".</documentation>
    </annotation>
```



```
  </element>
  <element name="solidMembers" type="gml:SolidArrayPropertyType">
    <annotation>
      <documentation>This property element contains a list of solids. The order of the elements is significant
and shall be preserved when processing the array.</documentation>
    </annotation>
  </element>
  <!-- some named geometry properties - for backward compatibility with GML2 -->
  <element name="multiCenterOf" type="gml:MultiPointPropertyType"
substitutionGroup="gml:multiPointProperty"/>
  <element name="multiPosition" type="gml:MultiPointPropertyType"
substitutionGroup="gml:multiPointProperty"/>
  <element name="multiCenterLineOf" type="gml:MultiCurvePropertyType"
substitutionGroup="gml:multiCurveProperty"/>
  <element name="multiEdgeOf" type="gml:MultiCurvePropertyType"
substitutionGroup="gml:multiCurveProperty"/>
  <element name="multiCoverage" type="gml:MultiSurfacePropertyType"
substitutionGroup="gml:multiSurfaceProperty"/>
  <element name="multiExtentOf" type="gml:MultiSurfacePropertyType"
substitutionGroup="gml:multiSurfaceProperty"/>
  <!--

      The following types and elements are deprecated and should not be used !

      -->
  <element name="multiLocation" type="gml:MultiPointPropertyType"
substitutionGroup="gml:multiPointProperty">
    <annotation>
      <appinfo>deprecated</appinfo>
      <documentation>deprecated as of version 3.0b1, 2001-07</documentation>
    </annotation>
  </element>
  <element name="lineStringMember" type="gml:LineStringPropertyType">
    <annotation>
      <documentation>Deprecated with GML 3.0 and included only for backwards compatibility with GML 2.0.
Use "curveMember" instead.
This property element either references a line string via the XLink-attributes or contains the line string
element.</documentation>
    </annotation>
  </element>
  <element name="polygonMember" type="gml:PolygonPropertyType">
    <annotation>
      <documentation>Deprecated with GML 3.0 and included only for backwards compatibility with GML 2.0.
Use "surfaceMember" instead.
This property element either references a polygon via the XLink-attributes or contains the polygon
element.</documentation>
    </annotation>
  </element>
  <!-- ============================================================== -->
</schema>
```

geometryComplexes.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified" version="3.00">
  <annotation>
```

```
    <appinfo source="urn:opengis:specification:gml:schema-
xsd:geometryComplexes:v3.00">geometryComplexes.xsd</appinfo>
    <documentation>
              Copyright (c) 2001-2002 OGC, All Rights Reserved.
    </documentation>
  </annotation>
  <include schemaLocation="geometryAggregates.xsd"/>
  <!-- ============================================================ -->
  <element name="CompositeCurve" type="gml:CompositeCurveType" substitutionGroup="gml:_Curve"/>
  <!-- ============================================================ -->
  <complexType name="CompositeCurveType">
    <annotation>
      <documentation>
                   A CompositeCurve is defined by a sequence of (orientable) curves such that the each curve
in the sequence terminates at the start point of the subsequent curve in the list.
              </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCurveType">
        <sequence>
          <element ref="gml:curveMember" maxOccurs="unbounded">
            <annotation>
              <documentation>This element references or contains one curve in the composite curve. The curves
are contiguous, the collection of curves is ordered.
NOTE: This definition allows for a nested structure, i.e. a CompositeCurve may use, for example, another
CompositeCurve as a curve member.</documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="CompositeSurface" type="gml:CompositeSurfaceType"
substitutionGroup="gml:_Surface"/>
  <!-- ============================================================ -->
  <complexType name="CompositeSurfaceType">
    <annotation>
      <documentation>A CompositeSurface is defined by a set of orientable surfaces. A composite surface is
geometry type with all the geometric properties of a (primitive) surface. Essentially, a composite surface is a
collection of surfaces that join in pairs on common boundary curves and which, when considered as a
whole, form a single surface. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractSurfaceType">
        <sequence>
          <element ref="gml:surfaceMember" maxOccurs="unbounded">
            <annotation>
              <documentation>This element references or contains one surface in the composite surface. The
surfaces are contiguous.
NOTE: This definition allows for a nested structure, i.e. a CompositeSurface may use, for example, another
CompositeSurface as a member.</documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="CompositeSolid" type="gml:CompositeSolidType" substitutionGroup="gml:_Solid"/>
```

```
  <!-- ============================================================ -->
  <complexType name="CompositeSolidType">
    <annotation>
      <documentation>
                  A composite solid is a geometry type with all the geometric properties of a (primitive) solid.
                  Essentially, a composite solid is a collection of solids that join in pairs on common boundary
surfaces and which, when considered as a whole, form a single solid.
               </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractSolidType">
        <sequence>
          <element ref="gml:solidMember" maxOccurs="unbounded">
            <annotation>
              <appinfo>
                <sch:pattern>
                  <sch:rule context="gml:solidMember">
                    <sch:extends rule="hrefOrContent"/>
                  </sch:rule>
                </sch:pattern>
              </appinfo>
              <documentation>This element references or contains one solid in the composite solid. The solids
are contiguous.
NOTE: This definition allows for a nested structure, i.e. a CompositeSolid may use, for example, another
CompositeSolid as a member.</documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <!-- complex/composite geometry objects -->
  <!-- ============================================================ -->
  <element name="GeometricComplex" type="gml:GeometricComplexType"
substitutionGroup="gml:_Geometry"/>
  <!-- ============================================================ -->
  <complexType name="GeometricComplexType">
    <annotation>
      <documentation>
                  A geometric complex.
               </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGeometryType">
        <sequence>
          <element name="element" type="gml:GeometricPrimitivePropertyType" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <complexType name="GeometricComplexPropertyType">
    <annotation>
      <documentation>A property that has a geometric complex 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.
NOTE: The allowed geometry elements contained in such a property (or referenced by it) have to be
modelled by an XML Schema choice element since the composites inherit both from geometric complex
```

```
*and* geometric primitive and are already part of the _GeometricPrimitive substitution
group.</documentation>
    </annotation>
    <choice minOccurs="0">
      <element ref="gml:GeometricComplex"/>
      <element ref="gml:CompositeCurve"/>
      <element ref="gml:CompositeSurface"/>
      <element ref="gml:CompositeSolid"/>
    </choice>
    <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>
  <!-- ============================================================== -->
</schema>
```

grids.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:grids:v3.00">grids.xsd</appinfo>
    <documentation xml:lang="en">Grid geometries
    A subset of implicit geometries
    Designed for use with GML Coverage schema, but maybe useful elsewhere as well.

    Copyright (c) 2002 OGC, All Rights Reserved.
    </documentation>
  </annotation>
  <!-- ==============================================================
       includes and imports
       ============================================================== -->
  <include schemaLocation="geometryBasic0d1d.xsd"/>
  <!-- ==============================================================
       global elements
       ============================================================== -->
  <element name="_ImplicitGeometry" type="gml:AbstractGeometryType" abstract="true"
substitutionGroup="gml:_Geometry"/>
  <!-- ============================================================== -->
  <element name="Grid" type="gml:GridType" substitutionGroup="gml:_ImplicitGeometry"/>
  <!-- ============================================================== -->
  <complexType name="GridType">
    <annotation>
      <documentation>Implicitly defines an unrectified grid, which is a network composed of two or more sets
of equally spaced parallel lines in which the members of each set intersect the members of the other sets at
right angles.</documentation>
    </annotation>
```



```
    <complexContent>
      <extension base="gml:AbstractGeometryType">
        <sequence>
          <element name="limits" type="gml:GridLimitsType"/>
          <element name="axisName" type="string" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="dimension" type="positiveInteger" use="required"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <complexType name="GridLimitsType">
    <sequence>
      <element name="GridEnvelope" type="gml:GridEnvelopeType"/>
    </sequence>
  </complexType>
  <!-- ============================================================ -->
  <complexType name="GridEnvelopeType">
    <annotation>
      <documentation>Provides grid coordinate values for the diametrically opposed corners of an envelope
that bounds a section of grid. The value of a single coordinate is the number of offsets from the origin of the
grid in the direction of a specific axis.</documentation>
    </annotation>
    <sequence>
      <element name="low" type="gml:integerList"/>
      <element name="high" type="gml:integerList"/>
    </sequence>
  </complexType>
  <!-- ============================================================ -->
  <element name="RectifiedGrid" type="gml:RectifiedGridType" substitutionGroup="gml:Grid"/>
  <!-- ============================================================ -->
  <complexType name="RectifiedGridType">
    <annotation>
      <documentation>A rectified grid has an origin and vectors that define its post
locations.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:GridType">
        <sequence>
          <element name="origin" type="gml:PointPropertyType"/>
          <element name="offsetVector" type="gml:VectorType" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
</schema>
```

topology.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:topology:v3.00">topology.xsd</appinfo>
    <documentation>
    </documentation>
```

```
  </annotation>
  <include schemaLocation="geometryComplexes.xsd"/>
  <!-- ==============================================================
       abstract supertype for topology objects
       ============================================================== -->
  <!-- ========================================================== -->
  <element name="_Topology" type="gml:AbstractTopologyType" abstract="true"
substitutionGroup="gml:_Object"/>
  <!-- ========================================================== -->
  <complexType name="AbstractTopologyType" abstract="true">
    <complexContent>
      <extension base="gml:AbstractGMLType"/>
    </complexContent>
  </complexType>
  <!-- ========================================================== -->
  <element name="_TopoPrimitive" type="gml:AbstractTopoPrimitiveType" abstract="true"
substitutionGroup="gml:_Topology">
    <annotation>
      <documentation>Substitution group branch for Topo Primitives, used by
TopoPrimitiveArrayAssociationType</documentation>
    </annotation>
  </element>
  <!-- ========================================================== -->
  <complexType name="AbstractTopoPrimitiveType" abstract="true">
    <complexContent>
      <extension base="gml:AbstractTopologyType">
        <sequence>
          <element ref="gml:isolated" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:container" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ========================================================== -->
  <element name="isolated" type="gml:IsolatedPropertyType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:isolated">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
    </annotation>
  </element>
  <!-- ========================================================== -->
  <complexType name="IsolatedPropertyType">
    <choice minOccurs="0">
      <element ref="gml:Node"/>
      <element ref="gml:Edge"/>
    </choice>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ========================================================== -->
  <element name="container" type="gml:ContainerPropertyType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:containerProperty">
            <sch:extends rule="hrefOrContent"/>
```

```
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
<!-- ============================================================ -->
<complexType name="ContainerPropertyType">
  <choice minOccurs="0">
    <element ref="gml:Face"/>
    <element ref="gml:TopoSolid"/>
  </choice>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!-- ============================================================= -->
<!-- primitive topology objects -->
<!-- ============================================================ -->
<element name="Node" type="gml:NodeType" substitutionGroup="gml:_TopoPrimitive"/>
<!-- ============================================================= -->
<complexType name="NodeType">
  <annotation>
    <documentation> Its optional co-boundary is a set of connected directedEdges.  The orientation of one
of these dirEdges is "+" if the Node is the "to" node of the Edge, and "-" if it is the "from" node.
</documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractTopoPrimitiveType">
      <sequence>
        <element ref="gml:directedEdge" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:pointProperty" minOccurs="0"/>
        <!-- <element name="geometry" type="gml:PointPropertyType" minOccurs="0"/> -->
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ===== Property for topology association - by Value or by Reference ===== -->
<element name="directedNode" type="gml:DirectedNodePropertyType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:directedNode">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
<!-- ============================================================= -->
<complexType name="DirectedNodePropertyType">
  <choice>
    <element ref="gml:Node" minOccurs="0"/>
  </choice>
  <attribute name="orientation" type="gml:SignType" default="+"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!-- ============================================================ -->
<!-- primitive topology objects (1-dimensional) -->
<!-- ============================================================ -->
<element name="Edge" type="gml:EdgeType" substitutionGroup="gml:_TopoPrimitive"/>
<!-- ============================================================ -->
<complexType name="EdgeType">
```

```
    <annotation>
      <documentation>There is precisely one positively directed and one negatively directed node in the
boundary of every edge. The negatively and positively directed nodes correspond to the start and end nodes
respectively. The optional coboundary of an edge is a circular sequence of directed faces which are incident
on this edge in document order. Faces which use a particular boundary edge in its positive orientation
appear with positive orientation on the coboundary of the same edge. In the 2D case, the orientation of the
face on the left of the edge is "+"; the orientation of the face on the right on its right is "-". An edge may
optionally be realised by a 1-dimensional (curve) geometric primitive.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractTopoPrimitiveType">
        <sequence>
          <element ref="gml:directedNode" minOccurs="2" maxOccurs="2"/>
          <element ref="gml:directedFace" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:curveProperty" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ===== Property for topology association - by Value or by Reference ===== -->
  <element name="directedEdge" type="gml:DirectedEdgePropertyType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:directedEdge">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <complexType name="DirectedEdgePropertyType">
    <choice>
      <element ref="gml:Edge" minOccurs="0"/>
    </choice>
    <attribute name="orientation" type="gml:SignType" default="+"/>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- =========================================================== -->
  <!-- primitive topology objects (2-dimensional) -->
  <!-- =========================================================== -->
  <element name="Face" type="gml:FaceType" substitutionGroup="gml:_TopoPrimitive"/>
  <!-- =========================================================== -->
  <complexType name="FaceType">
    <annotation>
      <documentation>. The topological boundary of a face consists of a set of directed edges. Note that all
edges associated with a Face, including dangling and interior edges, appear in the boundary.  Dangling and
interior edges are each referenced by pairs of directedEdges with opposing orientations.  The optional
coboundary of a face is a pair of directed solids which are bounded by this face. If present, there is precisely
one positively directed and one negatively directed solid in the coboundary of every face. The positively
directed solid corresponds to the solid which lies in the direction of the positively directed normal to the face
in any geometric realisation.  A face may optionally be realised by a 2-dimensional (surface) geometric
primitive.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractTopoPrimitiveType">
        <sequence>
          <element ref="gml:directedEdge" maxOccurs="unbounded"/>
          <element ref="gml:directedTopoSolid" minOccurs="0" maxOccurs="2"/>
```

```
        <element ref="gml:surfaceProperty" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ===== Property for topology association - by Value or by Reference ===== -->
<element name="directedFace" type="gml:DirectedFacePropertyType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:directedFace">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
<!-- ============================================================== -->
<complexType name="DirectedFacePropertyType">
  <choice>
    <element ref="gml:Face" minOccurs="0"/>
  </choice>
  <attribute name="orientation" type="gml:SignType" default="+"/>
  <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
<!-- ============================================================= -->
<!-- primitive topology objects (3-dimensional) -->
<!-- ============================================================= -->
<element name="TopoSolid" type="gml:TopoSolidType" substitutionGroup="gml:_TopoPrimitive"/>
<!-- ============================================================== -->
<complexType name="TopoSolidType">
  <annotation>
    <documentation>The topological boundary of a TopoSolid consists of a set of directed faces. Note that
all faces associated with the TopoSolid, including dangling faces, appear in the boundary. The coboundary
of a TopoSolid is empty and hence requires no representation.</documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractTopoPrimitiveType">
      <sequence>
        <element ref="gml:directedFace" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ============================================================= -->
<!-- ===== Property for topology association - by Value or by Reference ===== -->
<element name="directedTopoSolid" type="gml:DirectedTopoSolidPropertyType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:directedTopoSolid">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
<!-- ============================================================== -->
<complexType name="DirectedTopoSolidPropertyType">
  <choice>
```

```
      <element ref="gml:TopoSolid"/>
    </choice>
    <attribute name="orientation" type="gml:SignType" default="+"/>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ========================================================== -->
  <element name="TopoPoint" type="gml:TopoPointType"/>
  <!-- ========================================================== -->
  <complexType name="TopoPointType">
    <sequence>
      <element ref="gml:directedNode"/>
    </sequence>
  </complexType>
  <!-- ========================================================== -->
  <!-- ============================================================= -->
  <!-- ===== Property for topology association - by Value  ===== -->
  <element name="topoPointProperty" type="gml:TopoPointPropertyType"/>
  <!-- ============================================================= -->
  <complexType name="TopoPointPropertyType">
    <sequence>
      <element ref="gml:TopoPoint"/>
    </sequence>
  </complexType>
  <!-- ========================================================== -->
  <!-- ========================================================== -->
  <element name="TopoCurve" type="gml:TopoCurveType"/>
  <!-- ============================================================= -->
  <complexType name="TopoCurveType">
    <annotation>
      <documentation>The directed edges of a TopoCurveType when joined in document order shall be
isomorphic to a gml:_Curve in any geometric realization.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:directedEdge" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <!-- ========================================================== -->
  <!-- ===== Property for topology association - by Value ===== -->
  <element name="topoCurveProperty" type="gml:TopoCurvePropertyType"/>
  <!-- ============================================================= -->
  <complexType name="TopoCurvePropertyType">
    <sequence>
      <element ref="gml:TopoCurve"/>
    </sequence>
  </complexType>
  <!-- ========================================================== -->
  <!-- ========================================================== -->
  <element name="TopoSurface" type="gml:TopoSurfaceType"/>
  <!-- ============================================================= -->
  <complexType name="TopoSurfaceType">
    <annotation>
      <documentation/>
    </annotation>
    <sequence>
      <element ref="gml:directedFace" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <!-- ========================================================== -->
  <!-- ===== Property for topology association - by Value ===== -->
  <element name="topoSurfaceProperty" type="gml:TopoSurfacePropertyType"/>
```

```
<!-- ============================================================== -->
<complexType name="TopoSurfacePropertyType">
  <sequence>
    <element ref="gml:TopoSurface"/>
  </sequence>
</complexType>
<!-- =========================================================== -->
<element name="TopoVolume" type="gml:TopoVolumeType"/>
<!-- ============================================================== -->
<complexType name="TopoVolumeType">
  <sequence>
    <element ref="gml:directedTopoSolid" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<!-- =========================================================== -->
<!-- ===== Property for topology association - by Value  ===== -->
<element name="topoVolumeProperty" type="gml:TopoVolumePropertyType"/>
<!-- ============================================================== -->
<complexType name="TopoVolumePropertyType">
  <sequence>
    <element ref="gml:TopoVolume"/>
  </sequence>
</complexType>
<!-- =========================================================== -->
<!-- =========================================================== -->
<element name="TopoComplex" type="gml:TopoComplexType" substitutionGroup="gml:_Topology"/>
<!-- =========================================================== -->
<complexType name="TopoComplexType">
  <annotation>
    <documentation>This type represents a TP_Complex capable of holding topological
primitives.</documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractTopologyType">
      <sequence>
        <element ref="gml:maximalComplex"/>
        <element ref="gml:superComplex" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:subComplex" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:topoPrimitiveMember" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:topoPrimitiveMembers" minOccurs="0"/>
      </sequence>
      <attribute name="isMaximal" type="boolean" default="false"/>
    </extension>
  </complexContent>
</complexType>
<!-- ===== Property for topology association - by Value or Reference ===== -->
<element name="topoComplexProperty" type="gml:TopoComplexMemberType"/>
<!-- =========================================================== -->
<element name="subComplex" type="gml:TopoComplexMemberType">
  <annotation>
    <appinfo>
      <sch:pattern>
        <sch:rule context="gml:subComplex">
          <sch:extends rule="hrefOrContent"/>
        </sch:rule>
      </sch:pattern>
    </appinfo>
  </annotation>
</element>
<!-- =========================================================== -->
```

```
  <element name="superComplex" type="gml:TopoComplexMemberType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:superComplex">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <element name="maximalComplex" type="gml:TopoComplexMemberType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:subComplex">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
      <documentation>Need schamatron test here that isMaximal attribute value is true</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <complexType name="TopoComplexMemberType">
    <annotation>
      <documentation>This Property can be used to embed a TopoComplex in a feature
collection.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:TopoComplex" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ============================================================ -->
  <!-- ===== Property for topology association - by Value or Reference ===== -->
  <element name="topoPrimitiveMember" type="gml:topoPrimitiveMemberType">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:topoPrimitiveMember">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <complexType name="topoPrimitiveMemberType">
    <annotation>
      <documentation>This type supports embedding topological primitives in a
TopoComplex.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:_TopoPrimitive" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ============================================================ -->
```

```
  <!-- ===== Property for topology association - by Value ===== -->
  <element name="topoPrimitiveMembers" type="gml:TopoPrimitiveArrayAssociationType"
substitutionGroup="gml:members">
    <annotation>
      <appinfo>
        <sch:pattern>
          <sch:rule context="gml:topoPrimitiveMember">
            <sch:extends rule="hrefOrContent"/>
          </sch:rule>
        </sch:pattern>
      </appinfo>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <complexType name="TopoPrimitiveArrayAssociationType">
    <annotation>
      <documentation>This type supports embedding an array of topological primitives in a
TopoComplex</documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:ArrayAssociationType">
        <sequence>
          <element ref="gml:_TopoPrimitive" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
</schema>
```

direction.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:gml="http://www.opengis.net/gml"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:direction:v3.00">direction.xsd</appinfo>
    <documentation>
		This schema defines "direction" element and type.
    </documentation>
  </annotation>
  <!-- ============================================================
      includes and imports
      ============================================================ -->
  <include schemaLocation="geometryBasic0d1d.xsd"/>
  <!-- ============================================================ -->
  <!--==================================================================  -->
  <element name="direction" type="gml:DirectionPropertyType"/>
  <!--==================================================================  -->
  <complexType name="DirectionPropertyType">
    <annotation>
      <documentation/>
    </annotation>
    <choice>
      <element ref="gml:DirectionVector"/>
      <element ref="gml:CompassPoint"/>
      <element name="DirectionKeyword" type="gml:CodeType"/>
```

```
      <element name="DirectionString" type="gml:StringOrRefType"/>
    </choice>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!--===================================================================== -->
  <element name="DirectionVector" type="gml:DirectionVectorType"/>
  <!--===================================================================== -->
  <complexType name="DirectionVectorType">
    <annotation>
      <documentation>Direction expressed as a vector, either using components, or using angles.
</documentation>
    </annotation>
    <choice>
      <element ref="gml:vector"/>
      <sequence>
        <element name="horizontalAngle" type="gml:AngleType"/>
        <element name="verticalAngle" type="gml:AngleType"/>
      </sequence>
    </choice>
  </complexType>
  <!--===================================================================== -->
  <element name="CompassPoint" type="gml:CompassPointEnumeration"/>
  <!--===================================================================== -->
  <simpleType name="CompassPointEnumeration">
    <restriction base="string">
      <enumeration value="N"/>
      <enumeration value="NNE"/>
      <enumeration value="NE"/>
      <enumeration value="ENE"/>
      <enumeration value="E"/>
      <enumeration value="ESE"/>
      <enumeration value="SE"/>
      <enumeration value="SSE"/>
      <enumeration value="S"/>
      <enumeration value="SSW"/>
      <enumeration value="SW"/>
      <enumeration value="WSW"/>
      <enumeration value="W"/>
      <enumeration value="WNW"/>
      <enumeration value="NW"/>
      <enumeration value="NNW"/>
    </restriction>
  </simpleType>
  <!--===================================================================== -->
</schema>
```

feature.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:gml="http://www.opengis.net/gml"
xmlns:sch="http://www.ascc.net/xml/schematron" xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:feature:v3.00"/>
    <documentation>
            GML Feature schema.
            Copyright (c) 2001-2002 OGC, All Rights Reserved.
```

```
    </documentation>
  </annotation>
  <!-- ============================================================== -->
  <!-- ==================       includes and imports  ====================== -->
  <!-- ============================================================== -->
  <include schemaLocation="geometryBasic2d.xsd"/>
  <include schemaLocation="temporal.xsd"/>
  <!-- ============================================================== -->
  <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,
metaDataProperty, 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. A feature
must possess an identifying attribute ('id' - 'fid' has been deprecated).  </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGMLType">
        <sequence>
          <element ref="gml:boundedBy" minOccurs="0"/>
          <element ref="gml:location" minOccurs="0"/>
          <!-- additional properties must be specified in an application schema -->
        </sequence>
        <attribute name="fid" type="string">
          <annotation>
            <appinfo>deprecated</appinfo>
            <documentation>deprecated in GML version 3.0</documentation>
          </annotation>
        </attribute>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================== -->
  <!-- ===== property for feature association - by Value or by Reference ===== -->
  <element name="featureMember" type="gml:FeaturePropertyType"/>
  <element name="featureProperty" type="gml:FeaturePropertyType"/>
  <!-- ============================================================== -->
  <complexType name="FeaturePropertyType">
    <annotation>
      <documentation>Container for a feature - follow gml:AssociationType pattern.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:_Feature" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ============================================================== -->
  <!-- ===== property for association of an array of features ===== -->
  <element name="featureMembers" type="gml:FeatureArrayPropertyType"/>
  <!-- =========================================================== -->
  <complexType name="FeatureArrayPropertyType">
    <annotation>
      <documentation>Container for features - follow gml:ArrayAssociationType pattern.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:_Feature" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
```



```
 <!-- ================================================================ -->
 <!-- =========== Bounded feature ======================== -->
 <complexType name="BoundedFeatureType" abstract="true">
   <annotation>
     <documentation> Makes boundedBy mandatory - used to build Feature Collections </documentation>
   </annotation>
   <complexContent>
     <restriction base="gml:AbstractFeatureType">
       <sequence>
         <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
         <element ref="gml:description" minOccurs="0"/>
         <element ref="gml:name" minOccurs="0" maxOccurs="unbounded"/>
         <element ref="gml:boundedBy"/>
         <element ref="gml:location" minOccurs="0"/>
       </sequence>
     </restriction>
   </complexContent>
 </complexType>
 <!-- ============================================================ -->
 <element name="_FeatureCollection" type="gml:AbstractFeatureCollectionType" abstract="true"
substitutionGroup="gml:_Feature"/>
 <!-- ============================================================   -->
 <complexType name="AbstractFeatureCollectionType" abstract="true">
   <annotation>
     <documentation> A feature collection contains zero or more features.  </documentation>
   </annotation>
   <complexContent>
     <extension base="gml:BoundedFeatureType">
       <sequence>
         <element ref="gml:featureMember" minOccurs="0" maxOccurs="unbounded"/>
         <element ref="gml:featureMembers" minOccurs="0"/>
       </sequence>
     </extension>
   </complexContent>
 </complexType>
 <!-- ============================================================ -->
 <element name="FeatureCollection" type="gml:FeatureCollectionType"
substitutionGroup="gml:_Feature"/>
 <!-- ============================================================ -->
 <complexType name="FeatureCollectionType">
   <annotation>
     <documentation> Concrete generic feature collection.  </documentation>
   </annotation>
   <complexContent>
     <extension base="gml:AbstractFeatureCollectionType"/>
   </complexContent>
 </complexType>
 <!-- ============================================================ -->
 <element name="boundedBy" type="gml:BoundingShapeType"/>
 <!-- ====================================================================== -->
 <complexType name="BoundingShapeType">
   <annotation>
     <documentation>Bounding shape.</documentation>
   </annotation>
   <sequence>
     <group ref="gml:boundingShape"/>
   </sequence>
 </complexType>
 <!-- ================================================================ -->
 <group name="boundingShape">
```

```
    <choice>
      <element ref="gml:Envelope"/>
      <element ref="gml:Null"/>
    </choice>
  </group>
  <!-- ============================================================ -->
  <!-- ========   global element declarations ======================================= -->
  <element name="Box" type="gml:EnvelopeType" substitutionGroup="gml:Envelope">
    <annotation>
      <appinfo>deprecated</appinfo>
      <documentation>deprecated with GML version 3.0</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <element name="EnvelopeWithTimePeriod" type="gml:EnvelopeWithTimePeriodType"
substitutionGroup="gml:Envelope"/>
  <!-- ====================================================================== -->
  <complexType name="EnvelopeWithTimePeriodType">
    <annotation>
      <documentation>Envelope that includes also a temporal extent.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:EnvelopeType">
        <sequence>
          <element ref="gml:timePosition" minOccurs="2" maxOccurs="2"/>
        </sequence>
        <attribute name="frame" type="anyURI" use="optional" default="#ISO-8601"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- ====================================================================== -->
  <!-- =============================================================== -->
  <element name="location" type="gml:LocationPropertyType" substitutionGroup="gml:_property"/>
  <!-- =============================================================== -->
  <complexType name="LocationPropertyType">
    <annotation>
      <documentation>generalised locator property  - follow gml:AssociationType pattern</documentation>
    </annotation>
    <sequence>
      <group ref="gml:locator" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <group name="locator">
    <choice>
      <element ref="gml:_Geometry"/>
      <element ref="gml:LocationKeyWord"/>
      <element ref="gml:LocationString"/>
    </choice>
  </group>
  <!-- =============================================================== -->
  <element name="LocationKeyWord" type="gml:CodeType"/>
  <element name="LocationString" type="gml:StringOrRefType"/>
  <!-- ============================================================ -->
  <element name="priorityLocation" type="gml:PriorityLocationPropertyType"
substitutionGroup="gml:location"/>
  <!-- =============================================================== -->
  <complexType name="PriorityLocationPropertyType">
    <annotation>
      <documentation>G-XML component</documentation>
```

```
    </annotation>
    <complexContent>
      <extension base="gml:LocationPropertyType">
        <attribute name="priority" type="string" use="optional"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================== -->
  <!-- common aliases for geometry properties -->
  <element name="centerOf" type="gml:PointPropertyType" substitutionGroup="gml:pointProperty"/>
  <element name="position" type="gml:PointPropertyType" substitutionGroup="gml:pointProperty"/>
  <element name="edgeOf" type="gml:CurvePropertyType" substitutionGroup="gml:curveProperty"/>
  <element name="centerLineOf" type="gml:CurvePropertyType" substitutionGroup="gml:curveProperty"/>
  <element name="extentOf" type="gml:SurfacePropertyType" substitutionGroup="gml:surfaceProperty"/>
  <element name="coverage" type="gml:SurfacePropertyType" substitutionGroup="gml:surfaceProperty">
    <annotation>
      <appinfo>deprecated</appinfo>
      <documentation>deprecated in gml 3.0</documentation>
    </annotation>
  </element>
  <!-- ============================================================== -->
</schema>
```

dynamicFeature.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:dynamicFeature:v3.00"/>
    <documentation xml:lang="en">
    Basic support for tracking moving objects and objects with changing state.
    Copyright (c) 2002 OGC, All Rights Reserved.
    </documentation>
  </annotation>
  <!-- ================================================================== -->
  <include schemaLocation="feature.xsd"/>
  <include schemaLocation="direction.xsd"/>
  <!-- ================================================================== -->
  <element name="dataSource" type="gml:StringOrRefType"/>
  <element name="status" type="gml:StringOrRefType"/>
  <!-- ================================================================== -->
  <element name="_TimeSlice" type="gml:AbstractTimeSliceType" abstract="true"
substitutionGroup="gml:_GML"/>
  <!-- ===================================== -->
  <complexType name="AbstractTimeSliceType" abstract="true">
    <annotation>
      <documentation xml:lang="en">
        A timeslice encapsulates the time-varying properties of a dynamic feature--it
        must be extended to represent a timestamped projection of a feature. The dataSource
        property describes how the temporal data was acquired.
      </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGMLType">
        <sequence>
          <element ref="gml:timeStamp"/>
          <element ref="gml:dataSource" minOccurs="0"/>
```



```
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ====================================================================== -->
<element name="MovingObjectStatus" type="gml:MovingObjectStatusType"
substitutionGroup="gml:_TimeSlice"/>
<!-- ======================================== -->
<complexType name="MovingObjectStatusType">
  <annotation>
    <documentation xml:lang="en">
        This type encapsulates various dynamic properties of moving objects
        (points, lines, regions). It is useful for dealing with features whose
        geometry or topology changes over time.    </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractTimeSliceType">
      <sequence>
        <element ref="gml:location"/>
        <element name="speed" type="gml:MeasureType" minOccurs="0"/>
        <element name="bearing" type="gml:DirectionPropertyType" minOccurs="0"/>
        <element name="acceleration" type="gml:MeasureType" minOccurs="0"/>
        <element name="elevation" type="gml:MeasureType" minOccurs="0"/>
        <element ref="gml:status" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ====================================================================== -->
<element name="history" type="gml:HistoryPropertyType"/>
<!-- ======================================== -->
<complexType name="HistoryPropertyType">
  <annotation>
    <documentation xml:lang="en">
      The history relationship associates a feature with a sequence of TimeSlice instances.
    </documentation>
  </annotation>
  <sequence>
    <element ref="gml:_TimeSlice" maxOccurs="unbounded"/>
  </sequence>
</complexType>
<!-- ====================================================================== -->
<element name="track" type="gml:TrackType" substitutionGroup="gml:history"/>
<!-- ======================================== -->
<complexType name="TrackType">
  <annotation>
    <documentation xml:lang="en">    The track of a moving object is a sequence of specialized timeslices
that indicate the status of the object.    </documentation>
  </annotation>
  <complexContent>
    <restriction base="gml:HistoryPropertyType">
      <sequence>
        <element ref="gml:MovingObjectStatus" maxOccurs="unbounded"/>
      </sequence>
    </restriction>
  </complexContent>
</complexType>
<!-- ====================================================================== -->
<group name="dynamicProperties">
  <sequence>
```

```
      <element ref="gml:timeStamp" minOccurs="0"/>
      <element ref="gml:history" minOccurs="0"/>
      <element ref="gml:dataSource" minOccurs="0"/>
    </sequence>
  </group>
  <!-- ====================================================================== -->
  <complexType name="DynamicFeatureType">
    <annotation>
      <documentation>A dynamic feature may possess a history and/or a timestamp.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <group ref="gml:dynamicProperties"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- ====================================== -->
  <complexType name="DynamicFeatureCollectionType">
    <annotation>
      <documentation>A dynamic feature collection may possess a history and/or a
timestamp.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:FeatureCollectionType">
        <group ref="gml:dynamicProperties"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- ====================================== -->
</schema>
```

observation.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-
xsd:observation:v3.00">observation.xsd</appinfo>
    <documentation>Observation schema for GML 3.0
    </documentation>
  </annotation>
  <!-- ======================================================================= -->
  <!-- ===       includes and imports ============================================ -->
  <include schemaLocation="feature.xsd"/>
  <include schemaLocation="direction.xsd"/>
  <!-- ======================================================================= -->
  <!-- =================== properties =================================== -->
  <element name="using" type="gml:FeaturePropertyType">
    <annotation>
      <documentation>This element contains or points to a description of a sensor, instrument or procedure
used for the observation</documentation>
    </annotation>
  </element>
  <!-- ======================================================================= -->
  <element name="target" type="gml:TargetPropertyType">
    <annotation>
```

```
      <documentation>This element contains or points to the specimen, region or station which is the object of
the observation</documentation>
    </annotation>
  </element>
  <!-- ====================================================================== -->
  <element name="subject" type="gml:TargetPropertyType" substitutionGroup="gml:target">
    <annotation>
      <documentation>Synonym for target - common word used for photographs</documentation>
    </annotation>
  </element>
  <!-- ====================================================================== -->
  <complexType name="TargetPropertyType">
    <annotation>
      <documentation>Container for an object representing the target or subject of an
observation.</documentation>
    </annotation>
    <choice minOccurs="0">
      <element ref="gml:_Feature"/>
      <element ref="gml:_Geometry"/>
    </choice>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ====================================================================== -->
  <element name="resultOf" type="gml:AssociationType">
    <annotation>
      <documentation>The result of the observation: an image, external object, etc</documentation>
    </annotation>
  </element>
  <!-- ====================================================================== -->
  <!-- ===================== Features ========================== -->
  <element name="Observation" type="gml:ObservationType" substitutionGroup="gml:_Feature"/>
  <!-- ============================================================ -->
  <complexType name="ObservationType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:timeStamp"/>
          <element ref="gml:using" minOccurs="0"/>
          <element ref="gml:target" minOccurs="0"/>
          <element ref="gml:resultOf"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="DirectedObservation" type="gml:DirectedObservationType"
substitutionGroup="gml:_Feature"/>
  <!-- ============================================================ -->
  <complexType name="DirectedObservationType">
    <complexContent>
      <extension base="gml:ObservationType">
        <sequence>
          <element ref="gml:direction"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
</schema>
```

valueObjects.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-
xsd:valueObjects:v3.00">valueObjects.xsd</appinfo>
    <documentation>     Copyright (c) 2002 OGC, All Rights Reserved.  GML conformant schema for Values
in which the  * scalar Value types and lists have their values recorded in simpleContent elements * complex
Value types are built recursively
    </documentation>
  </annotation>
  <!-- ====================================================================== -->
  <!-- geometry and temporal included so that _Geometry and _TimeObject can be added to Value choice
group -->
  <include schemaLocation="geometryBasic0d1d.xsd"/>
  <include schemaLocation="temporal.xsd"/>
  <!-- ====================================================================== -->
  <group name="Value">
    <choice>
      <element ref="gml:_Value"/>
      <element ref="gml:_Geometry"/>
      <element ref="gml:_TimeObject"/>
      <element ref="gml:Null"/>
      <element ref="gml:measure"/>
    </choice>
    <!--   <xs:documentation>       <xs:annotation>Utility choice group which unifies generic Values defined in
this schema document with Geometry and Temporal objects and the Measures described above, so that any
of these may be used within aggregate Values.     </xs:annotation>     </xs:documentation> -->
  </group>
  <!-- ====================================================================== -->
  <element name="_Value" abstract="true" substitutionGroup="gml:_Object">
    <annotation>
      <documentation>Abstract element which acts as the head of a substitution group which contains
_ScalarValue, _ScalarValueList and CompositeValue and (transitively) the elements in their substitution
groups.  This element may be used in an application schema as a variable, so that in an XML instance
document any member of its substitution group may occur.  </documentation>
    </annotation>
  </element>
  <!-- ====================================================================== -->
  <!-- ================== Scalar Values ========================= -->
  <element name="_ScalarValue" abstract="true" substitutionGroup="gml:_Value">
    <annotation>
      <documentation>Abstract element which acts as the head of a substitution group which contains
Boolean, Category, Count and Quantity, and (transitively) the elements in their substitution groups.  This
element may be used in an application schema as a variable, so that in an XML instance document any
member of its substitution group may occur.  </documentation>
    </annotation>
  </element>
  <!-- ====================================================================== -->
  <element name="_ScalarValueList" abstract="true" substitutionGroup="gml:_Value">
    <annotation>
      <documentation>Abstract element which acts as the head of a substitution group which contains the
compact encodings BooleanList, CategoryList, CountList and QuantityList, and (transitively) the elements in
```

```
their substitution groups.  This element may be used in an application schema as a variable, so that in an XML instance document any member of its substitution group may occur.  </documentation>
   </annotation>
  </element>
  <!-- ====================================================================== -->
  <!-- ======================= Boolean ======================= -->
  <element name="Boolean" type="boolean" substitutionGroup="gml:_ScalarValue">
   <annotation>
     <documentation>A value from two-valued logic, using the XML Schema boolean type.  An instance may take the values {true, false, 1, 0}.  </documentation>
   </annotation>
  </element>
  <element name="BooleanList" type="gml:booleanOrNullList" substitutionGroup="gml:_ScalarValueList">
   <annotation>
     <documentation>XML List based on XML Schema boolean type.  An element of this type contains a space-separated list of boolean values {0,1,true,false}</documentation>
   </annotation>
  </element>
  <!-- ====================================================================== -->
  <!-- ======================= Category ========================= -->
  <element name="Category" type="gml:CodeType" substitutionGroup="gml:_ScalarValue">
   <annotation>
     <documentation>A term representing a classification.  It has an optional XML attribute codeSpace, whose value is a URI which identifies a dictionary, codelist or authority for the term.  </documentation>
   </annotation>
  </element>
  <element name="CategoryList" type="gml:CodeOrNullListType" substitutionGroup="gml:_ScalarValueList">
   <annotation>
     <documentation>A space-separated list of terms or nulls.  A single XML attribute codeSpace may be provided, which authorises all the terms in the list.    </documentation>
   </annotation>
  </element>
  <!-- ====================================================================== -->
  <!-- ========================== Quantity ========================== -->
  <element name="Quantity" type="gml:MeasureType" substitutionGroup="gml:_ScalarValue">
   <annotation>
     <documentation>A numeric value with a scale.  The content of the element is an amount using the XML Schema type double which permits decimal or scientific notation.  An XML attribute uom ("unit of measure") is required, whose value is a URI which identifies the definition of the scale or units by which the numeric value must be multiplied.  </documentation>
   </annotation>
  </element>
  <element name="QuantityList" type="gml:MeasureOrNullListType" substitutionGroup="gml:_ScalarValueList">
   <annotation>
     <documentation>A space separated list of amounts or nulls.  The amounts use the XML Schema type double.  A single XML attribute uom ("unit of measure") is required, whose value is a URI which identifies the definition of the scale or units by which all the amounts in the list must be multiplied.  </documentation>
   </annotation>
  </element>
  <!-- ====================================================================== -->
  <!-- ======================== Count ======================== -->
  <element name="Count" type="integer" substitutionGroup="gml:_ScalarValue">
   <annotation>
     <documentation>An integer representing a frequency of occurrence. </documentation>
   </annotation>
  </element>
  <element name="CountList" type="gml:integerOrNullList" substitutionGroup="gml:_ScalarValueList">
   <annotation>
```

```
    <documentation>A space-separated list of integers or nulls.  </documentation>
  </annotation>
</element>
<!-- ====================================================================== -->
<!--              aggregate Value types                    -->
<!-- ====================================================================== -->
<!-- ==================== ValueCollection ========================= -->
<complexType name="CompositeValueType">
  <annotation>
    <documentation>Aggregate value built from other Values using the Composite pattern. It contains zero
or an arbitrary number of valueComponent elements, and zero or one valueComponents elements.  It may
be used for strongly coupled aggregates (vectors, tensors) or for arbitrary collections of
values.</documentation>
  </annotation>
  <complexContent>
    <extension base="gml:AbstractGMLType">
      <sequence>
        <element ref="gml:valueComponent" minOccurs="0" maxOccurs="unbounded"/>
        <element ref="gml:valueComponents" minOccurs="0"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="CompositeValue" type="gml:CompositeValueType" substitutionGroup="gml:_Value">
  <annotation>
    <documentation>Aggregate value built using the Composite pattern.  </documentation>
  </annotation>
</element>
<!-- ====================================================================== -->
<!-- ======================== ValueArray ========================= -->
<complexType name="ValueArrayType">
  <annotation>
    <documentation>A Value Array is used for homogeneous arrays of primitive and aggregate values.  The
member values may be scalars, composites, arrays or lists.  ValueArray has the same content model as
CompositeValue, but the member values must be homogeneous.  The element declaration contains a
Schematron constraint which expresses this restriction precisely.          Since the members are
homogeneous, the referenceSystem (uom, codeSpace) may be specified on the ValueArray itself and
implicitly inherited by all the members if desired.    Note that a_ScalarValueList is preferred for arrays of
Scalar Values since this is a more efficient encoding.  </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:CompositeValueType">
      <attributeGroup ref="gml:referenceSystem"/>
    </extension>
  </complexContent>
</complexType>
<element name="ValueArray" type="gml:ValueArrayType" substitutionGroup="gml:CompositeValue">
  <annotation>
    <appinfo>
      <sch:pattern name="Check either codeSpace or uom not both">
        <sch:rule context="gml:ValueArray">
          <sch:report test="@codeSpace and @uom">ValueArray may not carry both a reference to a
codeSpace and a uom</sch:report>
        </sch:rule>
      </sch:pattern>
      <sch:pattern name="Check components are homogeneous">
        <sch:rule context="gml:ValueArray">
          <sch:assert test="count(gml:valueComponent/*) = count(gml:valueComponent/*[name() =
name(../../gml:valueComponent[1]/*[1])])">All components of <sch:name/> must be of the same
type</sch:assert>
```

```
      <sch:assert test="count(gml:valueComponents/*) = count(gml:valueComponents/*[name() =
name(../*[1])])">All components of <sch:name/> must be of the same type</sch:assert>
     </sch:rule>
    </sch:pattern>
   </appinfo>
   <documentation>A Value Array is used for homogeneous arrays of primitive and aggregate values.
_ScalarValueList is preferred for arrays of Scalar Values since this is more efficient.  Since "choice" is not
available for attribute groups, an external constraint (e.g. Schematron) would be required to enforce the
selection of only one of these through schema validation </documentation>
  </annotation>
 </element>
 <!-- attribute group required for ValueArray -->
 <attributeGroup name="referenceSystem">
  <attribute name="codeSpace" type="anyURI" use="optional"/>
  <attribute name="uom" type="anyURI" use="optional"/>
 </attributeGroup>
 <!-- ================================================================== -->
 <!-- ===================== Typed ValueExtents =========================== -->
 <element name="QuantityExtent" type="gml:QuantityExtentType" substitutionGroup="gml:_Value">
  <annotation>
   <documentation>Utility element to store a 2-point range of numeric values. If one member is a null, then
this is a single ended interval. </documentation>
  </annotation>
 </element>
 <!-- -->
 <complexType name="QuantityExtentType">
  <annotation>
   <documentation>Restriction of list type to store a 2-point range of numeric values. If one member is a
null, then this is a single ended interval. </documentation>
  </annotation>
  <simpleContent>
   <restriction base="gml:MeasureOrNullListType">
    <length value="2"/>
   </restriction>
  </simpleContent>
 </complexType>
 <!-- ================================================================== -->
 <element name="CategoryExtent" type="gml:CategoryExtentType" substitutionGroup="gml:_Value">
  <annotation>
   <documentation>Utility element to store a 2-point range of ordinal values. If one member is a null, then
this is a single ended interval. </documentation>
  </annotation>
 </element>
 <!-- -->
 <complexType name="CategoryExtentType">
  <annotation>
   <documentation>Restriction of list type to store a 2-point range of ordinal values. If one member is a
null, then this is a single ended interval. </documentation>
  </annotation>
  <simpleContent>
   <restriction base="gml:CodeOrNullListType">
    <length value="2"/>
   </restriction>
  </simpleContent>
 </complexType>
 <!-- ================================================================== -->
 <element name="CountExtent" type="gml:CountExtentType" substitutionGroup="gml:_Value">
  <annotation>
   <documentation>Utility element to store a 2-point range of frequency values. If one member is a null,
then this is a single ended interval. </documentation>
```

```
    </annotation>
  </element>
  <!-- -->
  <simpleType name="CountExtentType">
    <annotation>
      <documentation>Restriction of list type to store a 2-point range of frequency values. If one member is a
null, then this is a single ended interval. </documentation>
    </annotation>
    <restriction base="gml:integerOrNullList">
      <length value="2"/>
    </restriction>
  </simpleType>
  <!-- ====================================================================== -->
  <!-- ===================== pieces needed for compositing ==================== -->
  <element name="valueProperty" type="gml:ValuePropertyType">
    <annotation>
      <documentation>Element which refers to, or contains, a Value</documentation>
    </annotation>
  </element>
  <!-- ====================================================================== -->
  <element name="valueComponent" type="gml:ValuePropertyType">
    <annotation>
      <documentation>Element which refers to, or contains, a Value.  This version is used in
CompositeValues.  </documentation>
    </annotation>
  </element>
  <!-- ====================================================================== -->
  <complexType name="ValuePropertyType">
    <annotation>
      <documentation>GML property which refers to, or contains, a Value</documentation>
    </annotation>
    <sequence>
      <group ref="gml:Value" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ====================================================================== -->
  <!-- ====================================================================== -->
  <element name="valueComponents" type="gml:ValueArrayPropertyType">
    <annotation>
      <documentation>Element which refers to, or contains, a set of homogeneously typed Values.
</documentation>
    </annotation>
  </element>
  <!-- ====================================================================== -->
  <complexType name="ValueArrayPropertyType">
    <annotation>
      <documentation>GML property which refers to, or contains, a set of homogeneously typed Values.
</documentation>
    </annotation>
    <sequence>
      <group ref="gml:Value" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <!-- ====================== utility typed valueProperty types ===================  -->
  <complexType name="BooleanPropertyType">
    <annotation>
      <documentation>Property whose content is a Boolean value.</documentation>
    </annotation>
    <complexContent>
```

```
      <restriction base="gml:ValuePropertyType">
        <sequence>
          <element ref="gml:Boolean" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
  <complexType name="CategoryPropertyType">
    <annotation>
      <documentation>Property whose content is a Category.</documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:ValuePropertyType">
        <sequence>
          <element ref="gml:Category" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
  <complexType name="QuantityPropertyType">
    <annotation>
      <documentation>Property whose content is a Quantity.</documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:ValuePropertyType">
        <sequence>
          <element ref="gml:Quantity" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
  <complexType name="CountPropertyType">
    <annotation>
      <documentation>Property whose content is a Count.</documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:ValuePropertyType">
        <sequence>
          <element ref="gml:Count" minOccurs="0"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ====================================================================== -->
</schema>
```

coverage.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xlink="http://www.w3.org/1999/xlink"
elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-xsd:coverage:v3.00">coverage.xsd</appinfo>
    <documentation xml:lang="en"> GML Coverage schema.
    Copyright (c) 2002 OGC, All Rights Reserved.
    </documentation>
  </annotation>
```



```
  <!-- ==============================================================
       includes and imports
       ============================================================== -->
  <include schemaLocation="feature.xsd"/>
  <include schemaLocation="valueObjects.xsd"/>
  <include schemaLocation="grids.xsd"/>
  <include schemaLocation="geometryAggregates.xsd"/>
  <!-- ==============================================================
       global types and elements
       ============================================================== -->
  <!-- ================= Abstract coverage definition ================== -->
  <!-- ============================================================ -->
  <!-- ============================================================ -->
  <element name="_Coverage" type="gml:AbstractCoverageType" abstract="true"
substitutionGroup="gml:_Feature"/>
  <!-- ============================================================ -->
  <complexType name="AbstractCoverageType" abstract="true">
    <annotation>
      <documentation>A coverage is a set of attribute values (its range) associated to position within a
bounded space (its domain). This coverage uses a single coverage function to associate an element from
the domain with an element from its range.  This definition conforms to ISO 19123.  Note that coverage is a
GML feature.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:domainSet"/>
          <element ref="gml:rangeSet"/>
          <element ref="gml:coverageFunction" minOccurs="0"/>
        </sequence>
        <attribute name="dimension" type="positiveInteger" use="optional"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="domainSet" type="gml:DomainSetType"/>
  <!-- ============================================================ -->
  <complexType name="DomainSetType">
    <annotation>
      <documentation>The spatiotemporal domain of a coverage.
  Typically
  * a geometry collection,
  * an implicit geometry (e.g. a grid),
  * an explicit or implicit collection of time instances or periods, or
  * the geometries associated with a feature collection.

N.B. Temporal domains still to be implemented.</documentation>
    </annotation>
    <choice minOccurs="0">
      <element ref="gml:_Geometry"/>
      <element ref="gml:_TimeObject"/>
    </choice>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ============================================================ -->
  <element name="rangeSet" type="gml:RangeSetType"/>
  <!-- ============================================================ -->
  <complexType name="RangeSetType">
    <choice>
      <choice maxOccurs="unbounded">
```

```
        <element ref="gml:ValueArray">
          <annotation>
            <documentation>each member _Value holds a tuple or "row" from the equivalent
table</documentation>
          </annotation>
        </element>
        <element ref="gml:_ScalarValueList">
          <annotation>
            <documentation>each list holds the complete set of one scalar component from the values - i.e. a
"column" from the equivalent table</documentation>
          </annotation>
        </element>
      </choice>
      <element ref="gml:DataBlock">
        <annotation>
          <documentation>Its tuple list holds the values as space-separated tuples each of which contains
comma-separated components, and the tuple structure is specified using the rangeParameters property.
</documentation>
        </annotation>
      </element>
      <element ref="gml:File">
        <annotation>
          <documentation>a reference to an external source for the data, together with a description of how that
external source is structured
</documentation>
        </annotation>
      </element>
    </choice>
  </complexType>
  <!-- =========================================================== -->
  <element name="coverageFunction" type="gml:CoverageFunctionType"/>
  <!-- =========================================================== -->
  <complexType name="CoverageFunctionType">
    <annotation>
      <documentation>
      The function or rule which defines the map from members of the domainSet to the range.
      More functions will be added to this list
    </documentation>
    </annotation>
    <choice>
      <element ref="gml:MappingRule"/>
      <element ref="gml:GridFunction"/>
    </choice>
  </complexType>
  <!-- =========================================================== -->
  <!-- ============== Components for encoding the rangeSet ============= -->
  <!-- =========================================================== -->
  <element name="DataBlock" type="gml:DataBlockType"/>
  <!-- =========================================================== -->
  <complexType name="DataBlockType">
    <sequence>
      <element ref="gml:rangeParameters"/>
      <element ref="gml:tupleList"/>
    </sequence>
  </complexType>
  <!-- =========================================================== -->
  <element name="tupleList" type="gml:CoordinatesType"/>
  <!-- =========================================================== -->
  <element name="File" type="gml:FileType"/>
  <!-- =========================================================== -->
```



```
  <complexType name="FileType">
    <sequence>
      <element ref="gml:rangeParameters"/>
      <element name="fileName" type="anyURI"/>
      <element name="fileStructure" type="gml:FileValueModelType"/>
      <element name="mimeType" type="anyURI" minOccurs="0"/>
      <element name="compression" type="anyURI" minOccurs="0"/>
    </sequence>
  </complexType>
  <!-- ============================================================ -->
  <simpleType name="FileValueModelType">
    <annotation>
      <documentation>List of codes that identifies the file structure model for records stored in
files.</documentation>
    </annotation>
    <restriction base="string">
      <enumeration value="Record Interleaved"/>
    </restriction>
  </simpleType>
  <!-- ============================================================ -->
  <element name="rangeParameters" type="gml:RangeParametersType"/>
  <!-- ============================================================ -->
  <complexType name="RangeParametersType">
    <annotation>
      <documentation>
      Metadata about the rangeSet.  Definition of record structure.
      This is required if the rangeSet is encoded in a DataBlock.
      We use a gml:_Value with empty values as a map of the composite value structure.</documentation>
    </annotation>
    <sequence minOccurs="0">
      <element ref="gml:_Value"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ============================================================ -->
  <!-- ============= Components for coverageFunctions ================ -->
  <!-- ============================================================ -->
  <element name="MappingRule" type="gml:StringOrRefType">
    <annotation>
      <documentation>Description of a rule for associating members from the domainSet with members of the
rangeSet.  </documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <element name="GridFunction" type="gml:GridFunctionType"/>
  <!-- ============================================================ -->
  <complexType name="GridFunctionType">
    <annotation>
      <documentation>Defines how values in the domain are mapped to the range set. The start point and the
sequencing rule are specified here.</documentation>
    </annotation>
    <sequence>
      <element name="sequenceRule" type="gml:SequenceRuleType" minOccurs="0">
        <annotation>
          <documentation>If absent, the implied value is "Linear".</documentation>
        </annotation>
      </element>
      <element name="startPoint" type="gml:integerList" minOccurs="0">
        <annotation>
```

```
        <documentation>Index position of the first grid post.  If absent the startPoint is equal to the value of
gridEnvelope::low from the grid definition.  </documentation>
      </annotation>
    </element>
  </sequence>
 </complexType>
 <!-- ============================================================= -->
 <element name="IndexMap" type="gml:IndexMapType" substitutionGroup="gml:GridFunction"/>
 <!-- ============================================================= -->
 <complexType name="IndexMapType">
  <annotation>
    <documentation>Exends GridFunctionType with a lookUpTable.  This contains a list of indexes of
members within the rangeSet corresponding with the members of the domainSet.  The domainSet is
traversed in list order if it is enumerated explicitly, or in the order specified by a SequenceRule if the domain
is an implicit set.    The length of the lookUpTable corresponds with the length of the subset of the
domainSet for which the coverage is defined.  </documentation>
  </annotation>
  <complexContent>
    <extension base="gml:GridFunctionType">
      <sequence>
        <element name="lookUpTable" type="gml:integerList"/>
      </sequence>
    </extension>
  </complexContent>
 </complexType>
 <!-- ============================================================= -->
 <complexType name="SequenceRuleType">
  <simpleContent>
    <extension base="gml:SequenceRuleNames">
      <attribute name="order" type="gml:IncrementOrder" use="optional"/>
    </extension>
  </simpleContent>
 </complexType>
 <!-- ============================================================= -->
 <simpleType name="SequenceRuleNames">
  <annotation>
    <documentation>List of codes (adopted from ISO 19123 Annex C) that identifies the rule for traversing a
grid to correspond with the sequence of members of the rangeSet.</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="Linear"/>
    <enumeration value="Boustrophedonic"/>
    <enumeration value="Cantor-diagonal"/>
    <enumeration value="Spiral"/>
    <enumeration value="Morton"/>
    <enumeration value="Hilbert"/>
  </restriction>
 </simpleType>
 <!-- ============================================================= -->
 <simpleType name="IncrementOrder">
  <annotation>
    <documentation>The enumeration value here indicates the incrementation order  to be used on the first
2 axes, i.e. "+x-y" means that the points on the first axis are to be traversed from lowest to highest and  the
points on the second axis are to be traversed from highest to lowest. The points on all other axes (if any)
beyond the first 2 are assumed to increment from lowest to highest.</documentation>
  </annotation>
  <restriction base="string">
    <enumeration value="+x+y"/>
    <enumeration value="+y+x"/>
    <enumeration value="+x-y"/>
```

```
      <enumeration value="-x-y"/>
    </restriction>
  </simpleType>
  <!-- ============================================================ -->
  <!-- == Specialised Coverage types - typed by the structure of the domain set == -->
  <!-- ============================================================ -->
  <element name="MultiPointCoverage" type="gml:MultiPointCoverageType"
substitutionGroup="gml:_Coverage"/>
  <!-- ============================================================ -->
  <complexType name="MultiPointCoverageType">
    <annotation>
      <documentation>A discrete coverage type whose domain is defined by a collection of point
            </documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:AbstractCoverageType">
        <sequence>
          <element ref="gml:multiPointDomain"/>
          <element ref="gml:rangeSet"/>
          <element ref="gml:coverageFunction" minOccurs="0"/>
        </sequence>
        <attribute name="dimension" type="positiveInteger" use="required"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="multiPointDomain" type="gml:MultiPointDomainType"
substitutionGroup="gml:domainSet"/>
  <!-- ============================================================ -->
  <complexType name="MultiPointDomainType">
    <complexContent>
      <restriction base="gml:DomainSetType">
        <sequence minOccurs="0">
          <element ref="gml:MultiPoint"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="MultiSurfaceCoverage" type="gml:MultiSurfaceCoverageType"
substitutionGroup="gml:_Coverage"/>
  <!-- ============================================================ -->
  <complexType name="MultiSurfaceCoverageType">
    <annotation>
      <documentation>A discrete coverage type whose domain is defined by a collection of surface patches
(includes polygons, triangles, rectangles, etc).
            </documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:AbstractCoverageType">
        <sequence>
          <element ref="gml:multiSurfaceDomain"/>
          <element ref="gml:rangeSet"/>
          <element ref="gml:coverageFunction" minOccurs="0"/>
        </sequence>
        <attribute name="dimension" type="positiveInteger" use="required"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
```

```
  <element name="multiSurfaceDomain" type="gml:MultiSurfaceDomainType"
substitutionGroup="gml:domainSet"/>
  <!-- ============================================================ -->
  <complexType name="MultiSurfaceDomainType">
    <complexContent>
      <restriction base="gml:DomainSetType">
        <sequence minOccurs="0">
          <element ref="gml:MultiSurface"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="GridCoverage" type="gml:GridCoverageType" substitutionGroup="gml:_Coverage"/>
  <!-- ============================================================ -->
  <complexType name="GridCoverageType">
    <complexContent>
      <restriction base="gml:AbstractCoverageType">
        <sequence>
          <element ref="gml:gridDomain"/>
          <element ref="gml:rangeSet"/>
          <element ref="gml:coverageFunction" minOccurs="0"/>
        </sequence>
        <attribute name="dimension" type="positiveInteger" use="required"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="gridDomain" type="gml:GridDomainType" substitutionGroup="gml:domainSet"/>
  <!-- ============================================================ -->
  <complexType name="GridDomainType">
    <complexContent>
      <restriction base="gml:DomainSetType">
        <choice minOccurs="0">
          <element ref="gml:Grid"/>
        </choice>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="RectifiedGridCoverage" type="gml:RectifiedGridCoverageType"
substitutionGroup="gml:_Coverage"/>
  <!-- ============================================================ -->
  <complexType name="RectifiedGridCoverageType">
    <complexContent>
      <restriction base="gml:AbstractCoverageType">
        <sequence>
          <element ref="gml:rectifiedGridDomain"/>
          <element ref="gml:rangeSet"/>
          <element ref="gml:coverageFunction" minOccurs="0"/>
        </sequence>
        <attribute name="dimension" type="positiveInteger" use="required"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="rectifiedGridDomain" type="gml:RectifiedGridDomainType"
substitutionGroup="gml:domainSet"/>
  <!-- ============================================================ -->
```

```
  <complexType name="RectifiedGridDomainType">
    <complexContent>
      <restriction base="gml:DomainSetType">
        <choice minOccurs="0">
          <element ref="gml:RectifiedGrid"/>
        </choice>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
</schema>
```

defaultStyle.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:gml="http://www.opengis.net/gml" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-defaultStyle:v3.00">defaultStyle.xsd</appinfo>
    <documentation>
      <name>defaultStyle.xsd</name>
      <version>3.0</version>
      <scope/>
      <description>Default Style schema for GML 3.0</description>
      <copyright>Copyright (c) 2001-2002 OGC, All Rights Reserved.</copyright>
      <conformance>reference to ISO Specifications</conformance>
    </documentation>
  </annotation>
  <!-- ============================================================
       includes and imports
      ============================================================ -->
  <include schemaLocation="measures.xsd"/>
  <import namespace="http://www.w3.org/2001/SMIL20/" schemaLocation="../smil/smil20.xsd"/>
  <!-- ============================================================
      the property
      ============================================================ -->
  <element name="defaultStyle" type="gml:DefaultStylePropertyType" substitutionGroup="gml:_property">
    <annotation>
      <documentation>Top-level property. Used in application schemas to "attach" the styling information to
GML data. The link between the data and the style should be established through this property
only.</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <complexType name="DefaultStylePropertyType">
    <annotation>
      <documentation>[complexType of] Top-level property. Used in application schemas to "attach" the
styling information to GML data. The link between the data and the style should be established through this
property only.</documentation>
    </annotation>
    <sequence>
      <element ref="gml:_Style" minOccurs="0"/>
    </sequence>
    <attribute name="about" type="anyURI" use="optional"/>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
```

```xml
  <!-- ============================================================
      the style
     ============================================================== -->
  <element name="_Style" type="gml:AbstractStyleType" abstract="true" substitutionGroup="gml:_GML">
    <annotation>
      <documentation>The value of the top-level property. It is an abstract element. Used as the head element
of the substitution group for extensibility purposes.</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <complexType name="AbstractStyleType" abstract="true">
    <annotation>
      <documentation>[complexType of] The value of the top-level property. It is an abstract element. Used as
the head element of the substitution group for extensibility purposes.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGMLType"/>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="Style" type="gml:StyleType" substitutionGroup="gml:_Style">
    <annotation>
      <documentation>Predefined concrete value of the top-level property. Encapsulates all other styling
information.</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <complexType name="StyleType">
    <annotation>
      <documentation>[complexType of] Predefined concrete value of the top-level property. Encapsulates all
other styling information.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractStyleType">
        <sequence>
          <element ref="gml:FeatureStyle" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:GraphStyle" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================
      base style descriptor type (for geometry, topology, label, graph)
     ============================================================== -->
  <complexType name="BaseStyleDescriptorType">
    <annotation>
      <documentation>Base complex type for geometry, topology, label and graph styles.</documentation>
    </annotation>
    <sequence>
      <element name="spatialResolution" type="gml:ScaleType" minOccurs="0"/>
      <element name="styleVariation" type="gml:StyleVariationType" minOccurs="0"
maxOccurs="unbounded"/>
      <element ref="smil20:animate" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="smil20:animateMotion" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="smil20:animateColor" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="smil20:set" minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
  </complexType>
  <!-- ============================================================
      style descriptors
```



```
    ============================================================== -->
  <element name="FeatureStyle" type="gml:FeatureStyleType">
    <annotation>
      <documentation>The style descriptor for features.</documentation>
    </annotation>
  </element>
  <!-- ============================================================== -->
  <complexType name="FeatureStyleType">
    <annotation>
      <documentation>[complexType of] The style descriptor for features.</documentation>
    </annotation>
    <sequence>
      <element name="featureConstraint" type="string" minOccurs="0"/>
      <element ref="gml:GeometryStyle" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="gml:TopologyStyle" minOccurs="0" maxOccurs="unbounded"/>
      <element ref="gml:LabelStyle" minOccurs="0"/>
    </sequence>
    <attribute name="featureType" type="string" use="optional"/>
    <attribute name="baseType" type="string" use="optional"/>
  </complexType>
  <!-- ============================================================== -->
  <element name="GeometryStyle" type="gml:GeometryStyleType">
    <annotation>
      <documentation>The style descriptor for geometries of a feature.</documentation>
    </annotation>
  </element>
  <!-- ============================================================== -->
  <complexType name="GeometryStyleType">
    <annotation>
      <documentation>[complexType of] The style descriptor for geometries of a feature.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:BaseStyleDescriptorType">
        <sequence>
          <choice>
            <element ref="gml:symbol"/>
            <element name="style" type="string"/>
          </choice>
          <element ref="gml:LabelStyle" minOccurs="0"/>
        </sequence>
        <attribute name="geometryProperty" type="string"/>
        <attribute name="geometryType" type="string"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================== -->
  <element name="TopologyStyle" type="gml:TopologyStyleType">
    <annotation>
      <documentation>The style descriptor for topologies of a feature. Describes individual topology elements
styles.</documentation>
    </annotation>
  </element>
  <!-- ============================================================== -->
  <complexType name="TopologyStyleType">
    <annotation>
      <documentation>[complexType of] The style descriptor for topologies of a feature. Describes individual
topology elements styles.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:BaseStyleDescriptorType">
```

```
        <sequence>
          <choice>
            <element ref="gml:symbol"/>
            <element name="style" type="string"/>
          </choice>
          <element ref="gml:LabelStyle" minOccurs="0"/>
        </sequence>
        <attribute name="topologyProperty" type="string"/>
        <attribute name="topologyType" type="string"/>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="LabelStyle" type="gml:LabelStyleType">
    <annotation>
      <documentation>The style descriptor for labels of a feature, geometry or topology.</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <complexType name="LabelStyleType">
    <annotation>
      <documentation>[complexType of] The style descriptor for labels of a feature, geometry or
topology.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:BaseStyleDescriptorType">
        <sequence>
          <element name="style" type="string"/>
          <element name="label" type="gml:LabelType"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="GraphStyle" type="gml:GraphStyleType">
    <annotation>
      <documentation>The style descriptor for a graph consisting of a number of features. Describes graph-
specific style attributes.</documentation>
    </annotation>
  </element>
  <!-- ============================================================ -->
  <complexType name="GraphStyleType">
    <annotation>
      <documentation>[complexType of] The style descriptor for a graph consisting of a number of features.
Describes graph-specific style attributes.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:BaseStyleDescriptorType">
        <sequence>
          <element name="planar" type="boolean" minOccurs="0"/>
          <element name="directed" type="boolean" minOccurs="0"/>
          <element name="grid" type="boolean" minOccurs="0"/>
          <element name="minDistance" type="double" minOccurs="0"/>
          <element name="minAngle" type="double" minOccurs="0"/>
          <element name="graphType" type="gml:GraphTypeType" minOccurs="0"/>
          <element name="drawingType" type="gml:DrawingTypeType" minOccurs="0"/>
          <element name="lineType" type="gml:LineTypeType" minOccurs="0"/>
          <element name="aestheticCriteria" type="gml:AesheticCriteriaType" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
```



```
    </extension>
   </complexContent>
  </complexType>
  <!-- ============================================================== -->
  <element name="symbol" type="gml:SymbolType">
   <annotation>
    <documentation>The symbol property. Extends the gml:AssociationType to allow for remote referencing
of symbols.</documentation>
   </annotation>
  </element>
  <!-- ============================================================== -->
  <complexType name="SymbolType">
   <annotation>
    <documentation>[complexType of] The symbol property. Extends the gml:AssociationType to allow for
remote referencing of symbols.</documentation>
   </annotation>
   <complexContent>
    <extension base="gml:AssociationType">
     <attribute name="symbolType" type="gml:SymbolTypeEnumeration" use="required"/>
     <attribute ref="gml:transform" use="optional"/>
    </extension>
   </complexContent>
  </complexType>
  <!-- ============================================================== -->
  <simpleType name="SymbolTypeEnumeration">
   <annotation>
    <documentation>Used to specify the type of the symbol used.</documentation>
   </annotation>
   <restriction base="string">
    <enumeration value="svg"/>
    <enumeration value="other"/>
   </restriction>
  </simpleType>
  <!-- ============================================================== -->
  <complexType name="LabelType" mixed="true">
   <annotation>
    <documentation>Label is mixed -- composed of text and XPath expressions used to extract the useful
information from the feature.</documentation>
   </annotation>
   <sequence>
    <element name="LabelExpression" type="string" minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
   <attribute ref="gml:transform" use="optional"/>
  </complexType>
  <!-- ============================================================== -->
  <attribute name="transform" type="string">
   <annotation>
    <documentation>Defines the geometric transformation of entities. There is no particular grammar
defined for this value.</documentation>
   </annotation>
  </attribute>
  <!-- ============================================================== -->
  <complexType name="StyleVariationType">
   <annotation>
    <documentation>Used to vary individual graphic parameters and attributes of the style, symbol or
text.</documentation>
   </annotation>
   <simpleContent>
    <extension base="string">
     <attribute name="styleProperty" type="string" use="required"/>
```

```xml
      <attribute name="featurePropertyRange" type="string" use="optional"/>
    </extension>
   </simpleContent>
  </complexType>
  <!-- ============================================================
      simple types
    ============================================================ -->
  <simpleType name="GraphTypeType">
   <annotation>
    <documentation>Graph-specific styling property.</documentation>
   </annotation>
   <restriction base="string">
    <enumeration value="TREE"/>
    <enumeration value="BICONNECTED"/>
   </restriction>
  </simpleType>
  <!-- ============================================================ -->
  <simpleType name="DrawingTypeType">
   <annotation>
    <documentation>Graph-specific styling property.</documentation>
   </annotation>
   <restriction base="string">
    <enumeration value="POLYLINE"/>
    <enumeration value="ORTHOGONAL"/>
   </restriction>
  </simpleType>
  <!-- ============================================================ -->
  <simpleType name="LineTypeType">
   <annotation>
    <documentation>Graph-specific styling property.</documentation>
   </annotation>
   <restriction base="string">
    <enumeration value="STRAIGHT"/>
    <enumeration value="BENT"/>
   </restriction>
  </simpleType>
  <!-- ============================================================ -->
  <simpleType name="AesheticCriteriaType">
   <annotation>
    <documentation>Graph-specific styling property.</documentation>
   </annotation>
   <restriction base="string">
    <enumeration value="MIN_CROSSINGS"/>
    <enumeration value="MIN_AREA"/>
    <enumeration value="MIN_BENDS"/>
    <enumeration value="MAX_BENDS"/>
    <enumeration value="UNIFORM_BENDS"/>
    <enumeration value="MIN_SLOPES"/>
    <enumeration value="MIN_EDGE_LENGTH"/>
    <enumeration value="MAX_EDGE_LENGTH"/>
    <enumeration value="UNIFORM_EDGE_LENGTH"/>
    <enumeration value="MAX_ANGULAR_RESOLUTION"/>
    <enumeration value="MIN_ASPECT_RATIO"/>
    <enumeration value="MAX_SYMMETRIES"/>
   </restriction>
  </simpleType>
</schema>
```

Annex D
(informative)

CRS Schemas

dataQuality.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified" xml:lang="en" version="3.00">
  <xsd:annotation>
    <xsd:appinfo source="urn:opengis:specification:gml:schema-dataQuality:v3.00"/>
    <xsd:documentation>
      <name>dataQuality.xsd</name>
      <version>3.0</version>
      <scope>How to encode positional data quality information. </scope>
      <description>Builds on measures.xsd to encode the data needed to describe the positional accuracy of
coordinate operations. GML 3.0 candidate schema, primary editor: Arliss Whiteside. Last updated
2002/11/21. </description>
      <copyright>Copyright (c) 2001-2002 OpenGIS, All Rights Reserved.</copyright>
      <conformance>This schema encodes the Data Quality (DQ) package of the extended UML Model for
OGC Abstract Specification Topic 2: Spatial Referencing by Coordinates. That UML model is adapted from
ISO 19111 - Spatial referencing by coordinates, as described in Annex B of Topic 2. </conformance>
    </xsd:documentation>
  </xsd:annotation>
  <!-- ==============================================================
       includes and imports
       ============================================================== -->
  <xsd:include schemaLocation="units.xsd"/>
  <!-- ==============================================================
       elements and types
       ============================================================== -->
  <xsd:element name="_PositionalAccuracy" type="gml:AbstractPositionalAccuracyType" abstract="true"/>
  <!-- ============================================================== -->
  <xsd:complexType name="AbstractPositionalAccuracyType" abstract="true">
    <xsd:annotation>
      <xsd:documentation>Position error estimate (or accuracy) data. </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element name="measureDescription" type="string" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>A description of the position accuracy parameter(s) provided.
</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
    </xsd:sequence>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="AbsoluteExternalPositionalAccuracy"
type="gml:AbsoluteExternalPositionalAccuracyType" substitutionGroup="gml:_PositionalAccuracy"/>
  <!-- ============================================================== -->
  <xsd:complexType name="AbsoluteExternalPositionalAccuracyType">
    <xsd:annotation>
```

```
      <xsd:documentation>Closeness of reported coordinate values to values accepted as or being true.
</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractPositionalAccuracyType">
        <xsd:sequence>
          <xsd:element name="result" type="gml:MeasureType" maxOccurs="2">
            <xsd:annotation>
              <xsd:documentation>Quantitative result determined by evaluation procedure used.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="RelativeInternalPositionalAccuracy"
type="gml:RelativeInternalPositionalAccuracyType" substitutionGroup="gml:_PositionalAccuracy"/>
  <!-- ============================================================== -->
  <xsd:complexType name="RelativeInternalPositionalAccuracyType">
    <xsd:annotation>
      <xsd:documentation>Closeness of the relative positions of two or more positions to their respective
relative positions accepted as or being true. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractPositionalAccuracyType">
        <xsd:sequence>
          <xsd:element name="result" type="gml:MeasureType" maxOccurs="2">
            <xsd:annotation>
              <xsd:documentation>Quantitative result determined by evaluation procedure used.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="CovarianceMatrix" type="gml:CovarianceMatrixType"
substitutionGroup="gml:_PositionalAccuracy"/>
  <!-- ============================================================== -->
  <xsd:complexType name="CovarianceMatrixType">
    <xsd:annotation>
      <xsd:documentation>Error estimate covariance matrix. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractPositionalAccuracyType">
        <xsd:sequence>
          <xsd:element ref="gml:unitOfMeasure" maxOccurs="unbounded">
            <xsd:annotation>
              <xsd:documentation>Ordered sequence of units of measure, corresponding to the row and column
index numbers of the covariance matrix, starting with row and column 1 and ending with row/column N.
Each unit of measure is for the ordinate reflected in the relevant row and column of the covariance matrix.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="includesElement" type="gml:CovarianceElementType"
maxOccurs="unbounded">
            <xsd:annotation>
```



```
            <xsd:documentation>Unordered sequence of elements in this covariance matrix. Because the covariance matrix is symmetrical, only the elements in upper or lower diagonal part (including the main diagonal) of the matrix need to be specified. Any zero valued covariance elements can be omitted.</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:complexType name="CovarianceElementType">
  <xsd:annotation>
    <xsd:documentation>An element of a covariance matrix.</xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="rowIndex" type="positiveInteger">
      <xsd:annotation>
        <xsd:documentation>Row number of this covariance element value. </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="columnIndex" type="positiveInteger">
      <xsd:annotation>
        <xsd:documentation>Column number of this covariance element value. </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
    <xsd:element name="covariance" type="double">
      <xsd:annotation>
        <xsd:documentation>Value of covariance element. </xsd:documentation>
      </xsd:annotation>
    </xsd:element>
  </xsd:sequence>
 </xsd:complexType>
</xsd:schema>
```

referenceSystems.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:gml="http://www.opengis.net/gml" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="3.00" xml:lang="en">
 <annotation>
  <appinfo source="urn:opengis:specification:gml:schema-referenceSystems:v3.00"/>
  <documentation>
    <name>referenceSystems.xsd</name>
    <version>3.0</version>
    <scope>How to encode reference system definitions. </scope>
    <description>Builds on gmlBase.xsd to encode the data needed to define reference systems. GML 3.0 candidate schema, primary editor: Arliss Whiteside. Last updated 2002/11/21. </description>
    <copyright>Copyright (c) 2001-2002 OpenGIS, All Rights Reserved.</copyright>
    <conformance>This schema encodes the Reference System (RS_) package of the extended UML Model for OGC Abstract Specification Topic 2: Spatial Referencing by Coordinates. That UML model is adapted from ISO 19111 - Spatial referencing by coordinates, as described in Annex B of Topic 2. The CS_CRS class is also encoded here, to eliminate the (circular) references from coordonateOperations.xsd to coordinateReferenceSystems.xsd. A modified version of the EX_Extent (DataType) class is also currently encoded here, using GML 3 schema types. (A more extensive version of the EX_Extent package might be XML encoded in the future, probably in a separate extent.xsd schema.) </conformance>
  </documentation>
```

```
  </annotation>
  <!-- ================================================================        includes and
imports      ================================================================= -->
  <include schemaLocation="geometryBasic2d.xsd"/>
  <include schemaLocation="temporal.xsd"/>
  <!-- ================================================================        elements and
types    ================================================================= -->
  <element name="_CRSObject" type="gml:AbstractCRSObjectType" abstract="true"
substitutionGroup="gml:_Object"/>
  <!-- ============================================================= -->
  <complexType name="AbstractCRSObjectType" abstract="true">
    <annotation>
      <documentation>Basic encoding for objects in the CRS model, simplifying the AbstracGMLType as
needed for CRS objects. </documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:AbstractGMLType">
        <sequence>
          <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute ref="gml:id" use="optional"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="_ReferenceSystem" type="gml:AbstractReferenceSystemType" abstract="true"
substitutionGroup="gml:_CRSObject"/>
  <!-- ============================================================= -->
  <complexType name="AbstractReferenceSystemType" abstract="true">
    <annotation>
      <documentation>Description of a spatial and/or temporal reference system used by a
dataset.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCRSObjectType">
        <sequence>
          <element name="crsID" type="gml:ExtendedIdentifierType">
            <annotation>
              <documentation>Identification of this Reference System. </documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="referenceSystemRef" type="gml:ReferenceSystemRefType"/>
  <!-- ============================================================= -->
  <complexType name="ReferenceSystemRefType">
    <annotation>
      <documentation>Association to a Reference System, either referencing or containing the definition of
that Reference System. </documentation>
    </annotation>
    <sequence>
      <element ref="gml:_ReferenceSystem" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ============================================================= -->
```

```
  <element name="_CRS" type="gml:AbstractCRSType" abstract="true"
substitutionGroup="gml:_ReferenceSystem"/>
  <!-- ====================================================================== -->
  <complexType name="AbstractCRSType" abstract="true">
    <annotation>
      <documentation>Abstract coordinate reference system, usually defined by a coordinate system and a
datum. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractReferenceSystemType"/>
    </complexContent>
  </complexType>
  <!-- ====================================================================== -->
  <element name="crsRef" type="gml:CRSRefType" substitutionGroup="gml:referenceSystemRef"/>
  <!-- ====================================================================== -->
  <complexType name="CRSRefType">
    <annotation>
      <documentation>Association to a CRS abstract coordinate reference system, either referencing or
containing the definition of that CRS. </documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:ReferenceSystemRefType">
        <sequence>
          <element ref="gml:_CRS" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ====================================================================== -->
  <!-- ====================================================================== -->
  <complexType name="IdentifierType">
    <annotation>
      <documentation>Identification of a reference system object. All of the attributes are optional, but there is
a constraint that one or both of the "code" and "name" attributes must be included. </documentation>
    </annotation>
    <sequence>
      <element name="code" type="string" minOccurs="0">
        <annotation>
          <documentation>Identifier code or name, optionally from a controlled list or pattern defined by a code
space. </documentation>
        </annotation>
      </element>
      <element name="codeSpace" type="string" minOccurs="0">
        <annotation>
          <documentation>Identifier of a code space within which one or more codes are defined. This attribute
is optionally included only when the "code" attribute is included. This code space is often defined by some
authority organization, where one organization may define multiple code spaces. The range and format of
each Code Space identifier is defined by that code space authority. </documentation>
        </annotation>
      </element>
      <element name="version" type="string" minOccurs="0">
        <annotation>
          <documentation>Identifier of the version of the associated codeSpace or code, as specified by the
codeSpace or code authority. This attribute is optionally included only when the "code" attribute is included.
When appropriate, the version is identified by the effective date, coded using ISO 8601 date format.
</documentation>
        </annotation>
      </element>
      <element name="name" type="string" minOccurs="0">
```



```xml
        <annotation>
          <documentation>The name by which this object is identified. </documentation>
        </annotation>
      </element>
      <element name="alias" type="gml:AliasType" minOccurs="0" maxOccurs="unbounded">
        <annotation>
          <documentation>Unordered list of aliases by which this object is known. </documentation>
        </annotation>
      </element>
      <element name="remarks" type="string" minOccurs="0">
        <annotation>
          <documentation>Comments on or information about this object, including data source information.
</documentation>
        </annotation>
      </element>
    </sequence>
  </complexType>
  <!-- ============================================================ -->
  <complexType name="AliasType">
    <annotation>
      <documentation>Alternative identifier or name by which this object is known. </documentation>
    </annotation>
    <sequence>
      <element name="aliasName" type="string">
        <annotation>
          <documentation>An alias of this object, as defined in the alias name space. </documentation>
        </annotation>
      </element>
      <element name="aliasNameSpace" type="string">
        <annotation>
          <documentation>Identifier of the namespace that this alias name is defined in; for example, "ISO 2-
char country code" or "EPSG abbreviation". </documentation>
        </annotation>
      </element>
      <element name="aliasRemarks" type="string" minOccurs="0">
        <annotation>
          <documentation>Remarks applying to this alias. </documentation>
        </annotation>
      </element>
    </sequence>
  </complexType>
  <!-- ============================================================ -->
  <complexType name="ExtendedIdentifierType">
    <annotation>
      <documentation>Extended identification and description of a reference system object. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:IdentifierType">
        <sequence>
          <element name="validArea" type="gml:ExtentType" minOccurs="0">
            <annotation>
              <documentation>Area or region in which this object is valid. </documentation>
            </annotation>
          </element>
          <element name="scope" type="string" minOccurs="0">
            <annotation>
              <documentation>Description of domain of usage, or limitations of usage, for which this object is
valid. </documentation>
            </annotation>
          </element>
```



```
      </sequence>
    </extension>
  </complexContent>
 </complexType>
 <!-- =========================================================== -->
 <complexType name="ExtentType">
  <annotation>
    <documentation>Information about the spatial, vertical, and/or temporal extent of a reference system
object. Constraints: At least one of the elements "description", "boundingBox", "boundingPolygon",
"verticalExtent", and temporalExtent" must be included, but more that one can be included when
appropriate. Furthermore, more than one "boundingBox", "boundingPolygon", "verticalExtent", or
temporalExtent" element can be included, with more than one meaning the union of the individual domains.
</documentation>
  </annotation>
  <sequence>
    <element ref="gml:description" minOccurs="0">
      <annotation>
        <documentation>Description of spatial and/or temporal extent of this object. </documentation>
      </annotation>
    </element>
    <choice minOccurs="0" maxOccurs="unbounded">
      <annotation>
        <documentation>Geographic domain of this reference system object. </documentation>
      </annotation>
      <element name="boundingBox" type="gml:EnvelopeType">
        <annotation>
          <documentation>Bounding box (or envelope) spatial domain of this object. </documentation>
        </annotation>
      </element>
      <element name="boundingPolygon" type="gml:PolygonType">
        <annotation>
          <documentation>Bounding polygon horizontal spatial domain of this object. </documentation>
        </annotation>
      </element>
    </choice>
    <element name="verticalExtent" type="gml:EnvelopeType" minOccurs="0" maxOccurs="unbounded">
      <annotation>
        <documentation>Vertical spatial domain of this object. </documentation>
      </annotation>
    </element>
    <element name="temporalExtent" type="gml:TimePeriodType" minOccurs="0"
maxOccurs="unbounded">
      <annotation>
        <documentation>Time period domain of this object. </documentation>
      </annotation>
    </element>
  </sequence>
 </complexType>
</schema>
```

datums.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified" xml:lang="en" version="3.00">
 <xsd:annotation>
  <xsd:appinfo source="urn:opengis:specification:gml:schema-datums:v3.00"/>
```



```
    <xsd:documentation>
      <name>datums.xsd</name>
      <version>3.0</version>
      <scope>How to encode datum definitions. </scope>
      <description>Builds on referenceSystems.xsd to encode the data needed to define datums, including the
specific subtypes of datums. GML 3.0 candidate schema, primary editor: Arliss Whiteside. Last updated
2002/11/21. </description>
      <copyright>Copyright (c) 2001-2002 OpenGIS, All Rights Reserved.</copyright>
      <conformance>This schema encodes the Datum (CD_) package of the extended UML Model for OGC
Abstract Specification Topic 2: Spatial Referencing by Coordinates. That UML model is adapted from ISO
19111 - Spatial referencing by coordinates, as described in Annex B of Topic 2. </conformance>
    </xsd:documentation>
  </xsd:annotation>
  <!-- ==============================================================
       includes and imports
      ============================================================== -->
  <xsd:include schemaLocation="referenceSystems.xsd"/>
  <!-- ==============================================================
       elements and types
      ============================================================== -->
  <xsd:element name="_Datum" type="gml:AbstractDatumType" abstract="true"
substitutionGroup="gml:_CRSObject"/>
  <!-- ============================================================== -->
  <xsd:complexType name="AbstractDatumType" abstract="true">
    <xsd:annotation>
      <xsd:documentation>A datum specifies the relationship of a coordinate system to the earth, thus
creating a coordinate reference system. A datum uses a parameter or set of parameters that determine the
location of the origin, the orientation, and the scale of a coordinate reference system. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCRSObjectType">
        <xsd:sequence>
          <xsd:element name="datumID" type="gml:ExtendedIdentifierType">
            <xsd:annotation>
              <xsd:documentation>Identification of this datum. </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="anchorPoint" type="string" minOccurs="0">
            <xsd:annotation>
              <xsd:documentation>Description, possibly including coordinates, of the point or points used to
anchor the datum to the Earth. Also known as the "origin", especially for Engineering and Image Datums.
- For a geodetic datum, this point is also known as the fundamental point, which is traditionally the point
where the relationship between geoid and ellipsoid is defined. In some cases, the "fundamental point" may
consist of a number of points; and the parameters defining the geoid/ellipsoid relationship have then been
averaged for a number of points and adopted as the datum definition.
- For an engineering datum, the anchor point may be a physical point, or it may be a point with defined
coordinates in another CRS.
- For an image datum, the anchor point is usually either the centre of the image or the corner of the image.
- For a temporal datum, this attribute is not defined. Instead of the anchor point, a temporal datum carries a
separate time origin of type DateTime. </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="realizationEpoch" type="date" minOccurs="0">
            <xsd:annotation>
              <xsd:documentation>The time for which this datum definition is valid. This time may be precise
(e.g. 1997.0 for IRTF97) or merely a year (e.g. 1983 for NAD83). In the latter case, the epoch usually refers
to the year in which a major recalculation of the geodetic control network, underlying the datum, was
executed or initiated. </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
```

```
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="datumRef" type="gml:DatumRefType"/>
 <!-- ============================================================ -->
 <xsd:complexType name="DatumRefType">
  <xsd:annotation>
    <xsd:documentation>Association to a Datum, either referencing or containing the definition of that
Datum. </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element ref="gml:_Datum" minOccurs="0"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="EngineeringDatum" type="gml:EngineeringDatumType"
substitutionGroup="gml:_Datum"/>
 <!-- ============================================================ -->
 <xsd:complexType name="EngineeringDatumType">
  <xsd:annotation>
    <xsd:documentation>Defines the origin and axes directions of an engineering coordinate reference
system. Used in a local context only: </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractDatumType"/>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="engineeringDatumRef" type="gml:EngineeringDatumRefType"
substitutionGroup="gml:datumRef"/>
 <!-- ============================================================ -->
 <xsd:complexType name="EngineeringDatumRefType">
  <xsd:annotation>
    <xsd:documentation>Association to an Engineering Datum, either referencing or containing the definition
of that Datum. </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:restriction base="gml:DatumRefType">
      <xsd:sequence>
        <xsd:element ref="gml:EngineeringDatum" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xsd:restriction>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="ImageDatum" type="gml:ImageDatumType" substitutionGroup="gml:_Datum"/>
 <!-- ============================================================ -->
 <xsd:complexType name="ImageDatumType">
  <xsd:annotation>
    <xsd:documentation>Defines the origin of an image coordinate reference system. Used in a local
context only: For more information, see OGC Abstract Specification Topic 2. </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractDatumType">
      <xsd:sequence>
        <xsd:element ref="gml:pixelInCell"/>
```

```
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
<!-- ============================================================== -->
<xsd:element name="imageDatumRef" type="gml:ImageDatumRefType"
substitutionGroup="gml:datumRef"/>
<!-- ============================================================== -->
<xsd:complexType name="ImageDatumRefType">
  <xsd:annotation>
    <xsd:documentation>Association to an Image Datum, either referencing or containing the definition of
that Datum. </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:restriction base="gml:DatumRefType">
      <xsd:sequence>
        <xsd:element ref="gml:ImageDatum" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xsd:restriction>
  </xsd:complexContent>
</xsd:complexType>
<!-- ============================================================== -->
<xsd:element name="pixelInCell" type="gml:PixelInCellType"/>
<!-- ============================================================== -->
<xsd:simpleType name="PixelInCellType">
  <xsd:annotation>
    <xsd:documentation>Specification of the way the image grid is associated with the image data
attributes. </xsd:documentation>
  </xsd:annotation>
  <xsd:restriction base="string">
    <xsd:enumeration value="cellCenter">
      <xsd:annotation>
        <xsd:documentation>The origin of the image coordinate system is the center of a grid cell or image
pixel. </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="cellCorner">
      <xsd:annotation>
        <xsd:documentation>The origin of the image coordinate system is the corner of a grid cell, or half-way
between adjacent image pixel centres. </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>
<!-- ============================================================== -->
<xsd:element name="VerticalDatum" type="gml:VerticalDatumType" substitutionGroup="gml:_Datum"/>
<!-- ============================================================== -->
<xsd:complexType name="VerticalDatumType">
  <xsd:annotation>
    <xsd:documentation>A textual description and/or a set of parameters identifying a particular reference
level surface used as a zero-height surface, including its position and orientation with respect to the Earth for
any of the height types recognized by this standard. </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractDatumType">
      <xsd:sequence>
        <xsd:element ref="gml:verticalDatumType" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
```



```
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:element name="verticalDatumRef" type="gml:VerticalDatumRefType"
substitutionGroup="gml:datumRef"/>
  <!-- ============================================================ -->
  <xsd:complexType name="VerticalDatumRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Vertical Datum, either referencing or containing the definition of
that Datum. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:DatumRefType">
        <xsd:sequence>
          <xsd:element ref="gml:VerticalDatum" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:element name="verticalDatumType" type="gml:VerticalDatumTypeType"/>
  <!-- ============================================================ -->
  <xsd:simpleType name="VerticalDatumTypeType">
    <xsd:annotation>
      <xsd:documentation>Type of a vertical datum.</xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="string">
      <xsd:enumeration value="geoidal">
        <xsd:annotation>
          <xsd:documentation>The zero value of the associated vertical coordinate system axis is defined to
approximate a constant potential surface, usually the geoid. Such a reference surface is usually determined
by a national or scientific authority, and is then a well-known, named datum. </xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="depth">
        <xsd:annotation>
          <xsd:documentation>The zero point of the vertical axis is defined by a surface that has meaning for
the purpose which the associated vertical measurements are used for. For hydrographic charts, this is often
a predicted nominal sea surface (i.e. without waves or other wind and current effects) that occurs at low tide.
For some examples, see OGC Abstract Specification Topic 2. </xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="barometric">
        <xsd:annotation>
          <xsd:documentation>Atmospheric pressure is the basis for the definition of the origin of the
associated vertical coordinate system axis. For more information, see OGC Abstract Specification Topic 2.
</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="othersurface">
        <xsd:annotation>
          <xsd:documentation>In some cases, e.g. oil exploration and production, a geological feature, such as
the top or bottom of a geologically identifiable and meaningful subsurface layer, is used as a vertical datum.
Other variations to the above three vertical datum types may exist and are all included in this type.
</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
    </xsd:restriction>
  </xsd:simpleType>
```

```
  <!-- ============================================================= -->
  <xsd:complexType name="TemporalDatumRestrictionType" abstract="true">
    <xsd:annotation>
      <xsd:documentation>Partially defines the origin of a temporal coordinate reference system. This type
restricts the AbstractDatumType to remove the "anchorPoint" and "realizationEpoch" elements.
</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:AbstractDatumType">
        <xsd:sequence>
          <xsd:element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
          <xsd:element name="datumID" type="gml:ExtendedIdentifierType"/>
        </xsd:sequence>
        <xsd:attribute ref="gml:id" use="optional"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="TemporalDatum" type="gml:TemporalDatumType"
substitutionGroup="gml:_Datum"/>
  <!-- ============================================================= -->
  <xsd:complexType name="TemporalDatumType">
    <xsd:annotation>
      <xsd:documentation>Defines the origin of a temporal coordinate reference system. This type extends
the TemporalDatumRestrictionType to add the "origin" element with the dateTime type.
</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:TemporalDatumRestrictionType">
        <xsd:sequence>
          <xsd:element name="origin" type="dateTime">
            <xsd:annotation>
              <xsd:documentation>The date and time origin of this temporal datum. </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="temporalDatumRef" type="gml:TemporalDatumRefType"
substitutionGroup="gml:datumRef"/>
  <!-- ============================================================= -->
  <xsd:complexType name="TemporalDatumRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Temporal Datum, either referencing or containing the definition of
that Datum. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:DatumRefType">
        <xsd:sequence>
          <xsd:element ref="gml:TemporalDatum" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="GeodeticDatum" type="gml:GeodeticDatumType" substitutionGroup="gml:_Datum"/>
  <!-- ============================================================= -->
```

```
  <xsd:complexType name="GeodeticDatumType">
    <xsd:annotation>
      <xsd:documentation>The Geodetic Datum is an ellipsoid with its position and orientation with respect to
the Earth. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractDatumType">
        <xsd:sequence>
          <xsd:element name="usesPrimeMeridian" type="gml:PrimeMeridianRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the prime meridian used by this geodetic datum.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="usesEllipsoid" type="gml:EllipsoidRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the ellipsoid used by this geodetic datum.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:element name="geodeticDatumRef" type="gml:GeodeticDatumRefType"
substitutionGroup="gml:datumRef"/>
  <!-- ============================================================ -->
  <xsd:complexType name="GeodeticDatumRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Geodetic Datum, either referencing or containing the definition of
that Datum. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:DatumRefType">
        <xsd:sequence>
          <xsd:element ref="gml:GeodeticDatum" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:element name="PrimeMeridian" type="gml:PrimeMeridianType"
substitutionGroup="gml:_CRSObject"/>
  <!-- ============================================================ -->
  <xsd:complexType name="PrimeMeridianType">
    <xsd:annotation>
      <xsd:documentation>A prime meridian defines the origin from which longitude values are
determined.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCRSObjectType">
        <xsd:sequence>
          <xsd:element name="meridianID" type="gml:IdentifierType">
            <xsd:annotation>
              <xsd:documentation>Identification of this prime meridian. </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="greenwichLongitude" type="gml:AngleChoiceType">
```



```
          <xsd:annotation>
            <xsd:documentation>Longitude of the prime meridian measured from the Greenwich meridian, positive eastward. If the datum type is geodetic and the prime meridian name is not supplied, then the prime meridian name is taken to be "Greenwich" and the Greenwich longitude value is taken to be zero. </xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================== -->
 <xsd:element name="primeMeridianRef" type="gml:PrimeMeridianRefType"/>
 <!-- ============================================================== -->
 <xsd:complexType name="PrimeMeridianRefType">
   <xsd:annotation>
     <xsd:documentation>Association to a Prime Meridian, either referencing or containing the definition of that Prime Meridian. </xsd:documentation>
   </xsd:annotation>
   <xsd:sequence>
     <xsd:element ref="gml:PrimeMeridian" minOccurs="0"/>
   </xsd:sequence>
   <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:complexType>
 <!-- ============================================================== -->
 <xsd:element name="Ellipsoid" type="gml:EllipsoidType" substitutionGroup="gml:_CRSObject"/>
 <!-- ============================================================== -->
 <xsd:complexType name="EllipsoidType">
   <xsd:annotation>
     <xsd:documentation>An ellipsoid is a geometric figure that can be used to describe the approximate shape of the earth. In mathematical terms, it is a surface formed by the rotation of an ellipse about an axis.</xsd:documentation>
   </xsd:annotation>
   <xsd:complexContent>
     <xsd:extension base="gml:AbstractCRSObjectType">
       <xsd:sequence>
         <xsd:element name="ellipsoidID" type="gml:IdentifierType">
           <xsd:annotation>
             <xsd:documentation>Identification of this ellipsoid. </xsd:documentation>
           </xsd:annotation>
         </xsd:element>
         <xsd:element name="semiMajorAxis" type="gml:LengthType">
           <xsd:annotation>
             <xsd:documentation>Length of the semi-major axis of the ellipsoid. </xsd:documentation>
           </xsd:annotation>
         </xsd:element>
         <xsd:element ref="gml:SecondDefiningParameter"/>
       </xsd:sequence>
     </xsd:extension>
   </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================== -->
 <xsd:element name="ellipsoidRef" type="gml:EllipsoidRefType"/>
 <!-- ============================================================== -->
 <xsd:complexType name="EllipsoidRefType">
   <xsd:annotation>
     <xsd:documentation>Association to an Ellipsoid, either referencing or containing the definition of that Ellipsoid. </xsd:documentation>
   </xsd:annotation>
   <xsd:sequence>
```

```
    <xsd:element ref="gml:Ellipsoid" minOccurs="0"/>
   </xsd:sequence>
   <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="SecondDefiningParameter" type="gml:SecondDefiningParameterType"/>
  <!-- ============================================================= -->
  <xsd:complexType name="SecondDefiningParameterType">
   <xsd:annotation>
    <xsd:documentation>Definition of the second parameter which defines the shape of an ellipsoid. An ellipsoid requires two defining parameters: semi-major axis and inverse flattening or semi-major axis and semi-minor axis. When the reference body is a sphere rather than an ellipsoid, only a single defining parameter is required, namely the radius of the sphere; in that case, the semi-major axis "degenerates" into the radius of the sphere.</xsd:documentation>
   </xsd:annotation>
   <xsd:choice>
    <xsd:element name="inverseFlattening" type="gml:ScaleType">
     <xsd:annotation>
      <xsd:documentation>Inverse flattening value of the ellipsoid. </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
    <xsd:element name="semiMinorAxis" type="gml:LengthType">
     <xsd:annotation>
      <xsd:documentation>Length of the semi-minor axis of the ellipsoid. </xsd:documentation>
     </xsd:annotation>
    </xsd:element>
    <xsd:element name="isSphere">
     <xsd:annotation>
      <xsd:documentation>The ellipsoid is degenerate and is actually a sphere. The sphere is completely defined by the semi-major axis, which is the radius of the sphere. </xsd:documentation>
     </xsd:annotation>
     <xsd:simpleType>
      <xsd:restriction base="string">
       <xsd:enumeration value="Sphere"/>
      </xsd:restriction>
     </xsd:simpleType>
    </xsd:element>
   </xsd:choice>
  </xsd:complexType>
</xsd:schema>
```

coordinateSystems.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified" xml:lang="en" version="3.00">
  <xsd:annotation>
   <xsd:appinfo source="urn:opengis:specification:gml:schema-coordinateSystems:v3.00"/>
   <xsd:documentation>
    <name>coordinateSystems.xsd</name>
    <version>3.0</version>
    <scope>How to encode coordinate system definitions. </scope>
    <description>Builds on referenceSystems.xsd to encode the data needed to define coordinate systems, including the specific subtypes of coordinate systems. GML 3.0 candidate schema, primary editor: Arliss Whiteside. Last updated 2002/11/21. </description>
    <copyright>Copyright (c) 2001-2002 OpenGIS, All Rights Reserved.</copyright>
```

```
   <conformance>This schema encodes the Coordinate System (CS_) package of the extended UML Model for OGC Abstract Specification Topic 2: Spatial Referencing by Coordinates. That UML model is adapted from ISO 19111 - Spatial referencing by coordinates, as described in Annex B of Topic 2.
</conformance>
  </xsd:documentation>
 </xsd:annotation>
 <!-- ==============================================================
    includes and imports
    ============================================================== -->
 <xsd:include schemaLocation="referenceSystems.xsd"/>
 <!-- ==============================================================
    elements and types
    ============================================================== -->
 <xsd:element name="CoordinateSystemAxis" type="gml:CoordinateSystemAxisType"
substitutionGroup="gml:_CRSObject"/>
 <!-- ============================================================ -->
 <xsd:complexType name="CoordinateSystemAxisType">
  <xsd:annotation>
   <xsd:documentation>Definition of a coordinate system axis. </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
   <xsd:extension base="gml:AbstractCRSObjectType">
    <xsd:sequence>
     <xsd:element name="axisID" type="gml:IdentifierType" minOccurs="0">
      <xsd:annotation>
       <xsd:documentation>Identification of this coordinate system axis. </xsd:documentation>
      </xsd:annotation>
     </xsd:element>
     <xsd:element name="axisAbbrev" type="string">
      <xsd:annotation>
       <xsd:documentation>The abbreviation used for this coordinate system axis. This abbreviation can be used to identify the ordinates in a coordinate tuple. Examples are X and Y. </xsd:documentation>
      </xsd:annotation>
     </xsd:element>
     <xsd:element name="axisDirection" type="string">
      <xsd:annotation>
       <xsd:documentation>Direction of this coordinate system axis (or in the case of Cartesian projected coordinates, the direction of this coordinate system axis at the origin). Examples: north or south, east or west, up or down. Within any set of coordinate system axis, only one of each pair of terms can be used.
</xsd:documentation>
      </xsd:annotation>
     </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="uom" type="anyURI" use="required">
     <xsd:annotation>
      <xsd:documentation>Identifier of the unit of measure used for this coordinate system axis.
</xsd:documentation>
     </xsd:annotation>
    </xsd:attribute>
   </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="coordinateSystemAxisRef" type="gml:CoordinateSystemAxisRefType"/>
 <!-- ============================================================ -->
 <xsd:complexType name="CoordinateSystemAxisRefType">
  <xsd:annotation>
   <xsd:documentation>Association to a Coordinate System Axis, either referencing or containing the definition of that Coordinate System Axis. </xsd:documentation>
  </xsd:annotation>
```

```
    <xsd:sequence>
      <xsd:element ref="gml:CoordinateSystemAxis" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
  </xsd:complexType>
  <!-- ============================================================ -->
  <!-- ============================================================ -->
  <xsd:element name="_CoordinateSystem" type="gml:AbstractCoordinateSystemType" abstract="true"
substitutionGroup="gml:_CRSObject"/>
  <!-- ============================================================ -->
  <xsd:complexType name="AbstractCoordinateSystemType" abstract="true">
    <xsd:annotation>
      <xsd:documentation>A coordinate system (CS) is the set of coordinate system axes that spans a given
coordinate space. A CS is derived from a set of (mathematical) rules for specifying how coordinates in a
given space are to be assigned to points. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCRSObjectType">
        <xsd:sequence>
          <xsd:element name="csID" type="gml:IdentifierType">
            <xsd:annotation>
              <xsd:documentation>Identification of this coordinate system. </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="usesAxis" type="gml:CoordinateSystemAxisRefType"
maxOccurs="unbounded">
            <xsd:annotation>
              <xsd:documentation>Associations to ordered list of Coordinate System Axes.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:element name="coordinateSystemRef" type="gml:CoordinateSystemRefType"/>
  <!-- ============================================================ -->
  <xsd:complexType name="CoordinateSystemRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Coordinate System, either referencing or containing the definition
of that Coordinate System. </xsd:documentation>
    </xsd:annotation>
    <xsd:sequence>
      <xsd:element ref="gml:_CoordinateSystem" minOccurs="0"/>
    </xsd:sequence>
    <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:element name="EllipsoidalCS" type="gml:EllipsoidalCSType"
substitutionGroup="gml:_CoordinateSystem"/>
  <!-- ============================================================ -->
  <xsd:complexType name="EllipsoidalCSType">
    <xsd:annotation>
      <xsd:documentation>A two- or three-dimensional coordinate system in which position is specified by
geodetic latitude, geodetic longitude, and (in the three-dimensional case) ellipsoidal height.
</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateSystemType"/>
```

```
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="ellipsoidalCSRef" type="gml:EllipsoidalCSRefType"
substitutionGroup="gml:coordinateSystemRef"/>
  <!-- ============================================================== -->
  <xsd:complexType name="EllipsoidalCSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to an Ellipsoidal Coordinate System, either referencing or containing
the definition of that Coordinate System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:EllipsoidalCS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="CartesianCS" type="gml:CartesianCSType"
substitutionGroup="gml:_CoordinateSystem"/>
  <!-- ============================================================== -->
  <xsd:complexType name="CartesianCSType">
    <xsd:annotation>
      <xsd:documentation>A 1-, 2-, or 3-dimensional coordinate system. Gives the position of points relative
to orthogonal straight axes in the 2- and 3-dimensional cases. In the 1-dimensional case, it contains a single
straight coordinate axis. In the multi-dimensional case, all axes shall have the same length unit of measure.
</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateSystemType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="cartesianCSRef" type="gml:CartesianCSRefType"
substitutionGroup="gml:coordinateSystemRef"/>
  <!-- ============================================================== -->
  <xsd:complexType name="CartesianCSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Cartesian Coordinate System, either referencing or containing the
definition of that Coordinate System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:CartesianCS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="GravityRelatedCS" type="gml:GravityRelatedCSType"
substitutionGroup="gml:_CoordinateSystem"/>
  <!-- ============================================================== -->
  <xsd:complexType name="GravityRelatedCSType">
    <xsd:annotation>
```

```
      <xsd:documentation>A one-dimensional coordinate system used to record the heights (or depths) of
points dependent on the Earth's gravity field. An exact definition is deliberately not provided as the
complexities of the subject fall outside the scope of this specification. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateSystemType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:element name="gravityRelatedCSRef" type="gml:GravityRelatedCSRefType"
substitutionGroup="gml:coordinateSystemRef"/>
  <!-- ============================================================ -->
  <xsd:complexType name="GravityRelatedCSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Gravity Related Coordinate System, either referencing or
containing the definition of that Coordinate System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:GravityRelatedCS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:element name="TemporalCS" type="gml:TemporalCSType"
substitutionGroup="gml:_CoordinateSystem"/>
  <!-- ============================================================ -->
  <xsd:complexType name="TemporalCSType">
    <xsd:annotation>
      <xsd:documentation>A 1-dimensional coordinate system containing a single time axis, used to describe
the temporal position of a point in the specified time units from a specified time origin. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateSystemType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:element name="temporalCSRef" type="gml:TemporalCSRefType"
substitutionGroup="gml:coordinateSystemRef"/>
  <!-- ============================================================ -->
  <xsd:complexType name="TemporalCSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Temporal Coordinate System, either referencing or containing the
definition of that Coordinate System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:TemporalCS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================ -->
  <xsd:element name="LinearCS" type="gml:LinearCSType" substitutionGroup="gml:_CoordinateSystem"/>
  <!-- ============================================================ -->
```

```
  <xsd:complexType name="LinearCSType">
    <xsd:annotation>
      <xsd:documentation>A coordinate system that is one-dimensional and consists of the points that lie on
the single axis described. The associated ordinate is the distance from the specified origin to the point along
the axis. Example: usage of the line feature representing a road to describe points on or along that road.
</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateSystemType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="linearCSRef" type="gml:LinearCSRefType"
substitutionGroup="gml:coordinateSystemRef"/>
  <!-- ============================================================= -->
  <xsd:complexType name="LinearCSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Linear Coordinate System, either referencing or containing the
definition of that Coordinate System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:LinearCS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="UserDefinedCS" type="gml:UserDefinedCSType"
substitutionGroup="gml:_CoordinateSystem"/>
  <!-- ============================================================= -->
  <xsd:complexType name="UserDefinedCSType">
    <xsd:annotation>
      <xsd:documentation>A two- or three-dimensional coordinate system that consists of any combination of
coordinate axes not covered by any other Coordinate System type. An example is a multilinear coordinate
system which contains one coordinate axis that may have any 1-D shape which has no intersections with
itself. This non-straight axis is supplemented by one or two straight axes to complete a 2 or 3 dimensional
coordinate system. The non-straight axis is typically incrementally straight or curved. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateSystemType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="userDefinedCSRef" type="gml:UserDefinedCSRefType"
substitutionGroup="gml:coordinateSystemRef"/>
  <!-- ============================================================= -->
  <xsd:complexType name="UserDefinedCSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a User Defined Coordinate System, either referencing or containing
the definition of that Coordinate System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:UserDefinedCS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
```

```
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="SphericalCS" type="gml:SphericalCSType"
substitutionGroup="gml:_CoordinateSystem"/>
  <!-- ============================================================= -->
  <xsd:complexType name="SphericalCSType">
    <xsd:annotation>
      <xsd:documentation>A three-dimensional coordinate system with one distance measured from the origin
and two angular coordinates. Not to be confused with an ellipsoidal coordinate system based on an ellipsoid
"degenerated" into a sphere. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateSystemType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="sphericalCSRef" type="gml:SphericalCSRefType"
substitutionGroup="gml:coordinateSystemRef"/>
  <!-- ============================================================= -->
  <xsd:complexType name="SphericalCSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Spherical Coordinate System, either referencing or containing the
definition of that Coordinate System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:SphericalCS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="PolarCS" type="gml:PolarCSType" substitutionGroup="gml:_CoordinateSystem"/>
  <!-- ============================================================= -->
  <xsd:complexType name="PolarCSType">
    <xsd:annotation>
      <xsd:documentation>A two-dimensional coordinate system in which position is specified by the distance
to the origin and the angle between the line from the origin to a point and a reference direction.
</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateSystemType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="polarCSRef" type="gml:PolarCSRefType"
substitutionGroup="gml:coordinateSystemRef"/>
  <!-- ============================================================= -->
  <xsd:complexType name="PolarCSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Polar Coordinate System, either referencing or containing the
definition of that Coordinate System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateSystemRefType">
        <xsd:sequence>
```

```
        <xsd:element ref="gml:PolarCS" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xsd:restriction>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="CylindricalCS" type="gml:CylindricalCSType"
substitutionGroup="gml:_CoordinateSystem"/>
 <!-- ============================================================ -->
 <xsd:complexType name="CylindricalCSType">
  <xsd:annotation>
    <xsd:documentation>A three-dimensional coordinate system consisting of a polar coordinate system
extended by a straight coordinate axis perpendicular to the plane spanned by the polar coordinate system.
</xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractCoordinateSystemType"/>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="cylindricalCSRef" type="gml:CylindricalCSRefType"
substitutionGroup="gml:coordinateSystemRef"/>
 <!-- ============================================================ -->
 <xsd:complexType name="CylindricalCSRefType">
  <xsd:annotation>
    <xsd:documentation>Association to a Cylindrical Coordinate System, either referencing or containing the
definition of that Coordinate System. </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:restriction base="gml:CoordinateSystemRefType">
      <xsd:sequence>
        <xsd:element ref="gml:CylindricalCS" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xsd:restriction>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="ObliqueCartesianCS" type="gml:ObliqueCartesianCSType"
substitutionGroup="gml:_CoordinateSystem"/>
 <!-- ============================================================ -->
 <xsd:complexType name="ObliqueCartesianCSType">
  <xsd:annotation>
    <xsd:documentation>A coordinate system with straight lines that are not necessarily orthogonal.
</xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractCoordinateSystemType"/>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="obliqueCartesianCSRef" type="gml:ObliqueCartesianCSRefType"
substitutionGroup="gml:coordinateSystemRef"/>
 <!-- ============================================================ -->
 <xsd:complexType name="ObliqueCartesianCSRefType">
  <xsd:annotation>
    <xsd:documentation>Association to an Oblique Cartesian Coordinate System, either referencing or
containing the definition of that Coordinate System. </xsd:documentation>
  </xsd:annotation>
```

```
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:ObliqueCartesianCS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

coordinateOperations.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/gml" xmlns:gml="http://www.opengis.net/gml"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" xml:lang="en" version="3.00">
  <annotation>
    <appinfo source="urn:opengis:specification:gml:schema-coordinateOperations:v3.00"/>
    <documentation>
      <name>coordinateOperations.xsd</name>
      <version>3.0</version>
      <scope>How to encode coordinate operation definitions. </scope>
      <description>Builds on referenceSystems.xsd to encode the data needed to define coordinate
operations, including Transformations, Conversions, and other specific subtypes of operations. GML 3.0
candidate schema, primary editor: Arliss Whiteside. Last updated 2002/12/13. </description>
      <copyright>Copyright (c) 2001-2002 OpenGIS, All Rights Reserved.</copyright>
      <conformance>This schema encodes the Coordinate Operation (CC_) package of the extended UML
Model for OGC Abstract Specification Topic 2: Spatial Referencing by Coordinates. That UML model is
adapted from ISO 19111 - Spatial referencing by coordinates, as described in Annex B of Topic 2.
</conformance>
      <ChangeLog>
        <Changes>May-November 2002: See Section 3.5 of document 02-095. </Changes>
        <Change>December 3, 2002. Corrected extension base in OperationMethodType. </Change>
        <Change>December 13, 2002. Corrected "source" attribute in appinfo. </Change>
      </ChangeLog>
    </documentation>
  </annotation>
  <!-- ==============================================================
       includes and imports
      ============================================================== -->
  <include schemaLocation="referenceSystems.xsd"/>
  <include schemaLocation="dataQuality.xsd"/>
  <!-- ==============================================================
       elements and types
      ============================================================== -->
  <element name="_CoordinateOperation" type="gml:AbstractCoordinateOperationType" abstract="true"
substitutionGroup="gml:_CRSObject"/>
  <!-- ============================================================ -->
  <complexType name="AbstractCoordinateOperationType" abstract="true">
    <annotation>
      <documentation>A mathematical operation on coordinates that transforms or converts coordinates to
another coordinate reference system.</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCRSObjectType">
        <sequence>
          <element name="coordinateOperationID" type="gml:ExtendedIdentifierType">
```

```
          <annotation>
            <documentation>Identification of this Coordinate Operation. </documentation>
          </annotation>
        </element>
        <element name="coordinateOperationVersion" type="string" minOccurs="0">
          <annotation>
            <documentation>Version of the coordinate transformation (i.e., instantiation due to the stochastic
nature of the parameters). Mandatory when describing a transformation, and should not be supplied for a
conversion. </documentation>
          </annotation>
        </element>
        <element name="sourceDimensions" type="positiveInteger">
          <annotation>
            <documentation>Required number of dimensions in the source CRS. </documentation>
          </annotation>
        </element>
        <element name="targetDimensions" type="positiveInteger">
          <annotation>
            <documentation>Required number of dimensions in the target CRS. </documentation>
          </annotation>
        </element>
        <element ref="gml:_PositionalAccuracy" minOccurs="0">
          <annotation>
            <documentation>Estimate of the impact of this coordinate operation on point position accuracy.
Gives position error estimates for target coordinates of this coordinate operation, assuming no errors in
source coordinates. </documentation>
          </annotation>
        </element>
        <element name="sourceCRS" type="gml:CRSRefType" minOccurs="0">
          <annotation>
            <documentation>Association to the source CRS (coordinate reference system) of this coordinate
operation. </documentation>
          </annotation>
        </element>
        <element name="targetCRS" type="gml:CRSRefType" minOccurs="0">
          <annotation>
            <documentation>Association to the target CRS (coordinate reference system) of this coordinate
operation. For constraints on multiplicity of "sourceCRS" and "targetCRS", see UML model of Coordinate
Operation package in OGC Abstract Specification topic 2. </documentation>
          </annotation>
        </element>
      </sequence>
    </extension>
  </complexContent>
 </complexType>
 <!-- ============================================================ -->
 <element name="coordinateOperationRef" type="gml:CoordinateOperationRefType"/>
 <!-- ============================================================ -->
 <complexType name="CoordinateOperationRefType">
   <annotation>
     <documentation>Association to a Coordinate Operation, either referencing or containing the definition of
that Coordinate Operation. </documentation>
   </annotation>
   <sequence>
     <element ref="gml:_CoordinateOperation" minOccurs="0"/>
   </sequence>
   <attributeGroup ref="gml:AssociationAttributeGroup"/>
 </complexType>
 <!-- ============================================================ -->
```

```
  <element name="ConcatenatedOperation" type="gml:ConcatenatedOperationType"
substitutionGroup="gml:_CoordinateOperation"/>
  <!-- ============================================================= -->
  <complexType name="ConcatenatedOperationType">
    <annotation>
      <documentation>An ordered sequence of two or more single coordinate operations. The sequence of
operations is constrained by the requirement that the source coordinate reference system of step (n+1) must
be the same as the target coordinate reference system of step (n). The source coordinate reference system
of the first step and the target coordinate reference system of the last step are the source and target
coordinate reference system associated with the concatenated operation. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCoordinateOperationType">
        <sequence>
          <element name="usesOperation" type="gml:SingleOperationRefType" minOccurs="2"
maxOccurs="unbounded">
            <annotation>
              <documentation>Ordered sequence of associations to the two or more single operations used by
this concatenated operation. </documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="concatenatedOperationRef" type="gml:ConcatenatedOperationRefType"
substitutionGroup="gml:coordinateOperationRef"/>
  <!-- ============================================================= -->
  <complexType name="ConcatenatedOperationRefType">
    <annotation>
      <documentation>Association to a Concatenated Operation, either referencing or containing the definition
of that Concatenated Operation. </documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:CoordinateOperationRefType">
        <sequence>
          <element ref="gml:ConcatenatedOperation" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="_SingleOperation" type="gml:AbstractSingleOperationType" abstract="true"
substitutionGroup="gml:_CoordinateOperation"/>
  <!-- ============================================================= -->
  <complexType name="AbstractSingleOperationType" abstract="true">
    <annotation>
      <documentation>A single (not concatenated) coordinate operation. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCoordinateOperationType"/>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="singleOperationRef" type="gml:SingleOperationRefType"
substitutionGroup="gml:coordinateOperationRef"/>
  <!-- ============================================================= -->
  <complexType name="SingleOperationRefType">
```

```
   <annotation>
     <documentation>Association to a Single Operation, either referencing or containing the definition of that Single Operation. </documentation>
   </annotation>
   <complexContent>
     <restriction base="gml:CoordinateOperationRefType">
       <sequence>
         <element ref="gml:_SingleOperation" minOccurs="0"/>
       </sequence>
       <attributeGroup ref="gml:AssociationAttributeGroup"/>
     </restriction>
   </complexContent>
 </complexType>
 <!-- ============================================================ -->
 <element name="PassThroughOperation" type="gml:PassThroughOperationType" substitutionGroup="gml:_SingleOperation"/>
 <!-- ============================================================ -->
 <complexType name="PassThroughOperationType">
   <annotation>
     <documentation>A pass-through operation specifies that a subset of a coordinate tuple is subject to a specific coordinate operation. </documentation>
   </annotation>
   <complexContent>
     <extension base="gml:AbstractSingleOperationType">
       <sequence>
         <element name="modifiedCoordinate" type="positiveInteger" maxOccurs="unbounded">
           <annotation>
             <documentation>Ordered sequence of positive integers defining the positions in a coordinate tuple of the coordinates affected by this pass-through operation. </documentation>
           </annotation>
         </element>
         <element name="usesOperation" type="gml:OperationRefType">
           <annotation>
             <documentation>Association to the operation applied to the specified ordinates. </documentation>
           </annotation>
         </element>
       </sequence>
     </extension>
   </complexContent>
 </complexType>
 <!-- ============================================================ -->
 <element name="passThroughOperationRef" type="gml:PassThroughOperationRefType" substitutionGroup="gml:singleOperationRef"/>
 <!-- ============================================================ -->
 <complexType name="PassThroughOperationRefType">
   <annotation>
     <documentation>Association to a Pass Through Operation, either referencing or containing the definition of that Pass Through Operation. </documentation>
   </annotation>
   <complexContent>
     <restriction base="gml:SingleOperationRefType">
       <sequence>
         <element ref="gml:PassThroughOperation" minOccurs="0"/>
       </sequence>
       <attributeGroup ref="gml:AssociationAttributeGroup"/>
     </restriction>
   </complexContent>
 </complexType>
 <!-- ============================================================ -->
```



```
  <element name="_Operation" type="gml:AbstractOperationType" abstract="true"
substitutionGroup="gml:_SingleOperation"/>
  <!-- ====================================================================== -->
  <complexType name="AbstractOperationType" abstract="true">
    <annotation>
      <documentation>A parameterized mathematical operation on coordinates that transforms or converts
coordinates to another coordinate reference system. This coordinate operation uses an operation method,
usually with associated parameter values. All concrete types derived from this type shall thus extend this
type to include a "usesMethod" element that references one element substitutable for the
"_GeneralOperationMethod" element. Similarly, all concrete types derived from this type shall extend this
type to include zero or more elements each named "uses...Value" that each use the type of an element
substitutable for the "_generalParameterValue" element. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractSingleOperationType"/>
    </complexContent>
  </complexType>
  <!-- ====================================================================== -->
  <element name="operationRef" type="gml:OperationRefType"
substitutionGroup="gml:singleOperationRef"/>
  <!-- ====================================================================== -->
  <complexType name="OperationRefType">
    <annotation>
      <documentation>Association to an abstract Operation, either referencing or containing the definition of
that Operation. </documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:SingleOperationRefType">
        <sequence>
          <element ref="gml:_Operation" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ====================================================================== -->
  <element name="_GeneralConversion" type="gml:AbstractGeneralConversionType" abstract="true"
substitutionGroup="gml:_Operation"/>
  <!-- ====================================================================== -->
  <complexType name="AbstractGeneralConversionType" abstract="true">
    <annotation>
      <documentation>An abstract operation on coordinates that does not include any change of Datum. The
best-known example of a coordinate conversion is a map projection. The parameters describing coordinate
conversions are defined rather than empirically derived. Note that some conversions have no parameters.
</documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:AbstractOperationType">
        <sequence>
          <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
          <element name="coordinateOperationID" type="gml:ExtendedIdentifierType"/>
          <element name="sourceDimensions" type="positiveInteger"/>
          <element name="targetDimensions" type="positiveInteger"/>
          <element name="_PositionalAccuracy" type="gml:AbstractPositionalAccuracyType" minOccurs="0"/>
        </sequence>
        <attribute ref="gml:id" use="optional"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ====================================================================== -->
```

```
  <element name="generalConversionRef" type="gml:GeneralConversionRefType"
substitutionGroup="gml:operationRef"/>
  <!-- ============================================================ -->
  <complexType name="GeneralConversionRefType">
    <annotation>
      <documentation>Association to a General Conversion, either referencing or containing the definition of
that Conversion. </documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:OperationRefType">
        <sequence>
          <element ref="gml:_GeneralConversion" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="Conversion" type="gml:ConversionType" substitutionGroup="gml:_GeneralConversion"/>
  <!-- ============================================================ -->
  <complexType name="ConversionType">
    <annotation>
      <documentation>A concrete general-purpose operation on coordinates that does not include any change
of Datum. The best-known example of a coordinate conversion is a map projection. The parameters
describing coordinate conversions are defined rather than empirically derived. Note that some conversions
have no parameters. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGeneralConversionType">
        <sequence>
          <element name="usesMethod" type="gml:OperationMethodRefType">
            <annotation>
              <documentation>Association to the operation method used by this conversion operation.
</documentation>
            </annotation>
          </element>
          <element name="usesValue" type="gml:ParameterValueType" minOccurs="0"
maxOccurs="unbounded">
            <annotation>
              <documentation>Unordered list of composition associations to the set of parameter values used by
this conversion operation. </documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="conversionRef" type="gml:ConversionRefType"
substitutionGroup="gml:generalConversionRef"/>
  <!-- ============================================================ -->
  <complexType name="ConversionRefType">
    <annotation>
      <documentation>Association to a concrete general-purpose Conversion, either referencing or containing
the definition of that Conversion. </documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:GeneralConversionRefType">
        <sequence>
          <element ref="gml:Conversion" minOccurs="0"/>
```

```
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>
<!-- ============================================================ -->
<element name="_GeneralTransformation" type="gml:AbstractGeneralTransformationType" abstract="true"
substitutionGroup="gml:_Operation"/>
<!-- ============================================================ -->
<complexType name="AbstractGeneralTransformationType" abstract="true">
  <annotation>
    <documentation>An abstract operation on coordinates that usually includes a change of Datum. The
parameters of a coordinate transformation are empirically derived from data containing the coordinates of a
series of points in both coordinate reference systems. This computational process is usually "over-
determined", allowing derivation of error (or accuracy) estimates for the transformation. Also, the stochastic
nature of the parameters may result in multiple (different) versions of the same coordinate transformation.
</documentation>
  </annotation>
  <complexContent>
    <restriction base="gml:AbstractOperationType">
      <sequence>
        <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
        <element name="coordinateOperationID" type="gml:ExtendedIdentifierType"/>
        <element name="coordinateOperationVersion" type="string"/>
        <element name="sourceDimensions" type="positiveInteger"/>
        <element name="targetDimensions" type="positiveInteger"/>
        <element name="_PositionalAccuracy" type="gml:AbstractPositionalAccuracyType" minOccurs="0"/>
        <element name="sourceCRS" type="gml:CRSRefType"/>
        <element name="targetCRS" type="gml:CRSRefType"/>
      </sequence>
      <attribute ref="gml:id" use="optional"/>
    </restriction>
  </complexContent>
</complexType>
<!-- ============================================================ -->
<element name="generalTransformationRef" type="gml:GeneralTransformationRefType"
substitutionGroup="gml:operationRef"/>
<!-- ============================================================ -->
<complexType name="GeneralTransformationRefType">
  <annotation>
    <documentation>Association to a General Transformation, either referencing or containing the definition
of that Transformation. </documentation>
  </annotation>
  <complexContent>
    <restriction base="gml:OperationRefType">
      <sequence>
        <element ref="gml:_GeneralTransformation" minOccurs="0"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>
<!-- ============================================================ -->
<element name="Transformation" type="gml:TransformationType"
substitutionGroup="gml:_GeneralTransformation"/>
<!-- ============================================================ -->
<complexType name="TransformationType">
  <annotation>
    <documentation>A concrete general-purpose operation on coordinates that usually includes a change of
Datum. The parameters of a coordinate transformation are empirically derived from data containing the
```

```xml
coordinates of a series of points in both coordinate reference systems. This computational process is usually
"over-determined", allowing derivation of error (or accuracy) estimates for the transformation. Also, the
stochastic nature of the parameters may result in multiple (different) versions of the same coordinate
transformation. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGeneralTransformationType">
        <sequence>
          <element name="usesMethod" type="gml:OperationMethodRefType">
            <annotation>
              <documentation>Association to the operation method used by this transformation operation.
</documentation>
            </annotation>
          </element>
          <element name="usesValue" type="gml:ParameterValueType" minOccurs="0"
maxOccurs="unbounded">
            <annotation>
              <documentation>Unordered list of composition associations to the set of parameter values used by
this transformation operation. </documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="transformationRef" type="gml:TransformationRefType"
substitutionGroup="gml:generalTransformationRef"/>
  <!-- ============================================================ -->
  <complexType name="TransformationRefType">
    <annotation>
      <documentation>Association to a Transformation, either referencing or containing the definition of that
Transformation. </documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:GeneralTransformationRefType">
        <sequence>
          <element ref="gml:Transformation" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="_generalParameterValue" type="gml:GeneralParameterValueType" abstract="true"/>
  <!-- ============================================================ -->
  <complexType name="GeneralParameterValueType" abstract="true">
    <annotation>
      <documentation>An abstract parameter value. All concrete types derived from this type shall extend this
type to include one "...value" element with an appropriate type. In addition, all derived concrete types shall
extend this type to include a "valueOfParameter" element that references one element substitutable for the
"OperationParameter" element. </documentation>
    </annotation>
    <sequence/>
  </complexType>
  <!-- ============================================================ -->
  <element name="parameterValue" type="gml:ParameterValueType"
substitutionGroup="gml:_generalParameterValue"/>
  <!-- ============================================================ -->
  <complexType name="ParameterValueType">
```

```
    <annotation>
      <documentation>A parameter value or reference to a file of parameter values. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:GeneralParameterValueType">
        <sequence>
          <choice>
            <element name="value" type="gml:MeasureType">
              <annotation>
                <documentation>Numeric value of an operation parameter, with its associated unit of measure.
</documentation>
              </annotation>
            </element>
            <element name="dmsAngleValue" type="gml:DMSAngleType">
              <annotation>
                <documentation>Value of an angle operation parameter, in either degree-minute-second format or
single value format. </documentation>
              </annotation>
            </element>
            <element name="stringValue" type="string">
              <annotation>
                <documentation>String value of an operation parameter. A string value does not have an
associated unit of measure. </documentation>
              </annotation>
            </element>
            <element name="integerValue" type="positiveInteger">
              <annotation>
                <documentation>Positive integer value of an operation parameter, usually used for a count. An
integer value does not have an associated unit of measure. </documentation>
              </annotation>
            </element>
            <element name="booleanValue" type="boolean">
              <annotation>
                <documentation>Boolean value of an operation parameter  A Boolean value does not have an
associated unit of measure. </documentation>
              </annotation>
            </element>
            <element name="valueList" type="gml:MeasureListType">
              <annotation>
                <documentation>List of two or more numeric values of an operation parameter list, where each
value has the same associated unit of measure. An element of this type contains a space-separated list of
double values. </documentation>
              </annotation>
            </element>
            <element name="integerValueList" type="gml:integerList">
              <annotation>
                <documentation>List of two or more integer values of an operation parameter list, usually used for
counts. These integer values do not have an associated unit of measure. An element of this type contains a
space-separated list of integer values. </documentation>
              </annotation>
            </element>
            <element name="valueFile" type="anyURI">
              <annotation>
                <documentation>Reference to a file containing multiple parameter values, each numeric value
with its associated unit of measure. </documentation>
              </annotation>
            </element>
          </choice>
          <element name="valueOfParameter" type="gml:OperationParameterRefType">
            <annotation>
```



```
        <documentation>Association to the operation parameter that this is a value of. </documentation>
       </annotation>
      </element>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="_GeneralOperationMethod" type="gml:GeneralOperationMethodType" abstract="true"
substitutionGroup="gml:_CRSObject"/>
  <!-- ============================================================ -->
  <complexType name="GeneralOperationMethodType" abstract="true">
   <annotation>
    <documentation>Abstract definition of an algorithm used to perform a coordinate operation. Most
operation methods use a number of operation parameters (although some coordinate conversions use
none). All concrete types derived from this type shall thus extend this type to include zero or more elements
each named "uses..." that each reference one element substitutable for the "OperationParameter" element.
</documentation>
   </annotation>
   <complexContent>
    <extension base="gml:AbstractCRSObjectType">
     <sequence>
      <element name="methodID" type="gml:IdentifierType">
       <annotation>
        <documentation>Identification of this operation method. </documentation>
       </annotation>
      </element>
      <element name="formula" type="string">
       <annotation>
        <documentation>Formula(s) used by this operation method. The value may be a reference to a
publication. Note that the operation method may not be analytic, in which case this element references or
contains the procedure, not an analytic formula.</documentation>
       </annotation>
      </element>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
  <!-- ============================================================ -->
  <element name="generalOperationMethodRef" type="gml:GeneralOperationMethodRefType"/>
  <!-- ============================================================ -->
  <complexType name="GeneralOperationMethodRefType">
   <annotation>
    <documentation>Association to an abstract Operation Method, either referencing or containing the
definition of that Operation Method. </documentation>
   </annotation>
   <sequence>
    <element ref="gml:_GeneralOperationMethod" minOccurs="0"/>
   </sequence>
   <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- ============================================================ -->
  <element name="OperationMethod" type="gml:OperationMethodType"
substitutionGroup="gml:_GeneralOperationMethod"/>
  <!-- ============================================================ -->
  <complexType name="OperationMethodType">
   <annotation>
    <documentation>Definition of a concrete general-purpose algorithm used to perform a coordinate
operation. Most operation methods use a number of operation parameters (although some coordinate
```

```
conversions use none). Each coordinate operation using the method assigns values to all these parameters.
Most parameter values are numeric, but other types of parameter values are possible. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:GeneralOperationMethodType">
        <sequence>
          <element name="usesParameter" type="gml:OperationParameterRefType" minOccurs="0"
maxOccurs="unbounded">
            <annotation>
              <documentation>Unordered list of associations to the set of operation parameters used by this
operation method. </documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="operationMethodRef" type="gml:OperationMethodRefType"
substitutionGroup="gml:generalOperationMethodRef"/>
  <!-- ============================================================= -->
  <complexType name="OperationMethodRefType">
    <annotation>
      <documentation>Association to a concrete general-purpose Operation Method, either referencing or
containing the definition of that Operation Method. </documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:GeneralOperationMethodRefType">
        <sequence>
          <element ref="gml:OperationMethod" minOccurs="0"/>
        </sequence>
        <attributeGroup ref="gml:AssociationAttributeGroup"/>
      </restriction>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="OperationParameter" type="gml:OperationParameterType"
substitutionGroup="gml:_CRSObject"/>
  <!-- ============================================================= -->
  <complexType name="OperationParameterType">
    <annotation>
      <documentation>The definition of a parameter used by an operation method. </documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractCRSObjectType">
        <sequence>
          <element name="parameterID" type="gml:IdentifierType">
            <annotation>
              <documentation>Identification of this Operation Parameter. </documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ============================================================= -->
  <element name="operationParameterRef" type="gml:OperationParameterRefType"/>
  <!-- ============================================================= -->
  <complexType name="OperationParameterRefType">
    <annotation>
```

```
      <documentation>Association to an Operation Parameter, either referencing or containing the definition of that Operation Parameter. </documentation>
    </annotation>
    <sequence>
      <element ref="gml:OperationParameter" minOccurs="0"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
</schema>
```

coordinateReferenceSystems.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" elementFormDefault="qualified" xml:lang="en" version="3.00">
  <xsd:annotation>
    <xsd:appinfo source="urn:opengis:specification:gml:schema-coordinateReferenceSystems:v3.00"/>
    <xsd:documentation>
      <name>coordinateReferenceSystems.xsd</name>
      <version>3.0</version>
      <scope>How to encode coordinate reference system definitions. </scope>
      <description>Builds on referenceSystems.xsd to encode the data needed to define coordinate reference systems, including the specific subtypes of coordinate reference systems. GML 3.0 candidate schema, primary editor: Arliss Whiteside. Last updated 2002/11/21. </description>
      <copyright>Copyright (c) 2001-2002 OpenGIS, All Rights Reserved.</copyright>
      <conformance>This schema encodes the Coordinate Reference System (SC_) package of the extended UML Model for OGC Abstract Specification Topic 2: Spatial Referencing by Coordinates, with the exception of the abstract "SC_CRS" class. The "SC_CRS" class is encoded in referenceSystems.xsd, to eliminate the (circular) references from coordinateOperations.xsd to coordinateReferenceSystems.xsd. That UML model is adapted from ISO 19111 - Spatial referencing by coordinates, as described in Annex B of Topic 2.
</conformance>
    </xsd:documentation>
  </xsd:annotation>
  <!-- ==============================================================
       includes and imports
      ============================================================== -->
  <xsd:include schemaLocation="coordinateSystems.xsd"/>
  <xsd:include schemaLocation="datums.xsd"/>
  <xsd:include schemaLocation="coordinateOperations.xsd"/>
  <!-- ==============================================================
       elements and types
      ============================================================== -->
  <xsd:element name="_CoordinateReferenceSystem"
type="gml:AbstractCoordinateReferenceSystemType" abstract="true" substitutionGroup="gml:_CRS"/>
  <!-- ============================================================ -->
  <xsd:complexType name="AbstractCoordinateReferenceSystemType" abstract="true">
    <xsd:annotation>
      <xsd:documentation>A coordinate reference system consists of a set of coordinate system axes that is related to the earth through a datum. A coordinate reference system is realised by a set of coordinates. A coordinate reference system is defined by one datum and by one coordinate system.

For the purposes of this specification, a coordinate reference system shall not change with time with the exception of engineering coordinate reference systems defined on moving platforms such as cars, ships, aircraft, and spacecraft. The intention is to exclude the option to describe the time variability of geodetic coordinate reference systems as a result of e.g. tectonic motion. This variability is part of the subject matter of geophysical and geodetic science. The model for spatial referencing by coordinates described in this Abstract Specification is in principle not suitable for such zero-order geodetic problems. Such time-variability
```

```
of coordinate reference systems shall be covered in the spatial referencing model described in this
document by creating different coordinate reference systems each with a different datum for (consecutive)
epochs. The date of realisation of the datum shall be included in its definition. It is further recommended to
include the date of realisation also in the names of the datums and the coordinate reference systems.

Coordinate reference systems are commonly divided into sub-types. The common classification criterion for
sub-typing of coordinate reference systems is the way in which they deal with earth curvature. This has a
direct effect on the portion of the earth's surface that can be covered by that type of CRS with an acceptable
degree of error. The exception to the rule is the subtype "Temporal" which has been added by analogy.
</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCRSType"/>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="coordinateReferenceSystemRef" type="gml:CoordinateReferenceSystemRefType"
substitutionGroup="gml:crsRef"/>
  <!-- ============================================================== -->
  <xsd:complexType name="CoordinateReferenceSystemRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Coordinate Reference System, either referencing or containing the
definition of that Reference System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CRSRefType">
        <xsd:sequence>
          <xsd:element ref="gml:_CoordinateReferenceSystem" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="CompoundCRS" type="gml:CompoundCRSType" substitutionGroup="gml:_CRS"/>
  <!-- ============================================================== -->
  <xsd:complexType name="CompoundCRSType">
    <xsd:annotation>
      <xsd:documentation>A coordinate reference system describing the position of points through two or
more independent coordinate reference systems. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCRSType">
        <xsd:sequence>
          <xsd:element name="includesCRS" type="gml:CoordinateReferenceSystemRefType" minOccurs="2"
maxOccurs="unbounded">
            <xsd:annotation>
              <xsd:documentation>Ordered sequence of associations to the component coordinate reference
systems included in this compound coordinate reference system. </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="compoundCRSRef" type="gml:CompoundCRSRefType"
substitutionGroup="gml:crsRef"/>
  <!-- ============================================================== -->
  <xsd:complexType name="CompoundCRSRefType">
```



```
    <xsd:annotation>
      <xsd:documentation>Association to a Compound Coordinate Reference System, either referencing or
containing the definition of that Reference System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CRSRefType">
        <xsd:sequence>
          <xsd:element ref="gml:CompoundCRS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- =========================================================== -->
  <xsd:element name="GeographicCRS" type="gml:GeographicCRSType"
substitutionGroup="gml:_CoordinateReferenceSystem"/>
  <!-- =========================================================== -->
  <xsd:complexType name="GeographicCRSType">
    <xsd:annotation>
      <xsd:documentation>A coordinate reference system based on an ellipsoidal approximation of the geoid;
this provides an accurate representation of the geometry of geographic features for a large portion of the
earth's surface.</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateReferenceSystemType">
        <xsd:sequence>
          <xsd:element name="usesCS" type="gml:EllipsoidalCSRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the ellipsoidal coordinate system used by this geographic CRS.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="usesDatum" type="gml:GeodeticDatumRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the geodetic datum used by this geographic CRS.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- =========================================================== -->
  <xsd:element name="geographicCRSRef" type="gml:GeographicCRSRefType"
substitutionGroup="gml:coordinateReferenceSystemRef"/>
  <!-- =========================================================== -->
  <xsd:complexType name="GeographicCRSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Geographic Coordinate Reference System, either referencing or
containing the definition of that Reference System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateReferenceSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:GeographicCRS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
```



```
  <!-- ============================================================== -->
  <xsd:element name="VerticalCRS" type="gml:VerticalCRSType"
substitutionGroup="gml:_CoordinateReferenceSystem"/>
  <!-- ============================================================== -->
  <xsd:complexType name="VerticalCRSType">
    <xsd:annotation>
      <xsd:documentation>A coordinate reference system used for recording heights or depths. Vertical CRSs
make use of the direction of gravity to define the concept of height or depth, but the relationship with gravity
may not be straightforward. By implication, ellipsoidal heights (h) cannot be captured in a vertical coordinate
reference system. Ellipsoidal heights cannot exist independently, but only as inseparable part of a 3D
coordinate tuple defined in a geographic 3D coordinate reference system. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateReferenceSystemType">
        <xsd:sequence>
          <xsd:element name="usesCS" type="gml:GravityRelatedCSRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the gravity related coordinate system used by this vertical
CRS. </xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="usesDatum" type="gml:VerticalDatumRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the vertical datum used by this vertical CRS.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="verticalCRSRef" type="gml:VerticalCRSRefType"
substitutionGroup="gml:coordinateReferenceSystemRef"/>
  <!-- ============================================================== -->
  <xsd:complexType name="VerticalCRSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Vertical Coordinate Reference System, either referencing or
containing the definition of that Reference System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateReferenceSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:VerticalCRS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="GeocentricCRS" type="gml:GeocentricCRSType"
substitutionGroup="gml:_CoordinateReferenceSystem"/>
  <!-- ============================================================== -->
  <xsd:complexType name="GeocentricCRSType">
    <xsd:annotation>
      <xsd:documentation>A coordinate reference system that deals with the earth's curvature by taking the
3D spatial view which obviates the need to model the earth's curvature. The origin of a geocentric CRS is at
the approximate centre of mass of the earth. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
```

```
      <xsd:extension base="gml:AbstractCoordinateReferenceSystemType">
        <xsd:sequence>
          <xsd:choice>
            <xsd:element name="usesCartesianCS" type="gml:CartesianCSRefType">
              <xsd:annotation>
                <xsd:documentation>Association to the Cartesian coordinate system used by this geocentric
CRS. </xsd:documentation>
              </xsd:annotation>
            </xsd:element>
            <xsd:element name="usesSphericalCS" type="gml:SphericalCSRefType">
              <xsd:annotation>
                <xsd:documentation>Association to the spherical coordinate system used by this geocentric
CRS.</xsd:documentation>
              </xsd:annotation>
            </xsd:element>
          </xsd:choice>
          <xsd:element name="usesDatum" type="gml:GeodeticDatumRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the geodetic datum used by this geocentric CRS.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ========================================================== -->
  <xsd:element name="geocentricCRSRef" type="gml:GeocentricCRSRefType"
substitutionGroup="gml:coordinateReferenceSystemRef"/>
  <!-- ========================================================== -->
  <xsd:complexType name="GeocentricCRSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Geocentric Coordinate Reference System, either referencing or
containing the definition of that Reference System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateReferenceSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:GeocentricCRS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ========================================================== -->
  <xsd:element name="_GeneralDerivedCRS" type="gml:AbstractGeneralDerivedCRSType" abstract="true"
substitutionGroup="gml:_CoordinateReferenceSystem"/>
  <!-- ========================================================== -->
  <xsd:complexType name="AbstractGeneralDerivedCRSType" abstract="true">
    <xsd:annotation>
      <xsd:documentation>A coordinate reference system that is defined by its coordinate conversion from
another coordinate reference system (not by a datum). </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateReferenceSystemType">
        <xsd:sequence>
          <xsd:element name="sourceCRS" type="gml:CoordinateReferenceSystemRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the coordinate system used by this derived CRS.
</xsd:documentation>
```

```
          </xsd:annotation>
        </xsd:element>
        <xsd:element name="definedByConversion" type="gml:GeneralConversionRefType">
          <xsd:annotation>
            <xsd:documentation>Association to the coordinate conversion used to define this derived CRS.
</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="generalDerivedCRSRef" type="gml:GeneralDerivedCRSRefType"
substitutionGroup="gml:coordinateReferenceSystemRef"/>
 <!-- ============================================================ -->
 <xsd:complexType name="GeneralDerivedCRSRefType">
  <xsd:annotation>
    <xsd:documentation>Association to a General Derived Coordinate Reference System, either referencing
or containing the definition of that Reference System. </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:restriction base="gml:CoordinateReferenceSystemRefType">
      <xsd:sequence>
        <xsd:element ref="gml:_GeneralDerivedCRS" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xsd:restriction>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="ProjectedCRS" type="gml:ProjectedCRSType"
substitutionGroup="gml:_GeneralDerivedCRS"/>
 <!-- ============================================================ -->
 <xsd:complexType name="ProjectedCRSType">
  <xsd:annotation>
    <xsd:documentation>A coordinate reference system used to approximate the shape of the earth on a
planar surface, but in such a way that the distortion that is inherent to the approximation is carefully
controlled and known. Distortion correction is commonly applied to calculated bearings and distances to
produce values that are a close match to actual field values. </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractGeneralDerivedCRSType">
      <xsd:sequence>
        <xsd:element name="usesCS" type="gml:CartesianCSRefType">
          <xsd:annotation>
            <xsd:documentation>Association to the Cartesian coordinate system used by this projected CRS.
</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 <xsd:element name="projectedCRSRef" type="gml:ProjectedCRSRefType"
substitutionGroup="gml:generalDerivedCRSRef"/>
 <!-- ============================================================ -->
 <xsd:complexType name="ProjectedCRSRefType">
  <xsd:annotation>
```

```
      <xsd:documentation>Association to a Projected Coordinate Reference System, either referencing or
containing the definition of that Reference System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:GeneralDerivedCRSRefType">
        <xsd:sequence>
          <xsd:element ref="gml:ProjectedCRS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="DerivedCRS" type="gml:DerivedCRSType"
substitutionGroup="gml:_GeneralDerivedCRS"/>
  <!-- ============================================================= -->
  <xsd:complexType name="DerivedCRSType">
    <xsd:annotation>
      <xsd:documentation>A coordinate reference system that is defined by its coordinate conversion from
another coordinate reference system but is not a projected coordinate reference system. This category
includes coordinate reference systems derived from a projected coordinate reference system.
</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractGeneralDerivedCRSType">
        <xsd:sequence>
          <xsd:element name="usesCS" type="gml:CoordinateSystemRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the coordinate system used by this derived CRS.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
        <xsd:attribute ref="gml:derivedCRSType" use="required">
          <xsd:annotation>
            <xsd:documentation>Type of this derived coordinate reference system. </xsd:documentation>
          </xsd:annotation>
        </xsd:attribute>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
  <xsd:element name="derivedCRSRef" type="gml:DerivedCRSRefType"
substitutionGroup="gml:generalDerivedCRSRef"/>
  <!-- ============================================================= -->
  <xsd:complexType name="DerivedCRSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a non-projected derived Coordinate Reference System, either
referencing or containing the definition of that Reference System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:GeneralDerivedCRSRefType">
        <xsd:sequence>
          <xsd:element ref="gml:DerivedCRS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================= -->
```

```
  <xsd:attribute name="derivedCRSType" type="gml:DerivedCRSTypeType"/>
  <!-- ============================================================== -->
  <xsd:simpleType name="DerivedCRSTypeType">
    <xsd:annotation>
      <xsd:documentation>Type of a derived coordinate reference system. </xsd:documentation>
    </xsd:annotation>
    <xsd:restriction base="string">
      <xsd:enumeration value="engineering">
        <xsd:annotation>
          <xsd:documentation>A contextually local coordinate reference system; which can be divided into two
broad categories:
- earth-fixed systems applied to engineering activities on or near the surface of the earth;
- CRSs on moving platforms such as road vehicles, vessels, aircraft, or spacecraft. </xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="image">
        <xsd:annotation>
          <xsd:documentation>An engineering coordinate reference system applied to locations in images.
</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="vertical">
        <xsd:annotation>
          <xsd:documentation>A coordinate reference system used for recording of heights or depths. Vertical
CRSs make use of the direction of gravity to define the concept of height or depth, but the relationship with
gravity may not be straightforward. </xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
      <xsd:enumeration value="temporal">
        <xsd:annotation>
          <xsd:documentation>A coordinate reference system used for the recording of time.
</xsd:documentation>
        </xsd:annotation>
      </xsd:enumeration>
    </xsd:restriction>
  </xsd:simpleType>
  <!-- ============================================================== -->
  <xsd:element name="EngineeringCRS" type="gml:EngineeringCRSType"
substitutionGroup="gml:_CoordinateReferenceSystem"/>
  <!-- ============================================================== -->
  <xsd:complexType name="EngineeringCRSType">
    <xsd:annotation>
      <xsd:documentation>A contextually local coordinate reference system; which can be divided into two
broad categories:
- earth-fixed systems applied to engineering activities on or near the surface of the earth;
- CRSs on moving platforms such as road vehicles, vessels, aircraft, or spacecraft.
For further information, see OGC Abstract Specification Topic 2. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateReferenceSystemType">
        <xsd:sequence>
          <xsd:element name="usesCS" type="gml:CoordinateSystemRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the coordinate system used by this engineering CRS.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="usesDatum" type="gml:EngineeringDatumRefType">
            <xsd:annotation>
```

```
            <xsd:documentation>Association to the engineering datum used by this engineering CRS.
</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================= -->
 <xsd:element name="engineeringCRSRef" type="gml:EngineeringCRSRefType"
substitutionGroup="gml:coordinateReferenceSystemRef"/>
 <!-- ============================================================= -->
 <xsd:complexType name="EngineeringCRSRefType">
  <xsd:annotation>
    <xsd:documentation>Association to an Engineering Coordinate Reference System, either referencing or
containing the definition of that Reference System. </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:restriction base="gml:CoordinateReferenceSystemRefType">
      <xsd:sequence>
        <xsd:element ref="gml:EngineeringCRS" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
    </xsd:restriction>
  </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================= -->
 <xsd:element name="ImageCRS" type="gml:ImageCRSType"
substitutionGroup="gml:_CoordinateReferenceSystem"/>
 <!-- ============================================================= -->
 <xsd:complexType name="ImageCRSType">
  <xsd:annotation>
    <xsd:documentation>An engineering coordinate reference system applied to locations in images. Image
coordinate reference systems are treated as a separate sub-type because a separate user community exists
for images with its own terms of reference. </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="gml:AbstractCoordinateReferenceSystemType">
      <xsd:sequence>
        <xsd:choice>
          <xsd:element name="usesCartesianCS" type="gml:CartesianCSRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the Cartesian coordinate system used by this image
CRS.</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="usesObliqueCartesianCS" type="gml:ObliqueCartesianCSRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the oblique Cartesian coordinate system used by this image
CRS.</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:choice>
        <xsd:element name="usesDatum" type="gml:ImageDatumRefType">
          <xsd:annotation>
            <xsd:documentation>Association to the image datum used by this image CRS.
</xsd:documentation>
          </xsd:annotation>
        </xsd:element>
      </xsd:sequence>
```

```
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="imageCRSRef" type="gml:ImageCRSRefType"
substitutionGroup="gml:coordinateReferenceSystemRef"/>
  <!-- ============================================================== -->
  <xsd:complexType name="ImageCRSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to an Image Coordinate Reference System, either referencing or
containing the definition of that Reference System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateReferenceSystemRefType">
        <xsd:sequence>
          <xsd:element ref="gml:ImageCRS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="TemporalCRS" type="gml:TemporalCRSType"
substitutionGroup="gml:_CoordinateReferenceSystem"/>
  <!-- ============================================================== -->
  <xsd:complexType name="TemporalCRSType">
    <xsd:annotation>
      <xsd:documentation>A coordinate reference system used for the recording of time.
</xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractCoordinateReferenceSystemType">
        <xsd:sequence>
          <xsd:element name="usesCS" type="gml:TemporalCSRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the temporal coordinate system used by this temporal CRS.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
          <xsd:element name="usesDatum" type="gml:TemporalDatumRefType">
            <xsd:annotation>
              <xsd:documentation>Association to the temporal datum used by this temporal CRS.
</xsd:documentation>
            </xsd:annotation>
          </xsd:element>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
  <!-- ============================================================== -->
  <xsd:element name="temporalCRSRef" type="gml:TemporalCRSRefType"
substitutionGroup="gml:coordinateReferenceSystemRef"/>
  <!-- ============================================================== -->
  <xsd:complexType name="TemporalCRSRefType">
    <xsd:annotation>
      <xsd:documentation>Association to a Temporal Coordinate Reference System, either referencing or
containing the definition of that Reference System. </xsd:documentation>
    </xsd:annotation>
    <xsd:complexContent>
      <xsd:restriction base="gml:CoordinateReferenceSystemRefType">
```

```
        <xsd:sequence>
          <xsd:element ref="gml:TemporalCRS" minOccurs="0"/>
        </xsd:sequence>
        <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
      </xsd:restriction>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

Annex E
(Informative)

Tutorial

E.1 General

This clause provides an introductory tutorial on GML 3.0. It assumes some familiarity with XML but does not assume any prior experience with GML.

Geography Markup Language (GML) was created to provide an XML-based means for:

- The encoding geographic information for sharing and exchange over the Internet.
- The expression of geographic vocabularies for different domains of discourse.
- The expression of message components for geographic web-based services.

GML 1.0 was passed as a recommendation paper of the OpenGIS Consortium (OGC) in May 2000. It was used primarily as a test vehicle for the exploration of encoding concepts for geographic data and only prototype commercial products were developed.

GML 2.0 was passed as an adopted specification of the OGC in March 2001. It was based on XML Schema rather than the DTD/RDFS used in GML 1.0. Many vendors have developed products and components based on GML 2. and several large datasets are now being delivered in GML2. GML 2. is not backwards compatible with GML 1.

The encoding model of GML 3. is the same as that of GML 2. and GML 3. is backwards compatible with GML 2. GML 3. greatly extends the set of built-elements that GML provides for the geographic application developer.

GML 3.0 provides for the encoding of several types of geographic objects including:

- Geographic features including their geometry, topology and temporal evolution.
- Geographic coverages including their geometry and attribute values.
- Geographic observations (e.g. as in measurement systems)
- Coordinate reference systems
- Abstract values including numerical quantities with units of measure, and observations based on counting, categorization and Boolean decisions.

The GML concept of feature is based on ISO 19109 and the OGC Abstract Specification. A simplified UML model based on ISO 19109 is shown in Figure E.1-1. This shows that a feature is defined by a collection of properties. Some of the properties correspond to UML attributes and some to UML associations between classes.

A coordinate reference system dictionary in GML is a collection of GML Coordinate Reference System definitions.

A units of measure dictionary in GML is a collection of GML Units of Measure definitions.

E.2 Geographic Datasets

E.2.1 Objects in GML

GML models resources that are required in dealing with the geographic information. These include in particular:

- Features (including coverages and observations).
- Coordinate Reference Systems
- Units of Measure
- Values (as values of feature properties)
- Topology and Geometry (as values of feature properties)
- Temporal (as values of feature properties)

In GML, an object is always modeled as a global element with an XML Schema associated content model, the content model defining the properties that describe the object. In a GML instance the child elements of the object instance are always properties of the object instance, and properties can only be expressed as child elements. The values of the properties are then one of the valid GML object types or are an arbitrary XML Schema complex type. It is incorrect to represent GML objects as elements who content models do not correspond to the requirements of the listed GML objects above. One cannot make a feature, topology etc. except by following the rules defined for application schemas in Chpater 7. of this specification.

E.2.2 Features

A geographic dataset in GML is a GML feature collection, that is a particular GML-defined structure that contains a set of GML feature instances. Feature instances represent real world entities such as persons, rivers, roads, countries, cities, vehicles, and buildings. Note that in GML 3.0 a GML Dataset can also include coverages (coverage is a kind of feature), and observations. Coverages and Observations are discussed in D2.2 and D

Each feature instance is represented by an XML element, whose name is the semantic type of the instance (e.g. road). The semantic type is the class or category of real world object to which the instance belongs in some domain of discourse.

Each feature instance has an identifier that is unique within the scope in which the feature is defined.

Each feature is described by a set of properties, all of which are XML child elements of the feature instance element. We thus might describe a Bridge (semantic type) instance

```
Example 1:

<abc:Bridge gml:id = “b1”>
  <abc:span>50</abc:span>
  <abc:heightAtCenterSpan>51</abc:heightAtCenterSpan>
  <abc:constructed>10-10-1996</abc:constructed>
</abc:Bridge>
```

as shown in the following example.

The example is a Bridge feature instance. The element name “Bridge” is the semantic type of the feature instance. We also refer to this as the geographic type or feature type of the feature instance.

The elements <abc:span>, <abc:heightAtCenterSpan>, <abc:constructed> are the properties of the feature. All of the properties in this example have simple types, namely integers or decimals (span, height) and date (constructed).

A GML 3.0 feature may have complex valued properties including in particular, geometries, other features, complex values (e.g. measured quantities with units), temporal values, and topology elements.

A GML 3.0 feature with a geometry-valued property is illustrated in the next example.

```
Example 2:

<abc:Bridge gml:id = “b1”>
  <abc:span>50</abc:span>
  <abc:heightAtCenterSpan>51</abc:heightAtCenterSpan>
  <abc:constructed>10-10-1996</abc:constructed>
  <gml:location>
    <gml:Point srsName = “#myRefSys”>
      <gml:coordinates>12435.2, 6756.34</gml:coordinates>
    </gml:Point>
  </gml:location>
</abc:Bridge>
```

A GML 3.0 feature with a temporal-valued property is shown in the next example:

```
Example 3:

<gml:FeatureCollection>
  <gml:boundedBy>
    <gml:Envelope>
      <gml:coordinates>0,0 10,10</gml:coordinates>
    </gml:Envelope>
  </gml:boundedBy>
  <gml:featureMember>
    <abc:Bridge gml:id="b1">
      <gml:location>
        <gml:Point srsName="#myRefSys">
          <gml:coordinates>12435.2, 6756.34</gml:coordinates>
        </gml:Point>
      </gml:location>
      <span>50</span>
      <heightAtCenterSpan>51</heightAtCenterSpan>
      <constructed>10-10-1996</constructed>
      <lifetime>P200Y</lifetime>
    </abc:Bridge>
  </gml:featureMember>
</gml:FeatureCollection>
```

```
Example 4:

<abc:Bridge gml:id = “b1”>
  <abc:span>50</abc:span>
  <abc:heightAtCenterSpan>51</abc:heightAtCenterSpan>
  <abc:constructed>10-10-1996</abc:constructed>
  <abc:location>
    <gml:Node id=”n1”/>
  </gml:location>
</abc:Bridge>
```

A GML 3.0 feature with a topology-valued property is shown the in next example:

A GML 3.0 feature with a measure-valued property is shown in the next example.

```
Example 5:

<abc:Bridge gml:id = “b1”>
  <abc:span uom=”#meters”>50</abc:span>
  <abc:heightAtCenterSpan uom =”#meters”>51</abc:heightAtCenterSpan>
  <abc:constructed>10-10-1996</abc:constructed>
</abc:Bridge>
```

The structure of any GML feature instance is governed by an associated GML

Application Schema. For the GML feature instance to be valid (i.e. be legal GML) it must be XML Schema valid with respect to a proper GML Application Schema.

A GML Application Schema is an XML Schema that conforms to some specific GML rules (see Clause 7.1 and 7.2 for the rule for writing application schemas for features, feature collections and coverages).

Each GML Application Schema defines a set of semantic types (features, coverages or feature collections) by specifying their name (e.g. Bridge, Road) and their properties (e.g. span, number of lanes).

Each GML Application Schema must import the required core GML Schemas described in Clause 6 as described in Clause 7. These imports are summarized in table E.2.1-1.

Note that in GML 3.0 a GML Dataset can also include coverages (coverage is a kind of feature), and observations. Coverages and Observations are discussed in D2.3 and D2.4.

GML Dataset Objects	**Schema to Import**	**Comments**
Features, Feature Collections	feature.xsd	feature.xsd includes geometry.xsd. topology.xsd includes geometry.xsd
Coverages	coverage.xsd	coverage.xsd includes feature.xsd
Observations	observation.xsd	observation.xsd includes feature.xsd, temporal.xsd and direction.xsd.

Figure E.2.1-1 GML Datasets

E.2.3 Coverages

A coverage data set contains one or more coverages and is a special case of a Feature Collection in which the individual members of the collection are coverages. Each coverage describes the distribution of some property or properties over some spatial region (in future this will be spatial-temporal). The spatial region can be 0, 1, 2, or 3 dimensional as shown in the examples in Table E.2.2-1.

Coverage Data	**Spatial Dimension**	**Coverage Type**
Soil samples	0	Multi-point Coverage

Digital Elevation Model	0	Multi-point Coverage
Rock bore samples	1	Multi-curve Coverage
Traffic queue lengths	1	Multi-curve Coverage
Soils distribution	2	Multi-polygon Coverage
Birth rate distribution by county	2	Multi-polygon Coverage
Optical satellite image	2	Rectified Grid Coverage
Digital Elevation Model	2	Multi-polygon Coverage

Table E.2.2-1 Coverages and Underlying Geometry

GML provides a basic coverage structure (see gml:_Coverage) and a set of specific coverage types based on the underlying geometry model (for details see Clause 7.12). Some example coverages will help to illustrate the basic concepts.

Example 1: Soil Distribution

The following example shows an application schema fragment for a single soil distribution coverage. This would be applicable to set of soil samples obtained at a discrete set of observation points.

```
<element name="SoilData" type="app:SoilDataType"/>

<complexType name="SoilDataType">
  <complexContent>
    <extension base="gml:MultiPointCoverageType"/>
  </complexContent>
</complexType>

<element name="SoilType" type="app:SoilTypeType" substitutionGroup="gml:_Value"/>

<complexType name="SoilTypeType">
  <simpleContent>
    <restriction base="gml:CategoryType">
      <attribute name="codespace" type="anyURI" use="optional"/>
    </restriction>
  </simpleContent>
</complexType>
```

Note that while the geometry type of the domain can be determined from the SoilDataType content model, the same cannot be said for the values in the range. The above example is a valid model for data values encoded in a data block (like a CSV), in inline XML, and in a binary file.

The codespace attribute in the SoilTypeType points to a dictionary of soil types. This is done rather than using an XML Schema enumerated list because the number of entries may be very large.

Example 2. Temperature and Pressure Distribution

This example shows the distribution of temperature and pressure measurements.

```
<element name="AverageTempPressure" type="app:AverageTempPressureType"/>

 <complexType name="AverageTempPressureType">
  <complexContent>
   <extension base="gml:MultiPointCoverageType"/>
  </complexContent>
 </complexType>

 <element name="Temperature" type="app:TemperatureType"
                        substitutionGroup="gml:_Value"/>

 <complexType name="TemperatureType">
  <simpleContent>
   <restriction base="gml:QuantityType">
    <attribute name="observable" type="anyURI" use="optional"/>
   </restriction>
  </simpleContent>
 </complexType>

 <element name="TemperatureReading" type="app:TemperatureReadingType"
                        substitutionGroup="gml:_Value"/>

 <complexType name="TemperatureReadingType">
  <simpleContent>
   <restriction base="gml:QuantityType">
    <attribute name="observable" type="anyURI" use="optional"/>
   </restriction>
  </simpleContent>
 </complexType>


 <element name="Pressure" type="app:PressureType" substitutionGroup="gml:_Value"/>

 <complexType name="PressureType">
  <simpleContent>
   <restriction base="gml:QuantityType">
    <attribute name="observable" type="anyURI" use="optional"/>
   </restriction>
  </simpleContent>
 </complexType>


 <element name="PressureReading" type="app:PressureReadingType" substitutionGroup="gml:_Value"/>

 <complexType name="PressureReadingType">
  <simpleContent>
   <restriction base="gml:QuantityType">
    <attribute name="observable" type="anyURI" use="optional"/>
   </restriction>
  </simpleContent>
 </complexType>
```

Note that this example contains two parameters, namely average Temperature and Pressure measured at a discrete set of points (hence a MultiPointCoverage). Note further that as in the previous example, one must examine the GML instance in order to know the structure of the value set (range).

Using the DataBlock encoding for the values we have:

```
<AverageTempPressure>
 <gml:domainSet>
  <gml:MultiPoint>
   <gml:pointMember>
    <gml:Point gid="p1">
     <gml:coordinates>1 1</gml:coordinates>
    </gml:Point>
   </gml:pointMember>
   <gml:pointMember>
    <gml:Point gid="p6">
     <gml:coordinates>2 2</gml:coordinates>
    </gml:Point>
   </gml:pointMember>
   <gml:pointMember>
    <gml:Point gid="p11">
     <gml:coordinates>3 3</gml:coordinates>
    </gml:Point>
   </gml:pointMember>
   <gml:pointMember>
    <gml:Point gid="p16">
     <gml:coordinates>4 4</gml:coordinates>
    </gml:Point>
   </gml:pointMember>
  </gml:MultiPoint>
 </gml:domainSet>
 <gml:rangeSet>
  <gml:DataBlock>
   <gml:tupleList>3,101.1 17,101.7 37,102.2 59,102.6</gml:tupleList>
   <gml:rangeParameters>
    <gml:CompositeValue>
     <gml:valueComponent>
      <Temperature/>
     </gml:valueComponent>
     <gml:valueComponent>
      <Pressure/>
     </gml:valueComponent>
    </gml:CompositeValue>
   </gml:rangeParameters>
  </gml:DataBlock>
 </gml:rangeSet>
</AverageTempPressure>
```

Using the binary File encoding for values we have:

```xml
<AverageTempPressure>
  <gml:domainSet>
    <gml:MultiPoint>
      <gml:pointMember>
        <gml:Point gid="p1">
          <gml:coordinates>1 1</gml:coordinates>
        </gml:Point>
      </gml:pointMember>
      <gml:pointMember>
        <gml:Point gid="p6">
          <gml:coordinates>2 2</gml:coordinates>
        </gml:Point>
      </gml:pointMember>
      <gml:pointMember>
        <gml:Point gid="p11">
          <gml:coordinates>3 3</gml:coordinates>
        </gml:Point>
      </gml:pointMember>
      <gml:pointMember>
        <gml:Point gid="p16">
          <gml:coordinates>4 4</gml:coordinates>
        </gml:Point>
      </gml:pointMember>
    </gml:MultiPoint>
  </gml:domainSet>
  <gml:rangeSet>
    <gml:File>
      <gml:fileName>.../temperature.dat</gml:fileName>
      <gml:fileStructure>Record Interleaved</gml:fileStructure>
      <gml:rangeParameters>
        <gml:CompositeValue>
          <gml:valueComponent>
            <Temperature/>
          </gml:valueComponent>
          <gml:valueComponent>
            <Pressure/>
          </gml:valueComponent>
        </gml:CompositeValue>
      </gml:rangeParameters>
    </gml:File>
  </gml:rangeSet>
</AverageTempPressure>
```

Using the in-line XML encoding for values we have:

```xml
<AverageTempPressure>
 <gml:domainSet>
  <gml:MultiPoint>
   <gml:pointMember>
    <gml:Point>
     <gml:pos>1 1 </gml:pos>
    </gml:Point>
   </gml:pointMember>
   <gml:pointMember>
    <gml:Point>
     <gml:pos>2 2 </gml:pos>
    </gml:Point>
   </gml:pointMember>
   <gml:pointMember>
    <gml:Point>
     <gml:pos>3 3 </gml:pos>
    </gml:Point>
   </gml:pointMember>
   <gml:pointMember>
    <gml:Point>
     <gml:pos>4 4 </gml:pos>
    </gml:Point>
   </gml:pointMember>
  </gml:MultiPoint>
 </gml:domainSet>
 <gml:rangeSet>
  <gml:ValueCollection>
   <gml:valueMembers>
    <TemperatureReading>3</TemperatureReading>
    <TemperatureReading>5</TemperatureReading>
    <TemperatureReading>7</TemperatureReading>
    <TemperatureReading>11</TemperatureReading>
    <TemperatureReading>13</TemperatureReading>
    <TemperatureReading>17</TemperatureReading>
    <TemperatureReading>19</TemperatureReading>
    <TemperatureReading>23</TemperatureReading>
    <TemperatureReading>29</TemperatureReading>
    <TemperatureReading>31</TemperatureReading>
    <TemperatureReading>37</TemperatureReading>
    <TemperatureReading>41</TemperatureReading>
    <TemperatureReading>43</TemperatureReading>
    <TemperatureReading>47</TemperatureReading>
    <TemperatureReading>53</TemperatureReading>
    <TemperatureReading>59</TemperatureReading>
   </gml:valueMembers>
  </gml:ValueCollection>
 </gml:rangeSet>
</AverageTempPressure>
```


E.2.4 Observations

Observations model the act of observing whether this is a causal observation such as photograph taken by a tourist or a precise measurement acquired by sophisticated instrumentation. Observations contain the record of the act of observing in terms of a set of observation values that are the “resultOf” the observation.

A collection of Observations is a GML Dataset and is a special kind of feature collection in which each feature member is an observation. Note that we can readily construct a coverage from a collection of observations, but that a coverage is inherently more general and may not result from a set of observations but rather from a numerical or analytical model.

An observation contains the minimal metadata about the act of observing including its location and the time of occurrence.

Example 1: Tourist Photo of a Building or Monument

Uses gml:ObservationType as the content model for CityPhoto.

Creates MonumentType based on gml:AbstractFeatureType and uses it as the content model for Monument, which can be used as the value of the gml:subject property.

```
<schema targetNamespace="http://www.opengis.net/app" xmlns:app="http://www.opengis.net/app"
xmlns:gml="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified" version="2.06">

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

  <element name="CityPhoto" type="gml:ObservationType"/>
  <element name="Monument" type="app:MonumentType"
                      substitutionGroup="gml:_Feature"/>

  <complexType name="MonumentType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence/>
      </extension>
    </complexContent>
  </complexType>

</schema>
```

A sample instance document fragment for this schema is:

```
<CityPhoto>
  <gml:timeStamp>
    <gml:TimeInstant>
      <gml:timePosition>2001-07-20T19:19:30</gml:timePosition>
    </gml:TimeInstant>
  </gml:timeStamp>
  <gml:subject>
    <Monument fid="m24">
      <gml:description>A hidden tomb</gml:description>
      <gml:name>Tomb of the Unknown Soldier</gml:name>
      <gml:location>
        <gml:Point srsName = "http://www.opengis.org/crsPortal#p4326">
          <gml:coordinates>25.45, 120.45</gml:coordinates>
        </gml:Point>
      </gml:location>
    </Monument>
  </gml:subject>
  <gml:resultOf xlink:href="pict0034.jpg"/>
</CityPhoto>
```

Note that this example describes a photograph of a Monument acquired at a particular time and location. The Monument is specified as the subject of the Photo.

Example 2: Measurement on a sample

In this example we use both the GML observation and value schemas. These are included via a stub schema observationAndValue.xsd which is defined as follows:

```
<xs:schema targetNamespace="http://www.opengis.net/gml" xmlns:gml="http://www.opengis.net/gml"
xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="1.0">
  <xs:annotation>
    <xs:documentation>
    observationAndValue.xsd
    Utility schema which simply includes both observation and value</xs:documentation>
  </xs:annotation>
  <xs:include schemaLocation="../gml/base/observation.xsd"/>
  <xs:include schemaLocation="../gml/base/valueObjects.xsd"/>
</xs:schema>
```

In the application schema, we first declare two properties sample and density of suitable type and assign these elements to the substitution groups of gml:target and gml:resultOf.

Then a specialised form of Observation is defined, as a restriction of the gml:ObservationType in which sample and density substitute for their substitution group heads:

```
<schema targetNamespace="http://cg-namespaces.arrc.csiro.au/my" xmlns:my="http://cg-
namespaces.arrc.csiro.au/my" xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <import namespace="http://www.opengis.net/gml" schemaLocation="./observationAndValue.xsd"/>

  <element name="sample" type="gml:FeaturePropertyType" substitutionGroup="gml:target"/>
```

```
  <element name="density" type="gml:QuantityPropertyType" substitutionGroup="gml:resultOf"/>

  <element name="DensityTest" type="my:DensityTestType"/>

  <complexType name="DensityTestType">
    <complexContent>
      <restriction base="gml:ObservationType">
        <sequence>
          <element ref="gml:metaDataProperty" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:description" minOccurs="0"/>
          <element ref="gml:name" minOccurs="0" maxOccurs="unbounded"/>
          <element ref="gml:boundedBy" minOccurs="0"/>
          <element ref="gml:location" minOccurs="0"/>
          <element ref="gml:timeStamp"/>
          <element ref="gml:using" minOccurs="0"/>
          <element ref="my:sample"/>
          <element ref="my:density"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>

</schema>
```

```
<my:DensityTest xmlns:my="http://cg-namespaces.arrc.csiro.au/my"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://cg-
namespaces.arrc.csiro.au/my
C:\DOCUME~1\cox075.NEXUS\MYDOCU~1\onto\xmml-dev\src\fun\test.xsd">

  <gml:timeStamp>
    <gml:TimeInstant><gml:timePosition>2002-12-02T12:20:00</gml:timePosition></gml:TimeInstant>
  </gml:timeStamp>

  <my:sample xlink:href="http://my.big.org/samples#876e"/>

  <my:density>
    <gml:Quantity uom="http://my.big.org/units#kg_T">2560.</gml:Quantity>
  </my:density>

</my:DensityTest>
```

An instance example which is valid according to this schema is as follows:

E.3 Software for GML

E.3.1 General

Software that processes GML data needs to have a mechanism to read, parse and interpret the data in accordance with an associated GML Application Schema. This process is common to interpreting any data based on XML Schema and is referred to as XML Schema Parsing. Such software needs to take note of the following characteristic of

GML in order to be able to detect the presence of GML objects in the data stream or in a GML application schema. These characteristics follow from the Conformance Rules discussed in Chapter X.

1. Properties and GML objects (features, coverages, geometries etc.) are always elements. An element is known to be a GML object by the fact that its content model derives from gml:AbstractGMLType. This can be determined by processing the associated GML Application Schema.
2. An element is known to be a particular kind of GML object (e.g. feature, geometry) by the fact that its content model derives from the appropriate GML abstract type (e.g. gml:AbstractFeatureType. This can be determined by processing the associated GML Application Schema.
3. An element is known to be the property of a GML object if it is a child element of that object (element).

Software developers should note that GML data is inherently hierarchical in nature (tree structured). In many cases GML can be mapped into flatter structures using GML processing software.

Software developers should also be aware that GML Feature Collections are features and that the member features of a collection may themselves also be features. Software for handling GML may need to contain iterators to deal with this fact.

Software developers should be aware that GML geometries such as Polygons may have an unspecified number of children (e.g. interior or exterior boundaries) and processing of such structure may require geometry iterators.

Annex F
(Informative)

Examples

F.1 General

This annex presents a number of examples to illustrate the construction of application schemas that employ various base GML schemas. All examples in this annex have been validated using the following suite of parsers:

- XSV (v1.4.1)
- Altova XML Spy (v4.4)
- Microsoft MSXML (v4.0)
- Topologi SchematronValidator

F.2 GML Version 2.1.2 Examples Demonstrating Backward Compatibility

The following examples were included in the GML Version 2.1.2 Specification. To accommodate changes in GML version 3, the example schemas have some minor changes, such as replacing gml:FeatureAssociationType with gml:FeaturePropertyType and gml:fid with gml:id. There have been no changes to the instance documents.

F.2.1 The Cambridge Example

This example has a single feature collection of type CityModel and contains two features using a containment relationship called 'cityMember'. See listing F.2.1.2 below. The feature collection has a string property called dateCreated with the value '2000-11' and a geometric property called boundedBy with a 'Box' value. The Box geometry (which represents the 'bounding box' of the feature collection) is expressed in the SRS identified by the value of the srsName attribute: this URI reference points to a fragment in a remote XML document that contains information about the reference system.

The first feature member is an instance of RiverType with the name "Cam" and description "The river that runs through Cambridge"; it has a geometric property called centerLineOf with a LineString value. The LineString geometry is expressed in the same SRS used by the bounding box.

The second feature member is an instance of RoadType with description "M11". It has a string property called classification with value "motorway" and an integer property called number with value "11". The road has a geometric property called linearGeometry with a LineString value; this LineString geometry is also expressed in the same SRS used by the bounding box.

The first feature member uses only standard property names defined by GML, whereas the second feature member uses application-specific property names, demonstrating how GML may be used in a custom application model.

Figure E2.1.1 is a UML diagram for the Cambridge example. As shown, allowable city members must be Road or River instances; a Mountain instance is not a valid member of the feature collection.

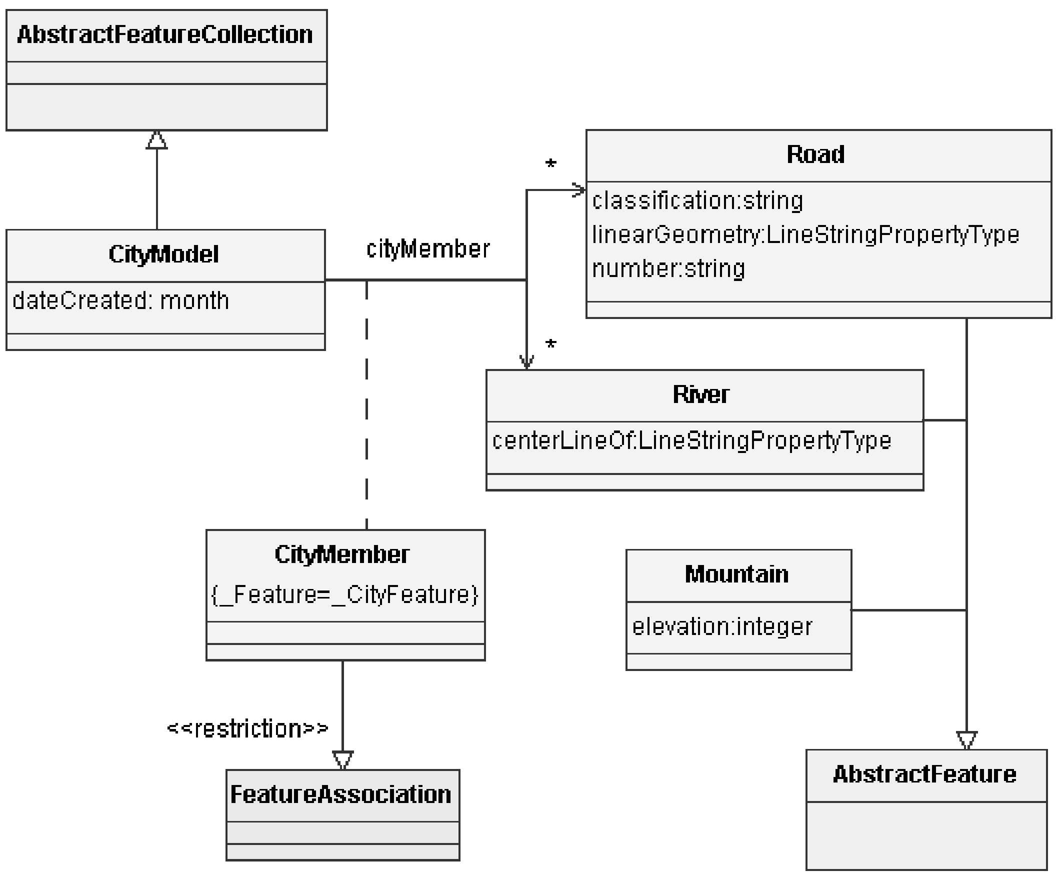


Figure F.2.1.1: UML diagram for the Cambridge example

Listing F.2.1.1 is a custom city schema for the Cambridge example. The explicit reference to "city.xsd" in the root element of the instance document in Listing F.2.1.2 (i.e. the value of the xsi:schemaLocation attribute) is not required, but in this case it provides a hint to the validating parser regarding the location of a relevant schema document.

Listing F.2.1.1: city.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/examples" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:ex="http://www.opengis.net/examples" elementFormDefault="qualified" version="3.0">
 <annotation>
  <appinfo>city.xsd v3-0 2002-07</appinfo>
  <documentation xml:lang="en">
   GML schema for the GML2 Cambridge example.  Changes for GML3 include import of xlink namespace, change to
CityMemberType for GML3
  </documentation>
 </annotation>
 <!-- import constructs from the GML feature, geometry, measures, units, gmlBase, and baseTypes schemas -->
 <import namespace="http://www.opengis.net/gml" schemaLocation="../../base/feature.xsd"/>
 <!-- import xlink namespace -->
 <import namespace="http://www.w3.org/1999/xlink" />
 <!-- ==============================================================
    global element declarations
 ============================================================== -->
 <element name="CityModel" type="ex:CityModelType"/>
 <element name="cityMember" type="ex:CityMemberType" substitutionGroup="gml:featureMember"/>
 <element name="Road" type="ex:RoadType" substitutionGroup="ex:_CityFeature"/>
 <element name="River" type="ex:RiverType" substitutionGroup="ex:_CityFeature"/>
 <element name="Mountain" type="ex:MountainType" substitutionGroup="gml:_Feature"/>
 <!-- a label for restricting membership in the CityModel collection -->
 <element name="_CityFeature" type="gml:AbstractFeatureType" abstract="true" substitutionGroup="gml:_Feature"/>
 <!-- ==============================================================
    type definitions for city model
 ============================================================== -->
 <complexType name="CityModelType">
  <complexContent>
   <extension base="gml:AbstractFeatureCollectionType">
    <sequence>
     <element name="dateCreated" type="gYearMonth"/>
    </sequence>
   </extension>
  </complexContent>
 </complexType>
 <complexType name="CityMemberType">
  <annotation>
   <documentation>
    A cityMember is restricted to those features (or feature
```



```
      collections) that are declared equivalent to ex:_CityFeature.
      In GML2 this type was based on gml:FeatureAssociationType.
    </documentation>
  </annotation>
  <complexContent>
    <restriction base="gml:FeaturePropertyType">
      <sequence> <!-- was  minOccurs="0" -->
        <element ref="ex:_CityFeature"  minOccurs="0"/>
      </sequence>
      <!-- was <attributeGroup ref="xlink:simpleLink"/>
      <attribute ref="gml:remoteSchema" use="optional"/> -->
<attributeGroup ref="gml:AssociationAttributeGroup"/>
    </restriction>
  </complexContent>
</complexType>
<complexType name="RiverType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:centerLineOf"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<complexType name="RoadType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="linearGeometry" type="gml:LineStringPropertyType"/>
        <element name="classification" type="string"/>
        <element name="number" type="string"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- this is just here to demonstrate feature member restriction -->
<complexType name="MountainType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
```

```
      <element name="elevation" type="integer"/>
     </sequence>
    </extension>
   </complexContent>
  </complexType>
</schema>
```

Note that the application schema targets the namespace bound to the prefix 'ex' ; it imports the GML feature and geometry constructs from the 'gml' namespace. The <boundedBy> element is defined in the Feature schema; the <name> and <description> elements are also defined there. The <CityModel> element is an instance of the user-defined ex:CityModelType type that is derived by extension from gml:AbstractFeatureCollectionType. The types ex:RiverType and ex:RoadType are both derived by extension from gml:AbstractFeatureType, which is defined in the GML Feature schema; these derivations assure that the application schema conforms with the GML implementation specification of the OGC Simple Feature model.

Listing F.2.1.2 is a simple schema-valid instance document that conforms to city.xsd. A few words of explanation about the <Mountain> feature are in order! If this particular cityMember is uncommented in Listing 7.2, it will raise a validation error because even though the mountain is a well-formed GML feature, it is not recognized as a valid city feature. Note that in city.xsd the <Road> and <River> features are declared equivalent to ex:_CityFeature using the substitutionGroup attribute; this abstract element functions as a label that restricts membership in the <CityModel> feature collection--only features so labeled are allowable members, as defined by CityMemberType. This technique demonstrates the application of the "Feature Filter" discussed in sub-secton

FIXME (X.Y.Z was 5.2.7 in GML2 doc) that restricts membership in GML feature collections.

One <cityMember> element in Listing E2.1.2 functions as a simple link by employing several XLink attributes; in effect we have a pointer entitled "Trinity Lane". Any <featureMember> element may behave as a simple link that references a remote resource. The link can point to a document fragment using an XPointer scheme that identifies a location, point, or range in the target document [XPointer]. In this case the value of the href attribute for the remote member contains an HTTP query string that can retrieve the feature instance; the remoteSchema attribute points to a schema fragment that constrains the instance: namely, the complex type definition in city.xsd that bears the name "RoadType".

Listing F.2.1.2: cambridge.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<CityModel xmlns="http://www.opengis.net/examples" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/examples city.xsd">
 <gml:name>Cambridge</gml:name>
 <gml:boundedBy>
```

```
    <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      <gml:coordinates>0.0,0.0</gml: coordinates >
      <gml: coordinates >100.0,100.0</gml: coordinates >
    </gml:Box>
  </gml:boundedBy>
  <cityMember>
    <River>
      <gml:description>The river that runs through Cambridge.</gml:description>
      <gml:name>Cam</gml:name>
      <gml:centerLineOf>
        <gml:LineString srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
          <gml:coordinates>0,50</gml:coordinates>
          <gml:coordinates>70,60</gml:coordinates>
          <gml:coordinates>100,50</gml:coordinates>
        </gml:LineString>
      </gml:centerLineOf>
    </River>
  </cityMember>
  <cityMember>
    <Road>
      <gml:name>M11</gml:name>
      <linearGeometry>
        <gml:LineString srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
          <gml:coordinates>0,5.0</gml:coordinates>
          <gml:coordinates>20.6,10.7</gml:coordinates>
          <gml:coordinates>80.5,60.9</gml:coordinates>
        </gml:LineString>
      </linearGeometry>
      <classification>motorway</classification>
      <number>11</number>
    </Road>
  </cityMember>
  <cityMember xlink:type="simple" xlink:title="Trinity Lane" xlink:href="http://www.foo.net/cgi-
bin/wfs?FeatureID=C10239" gml:remoteSchema="city.xsd#xpointer(//complexType[@name='RoadType'])"/>
<!-- a mountain doesn't belong here! Uncomment this cityMember and see
      the parser complain!
  <cityMember>
    <Mountain>
      <gml:description>World's highest mountain is in Nepal!</gml:description>
      <gml:name>Everest</gml:name>
```

```
      <elevation>8850</elevation>
    </Mountain>
  </cityMember>
  -->
  <dateCreated>2000-11</dateCreated>
</CityModel>
```

F.2.2 The Schools Example

This example shown in listing F.2.2.2 below illustrates how GML can represent a hierarchy of feature collections. The root feature collection of type StateType contains two features collections (instances of SchoolDistrictType) using the pre-defined membership relationship 'featureMember'. The State collection also has a studentPopulation property. Each of the SchoolDistrict collections contains two School or College features using the membership relationship 'schoolMember'.

A SchoolDistrict feature has a string property called name and a polygon property called extentOf. A School feature has a string property called address and a point property called location. A College feature also has a string property called address plus a point property called pointProperty.

Figure F.2.2.1 is a UML diagram for the Schools example. The SchoolDistrictclass is associated with the State class via the featureMember relationship, and instances of the Schoolor College classes are members of the SchoolDistrict collection.

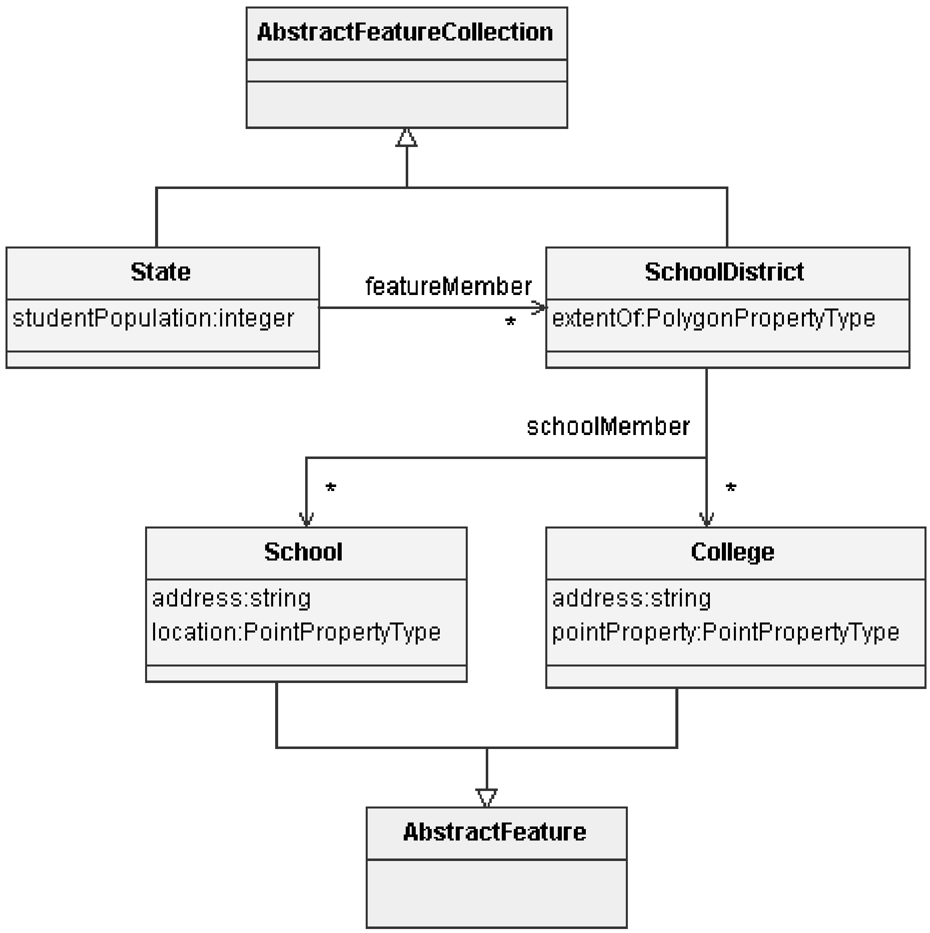


Figure F.2.2.1: UML diagram for the Schools example

Listing F.2.2.1 is an application schema for the Schools example. The purpose of this example is to demonstrate that feature collections may indeed contain other feature collections. To keep things fairly simple no attempt has been made to restrict membership in any of the collections; this means that a valid instance document could contain any GML feature within the <State> and <SchoolDistrict> collections, not just those pertaining to educational institutions. Sub-clause **FIXME (X.Y.Z was 5.2.7 in GML2 doc)** describes a design pattern for restricting collection membership.

Listing F.2.2.1: schools.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/examples" xmlns:ex="http://www.opengis.net/examples"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="3.0">
  <annotation>
    <appinfo>schools.xsd v3.0 2002-07</appinfo>
    <documentation xml:lang="en">
      GML schema for GML2 Schools example; change to schoolMember element for GML3
```



```xml
  </documentation>
 </annotation>
  <!-- import constructs from the GML feature, geometry, measures, units, gmlBase, and baseTypes schemas -->
 <import namespace="http://www.opengis.net/gml" schemaLocation="../../base/feature.xsd"/>
 <!-- ==============================================================
     global element declarations
 ================================================================== -->
 <element name="State" type="ex:StateType" substitutionGroup="gml:_FeatureCollection"/>
 <element name="SchoolDistrict" type="ex:SchoolDistrictType" substitutionGroup="gml:_FeatureCollection"/>
 <element name="schoolMember" type="gml:FeaturePropertyType" substitutionGroup="gml:featureMember">
  <annotation>
   <documentation>In GML2 this type was based on gml:FeatureAssociationType.</documentation>
  </annotation>
 </element>
 <element name="School" type="ex:SchoolType" substitutionGroup="gml:_Feature"/>
 <element name="College" type="ex:CollegeType" substitutionGroup="gml:_Feature"/>
 <element name="address" type="string"/>
 <!-- ==============================================================
     type definitions for state educational institutions
 ================================================================== -->
 <complexType name="StateType">
  <complexContent>
   <extension base="gml:AbstractFeatureCollectionType">
    <sequence>
     <element name="studentPopulation" type="integer"/>
    </sequence>
   </extension>
  </complexContent>
 </complexType>
 <complexType name="SchoolDistrictType">
  <complexContent>
   <extension base="gml:AbstractFeatureCollectionType">
    <sequence>
     <element ref="gml:extentOf"/>
    </sequence>
   </extension>
  </complexContent>
 </complexType>
 <complexType name="SchoolType">
  <complexContent>
```



```
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="ex:address"/>
          <element ref="gml:location"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <complexType name="CollegeType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="ex:address"/>
          <element ref="gml:pointProperty"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

A few interesting things are happening in this example. The root <State> element is an instance of ex:StateType, which is derived from the abstract gml:AbstractFeatureCollectionType defined in the GML Feature schema. One of the child elements, <SchoolDistrict>, is also a feature collection; in effect we have a feature collection containing a feature collection as one of its members. Listing E2.2.2 is a conforming instance document.

Listing F.2.2.2 schools.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<State xmlns="http://www.opengis.net/examples" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/examples schools.xsd">
  <gml:description>
    Educational institutions with student populations exceeding 500.
  </gml:description>
  <gml:name>School districts in the North Region.</gml:name>
  <gml:boundedBy>
    <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      <gml:coordinates>0,0</gml:coordinates>
      <gml:coordinates>50,50</gml:coordinates>
```



```
  </gml:Box>
</gml:boundedBy>
<gml:featureMember>
  <SchoolDistrict>
    <gml:name>District 28</gml:name>
    <gml:boundedBy>
      <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:coordinates>0,0</gml:coordinates>
        <gml:coordinates>50,40</gml:coordinates>
      </gml:Box>
    </gml:boundedBy>
    <schoolMember>
      <School>
        <gml:name>Alpha</gml:name>
        <address>100 Cypress Ave.</address>
        <gml:location>
          <gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
            <gml:coordinates>20.0,5.0</gml:coordinates>
          </gml:Point>
        </gml:location>
      </School>
    </schoolMember>
    <schoolMember>
      <School>
        <gml:name>Beta</gml:name>
        <address>1673 Balsam St.</address>
        <gml:location>
          <gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
            <gml:coordinates>40.0,5.0</gml:coordinates>
          </gml:Point>
        </gml:location>
      </School>
    </schoolMember>
    <gml:extentOf>
      <gml:Polygon srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:outerBoundaryIs>
          <gml:LinearRing>
            <gml:coordinates>0,0</gml:coordinates>
            <gml:coordinates>50,0</gml:coordinates>
            <gml:coordinates>50,40</gml:coordinates>
```

```
          <gml:coordinates>0,0</gml:coordinates>
        </gml:LinearRing>
      </gml:outerBoundaryIs>
    </gml:Polygon>
  </gml:extentOf>
</SchoolDistrict>
</gml:featureMember>
<gml:featureMember>
 <SchoolDistrict>
  <gml:name>District 32</gml:name>
  <gml:boundedBy>
   <gml:Box srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
    <gml:coordinates>0,0</gml:coordinates>
    <gml:coordinates>30,50</gml:coordinates>
   </gml:Box>
  </gml:boundedBy>
  <schoolMember>
   <School>
    <gml:name>Gamma</gml:name>
    <address>651 Sequoia Ave.</address>
    <gml:location>
     <gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      <gml:coordinates>5.0,20.0</gml:coordinates>
     </gml:Point>
    </gml:location>
   </School>
  </schoolMember>
  <schoolMember>
   <College>
    <gml:name>Delta</gml:name>
    <address>260 University Blvd.</address>
    <gml:pointProperty>
     <gml:Point srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
      <gml:coordinates>5.0,40.0</gml:coordinates>
     </gml:Point>
    </gml:pointProperty>
   </College>
  </schoolMember>
  <schoolMember xlink:type="simple" xlink:title="Epsilon High School" xlink:href="http:www.state.gov/schools/cgi-bin/wfs?schoolID=hs736" gml:remoteSchema="schools.xsd#xpointer(//complexType[@name='SchoolType'])"/>
```



```
      <gml:extentOf>
       <gml:Polygon srsName="http://www.opengis.net/gml/srs/epsg.xml#4326">
        <gml:outerBoundaryIs>
         <gml:LinearRing>
          <gml:coordinates>0,0</gml:coordinates>
          <gml:coordinates>40,50</gml:coordinates>
          <gml:coordinates>50,50</gml:coordinates>
          <gml:coordinates>0,0</gml:coordinates>
         </gml:LinearRing>
        </gml:outerBoundaryIs>
       </gml:Polygon>
      </gml:extentOf>
     </SchoolDistrict>
    </gml:featureMember>
    <studentPopulation>392620</studentPopulation>
   </State>
```

Note the use of <coordinates> elements to convey coordinate values; the XML parser constrains the number of tuples according to geometry type. For example, a <Point> element has exactly one coordinate tuple, and a <LinearRing> has at least four.

F.3 GML 3 Basic Examples

The examples in this subclause demonstrate the use of definitions retained from GML version 2 together with new definitions that have been added in GML version 3. Each example highlights a limited number of GML capabilities. Together, they show a representative sample of the new types and elements added in GML version 3.

F.3.1 The Road Infrastructure Example

The exampleRoad.xsd schema show in listing F.3.1.1 defines types used in two instance documents that demonstrate new geometry types and the use of XLinks to refer to and share geometry instances. The gml:curveProperty is the only new geometry property element explicitly referenced in exampleRoad.xsd.

Listing F.3.1.1: exampleRoad.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/examples" xmlns:ex="http://www.opengis.net/examples"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="0.0">
 <import namespace="http://www.opengis.net/gml" schemaLocation="../../base/feature.xsd"/>
 <element name="RoadInfrastructure" type="ex:RoadInfrastructureType" substitutionGroup="gml:FeatureCollection"/>
```

```
    <element name="Road" type="ex:RoadType" substitutionGroup="gml:_Feature"/>
    <element name="Bridge" type="ex:BridgeType" substitutionGroup="gml:_Feature"/>
    <complexType name="RoadInfrastructureType">
      <complexContent>
        <extension base="gml:FeatureCollectionType"/>
      </complexContent>
    </complexType>
    <complexType name="RoadType">
      <complexContent>
        <extension base="gml:AbstractFeatureType">
          <sequence>
            <element ref="gml:curveProperty"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
    <complexType name="BridgeType">
      <complexContent>
        <extension base="gml:AbstractFeatureType">
          <sequence>
            <element ref="gml:curveProperty"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
  </schema>
```

However, gml:curveProperty contains or points to a element substitutable for gml:_Curve, in this case a gml:LineString, that uses gml:pos elements of type gml:DirectPosition that are new with GML version 3. Listing F.3.1.2 provides a comparison between the representation of curves using direct positions and the gml:coordinates element carried over from GML version 2. It also demonstrates the use of XML IDs and XLinks to share point geometries.

All elements of types derived from gml:AbstractGMLType may have unique identity within an instance document that is specified with gml:id attributes. For example, roads r1, r2 and r3, and points p1111, p7654 and p9876 in examplePositions.xml have gml:id attributes.

A gml:pointRep property element may contain a gml:Point, as shown by points p1111 and p7654 in ex:Road r1 and point p9876 in ex:Road r3. Or instead of containing a gml:Point, a gml:pointRep property element may reference a gml:Point element defined elsewhere. For example, the ex:Road r3 uses bare name XPointers to reference points p1111 and p7654 in examplePositions.xml. It also uses XLink/XPath/XPointer expressions to reference points p1111 and p2345 in another document – positions.xml.

Note that XML IDs are only unique within the context of a single document. So "#p1111" and "positions.xml#p1111" in examplePositions.xml refer to different point instances. Those diffent point instances probably have different direct positions or coordinates. The fact that they share the XML ID "P1111" is purely coincidental.

Listing F.3.1.2: examplePositions.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ex:RoadInfrastructure xmlns:ex="http://www.opengis.net/examples" xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/examples
exampleRoad.xsd ">
 <description>This example shows different representations of a point array in curve segments.</description>
 <boundedBy>
  <Envelope srsName="somelistofcrs.xml#1234">
   <pos>0 0</pos>
   <pos>50 50</pos>
  </Envelope>
 </boundedBy>
 <featureMember>
  <ex:Road gml:id="r1">
   <curveProperty>
    <LineString srsName="somelistofcrs.xml#1234">
     <pointRep>
      <Point gml:id="p1111">
       <pos>1 1</pos>
      </Point>
     </pointRep>
     <pos>2 2</pos>
     <pos>3 4</pos>
     <pos>4 7</pos>
     <pos>5 11</pos>
     <pos>6 15</pos>
     <pointRep>
      <Point gml:id="p7654">
       <pos>7 20</pos>
      </Point>
     </pointRep>
```

```
        </LineString>
      </curveProperty>
    </ex:Road>
  </featureMember>
  <featureMember>
    <ex:Road gml:id="r2">
      <curveProperty>
        <LineString srsName="somelistofcrs.xml#1234">
          <coordinates>1,1 2,2 3,4 4,7 5,11 6,15 7,20</coordinates>
        </LineString>
      </curveProperty>
    </ex:Road>
  </featureMember>
  <featureMember>
    <ex:Road gml:id="r3">
      <curveProperty>
        <LineString srsName="somelistofcrs.xml#1234">
          <pointRep xlink:href="points.xml#p1111"/>
          <pointRep xlink:href="#p1111"/>
          <pos>2 2</pos>
          <pos>3 4</pos>
          <pointRep>
            <Point gml:id="p9876">
              <pos>4 7</pos>
            </Point>
          </pointRep>
          <pos>5 11</pos>
          <pos>6 15</pos>
          <pointRep xlink:href="#p7654"/>
          <pointRep xlink:href="points.xml#p2345"/>
        </LineString>
      </curveProperty>
    </ex:Road>
  </featureMember>
</ex:RoadInfrastructure>
```

Listing F.3.1.3 is an example of instances of two different feature types, a road and a bridge, sharing a gml:LineString geometry using an XLink. It also demonstrates a gml:CompositeCurve composed of gml:curveMember elements that are geometries with different interpolation. Some of the gml:curveMember elements are gml:LineStrings with linear interpolation between points. Others are instances of gml:CubicSpline. Both the gml:CompositeCurve and gml:CubicSpline are new with GML version 3.

Listing F.3.1.3: exampleRoad.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ex:RoadInfrastructure xmlns:ex="http://www.opengis.net/examples" xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/examples
exampleRoad.xsd">
  <description>Example of a road and a bridge sharing geometry. The geometry of the road uses different
interpolations.</description>
  <boundedBy>
    <Envelope srsName="somelistofcrs.xml#1234">
      <pos>0 0</pos>
      <pos>50 50</pos>
    </Envelope>
  </boundedBy>
  <featureMember>
    <ex:Road gml:id="r1">
      <curveProperty>
        <CompositeCurve srsName="somelistofcrs.xml#1234">
          <curveMember>
            <Curve gml:id="c101">
              <segments>
                <LineStringSegment>
                  <coordinates>...</coordinates>
                </LineStringSegment>
                <CubicSpline>
                  <coordinates>...</coordinates>
                  <vectorAtStart>1 0</vectorAtStart>
                  <vectorAtEnd>1 0</vectorAtEnd>
                </CubicSpline>
              </segments>
            </Curve>
          </curveMember>
          <curveMember>
            <LineString gml:id="c102">
              <coordinates>...</coordinates>
            </LineString>
          </curveMember>
          <curveMember>
            <Curve gml:id="c103">
              <segments>
                <CubicSpline>
                  <coordinates>...</coordinates>
                  <vectorAtStart>1 0</vectorAtStart>
                  <vectorAtEnd>1 0</vectorAtEnd>
                </CubicSpline>
                <LineStringSegment>
                  <coordinates>...</coordinates>
                </LineStringSegment>
              </segments>
            </Curve>
          </curveMember>
        </CompositeCurve>
      </curveProperty>
    </ex:Road>
  </featureMember>
  <featureMember>
    <ex:Bridge gml:id="br1">
```

```
      <curveProperty xlink:href="#c102"/>
    </ex:Bridge>
  </featureMember>
</ex:RoadInfrastructure>
```

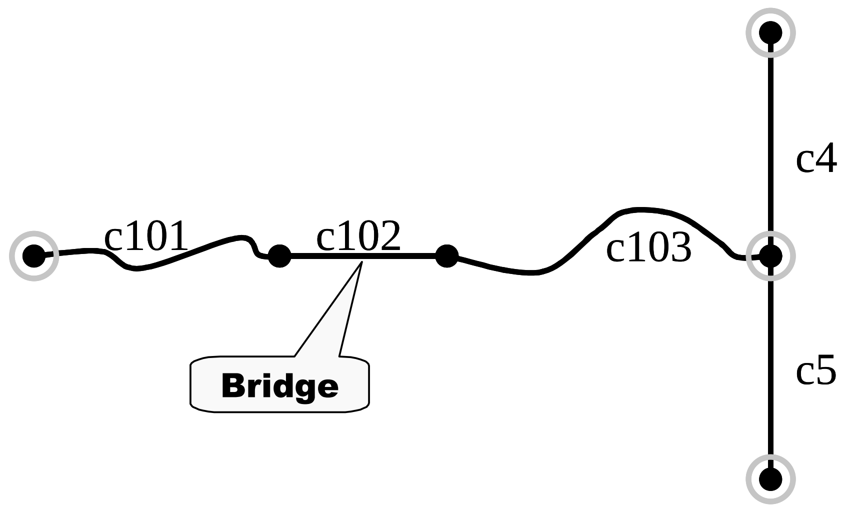


Figure F.3.1.3 – Road Example

F.3.2 The Bus Route Example

Figure F.3.2.1 represents data entirely constructed from nodes and edges that will support defining a bus route. The nodes have no "internal" structure. The edges have start (negative) and end (positive) nodes described as a pair of directed node properties, which form their boundary.

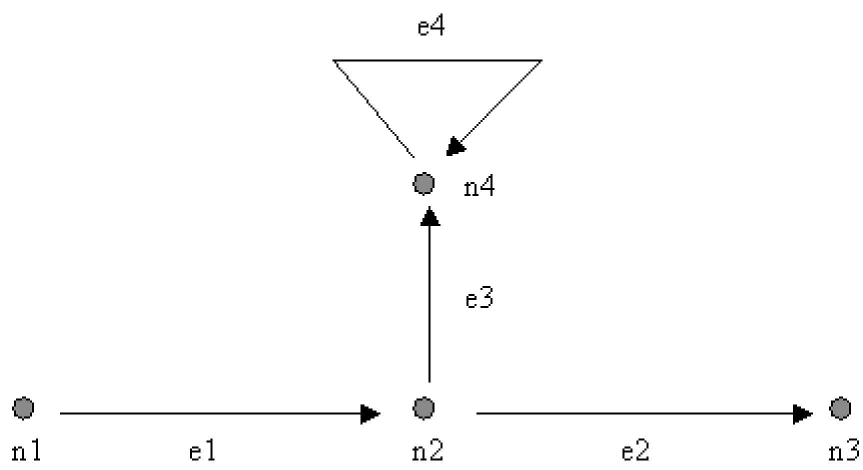


Figure F.3.2.1 - Simple Network

With such data we can ask questions of the form "which sequence of edges is required to move from one node to another?" - for instance, moving from n1 to n3 gives the sequence {+e1,+e2} and from n3 to n4 the sequence {-e2,+e3}.

Note that we can refer to edges in two directions. Because an edge has an implicit direction (i.e., e1 can be said to go from n1 to n2), it can be traversed in either direction. Thus -e1 can be said to go from n2 to n1.

The example in figure 3.2.1 may be represented by the GML data in simpleNetwork.xml:

Listing F.3.2.1 simpleNetwork.xml

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

<Topology xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app

networkExamples.xsd">

  <gml:Node gml:id="n1"/>

  <gml:Node gml:id="n2"/>

  <gml:Node gml:id="n3"/>

  <gml:Node gml:id="n4"/>

  <gml:Edge gml:id="e1">

    <gml:directedNode orientation="-" xlink:href="#n1"/>

    <gml:directedNode orientation="+" xlink:href="#n2"/>

  </gml:Edge>

  <gml:Edge gml:id="e2">

    <gml:directedNode orientation="-" xlink:href="#n2"/>

    <gml:directedNode orientation="+" xlink:href="#n3"/>

  </gml:Edge>

  <gml:Edge gml:id="e3">

    <gml:directedNode orientation="-" xlink:href="#n2"/>

    <gml:directedNode orientation="+" xlink:href="#n4"/>

  </gml:Edge>

  <gml:Edge gml:id="e4">

    <gml:directedNode orientation="-" xlink:href="#n4"/>

    <gml:directedNode orientation="+" xlink:href="#n4"/>

  </gml:Edge>

</Topology>
```

Figure F.3.2.2 shows the effect of adding real world objects to the pure topology. Each edge is now used by a Road, we have Towns at n1 and n3, and a BusStop at n4. This means that we can now identify which roads are between the two towns ({A14,A428}) and between Cambridge and the bus stop ({A428,B12}). The schema for all GML documents in this example is in listing F.3.2.7 networkExample.xsd.

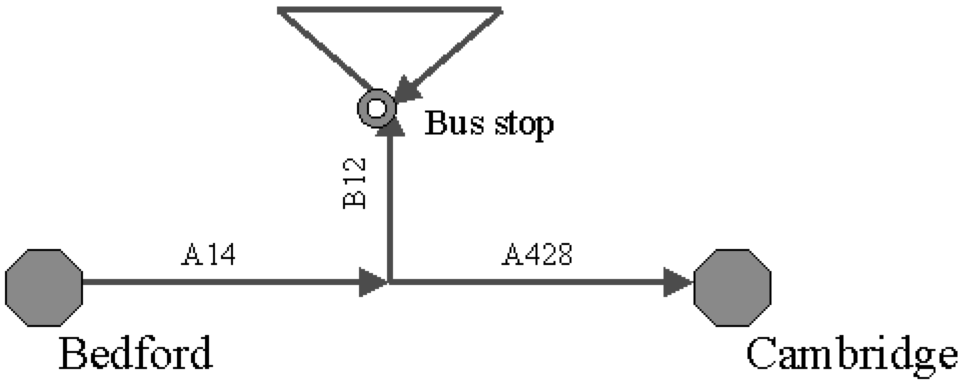


Figure F.3.2.2 – Real World Objects

Our real world Road, Town and Bus Stop objects relate to the edges and nodes via topological points and edges, which are properties of the features -- we do not sub-class them. This mirrors the non-topological situation, where a non-topological Road might have a centreLineOf property, which refers to a LineString object, it does not sub-class LineString.

The example in figure 3.2.2 may be represented by the GML data in realWorld.xml:

Listing F.3.2.2 realWorld.xml

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

<gml:FeatureCollection xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"

xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app networkExamples.xsd">

  <gml:boundedBy>

    <gml:Box>

      <gml:coordinates>0,0</gml:coordinates>

      <gml:coordinates>100,100</gml:coordinates>

    </gml:Box>

  </gml:boundedBy>

  <gml:featureMember>

    <Town fid="t1">

      <gml:name>Bedford</gml:name>

      <gml:directedNode orientation="+" xlink:href="simpleNetwork.xml#n1"/>

    </Town>

  </gml:featureMember>
```



```
  <gml:featureMember>
   <Town fid="t2">
    <gml:name>Cambridge</gml:name>
    <gml:directedNode orientation="+" xlink:href="simpleNetwork.xml#n3"/>
   </Town>
  </gml:featureMember>
  <gml:featureMember>
   <Road fid="r1">
    <gml:name>A14</gml:name>
    <avSpeed>60</avSpeed>
    <gml:directedEdge orientation="+" xlink:href="simpleNetwork.xml#e1"/>
   </Road>
  </gml:featureMember>
  <gml:featureMember>
   <Road fid="r2">
    <gml:name>A428</gml:name>
    <avSpeed>45</avSpeed>
    <gml:directedEdge orientation="+" xlink:href="simpleNetwork.xml#e2"/>
   </Road>
  </gml:featureMember>
  <gml:featureMember>
   <Road fid="r3">
    <gml:name>B12</gml:name>
    <avSpeed>30</avSpeed>
    <gml:directedEdge orientation="+" xlink:href="simpleNetwork.xml#e3"/>
   </Road>
  </gml:featureMember>
  <gml:featureMember>
   <BusStop fid="b1">
    <gml:directedNode orientation="+" xlink:href="simpleNetwork.xml#n4"/>
   </BusStop>
  </gml:featureMember>
 </gml:FeatureCollection>
```


Figure F.3.2.3 shows the effect of adding geometry to the edges and nodes. Once this has been done, the purpose of the edges and nodes is to abstract out the geometry -- this is what defines them -- not their role in the network. Assume that we are working at a scale where it is not necessary to provide alternate geometries for different carriageways, and thus Roads are still represented by single edges.

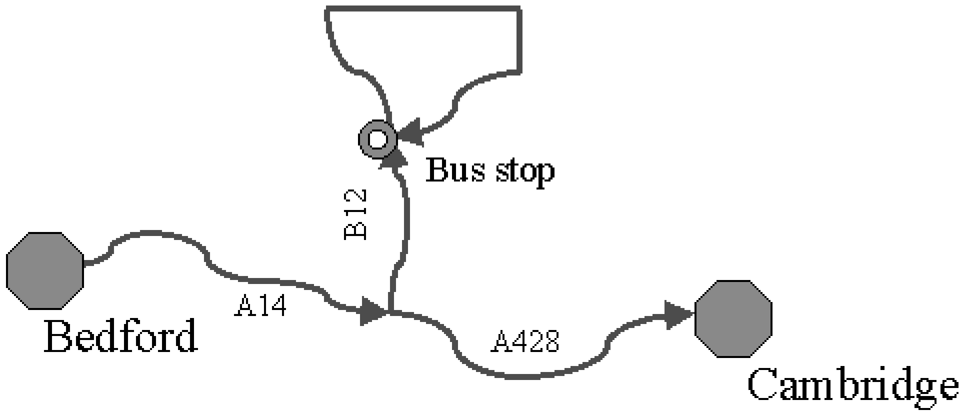


Figure F.3.2.3 – Adding Geometry

We can now begin to use some of the attribution on the Roads (for instance, average speed) to perform time and distance calculations. And the results vary depending upon how we travel. Roads and cycle tracks may use exactly the same edges but the speed of travel differs greatly.

Finding an optimal route must make use of the RWO information, as well as the topological and geometric information. In fact route finding works with the RWOs,

and uses the fact that the associated topological primitives (i.e. edges and nodes) are connected to infer the connectivity of the RWOs.

A route finding application can then take start and end points (the RWO Town objects), follow down to the relevant nodes, determine the edges connecting those nodes (as in Example F.3.2.1), derive distances per edge from the attached geometries, and follow back up to the Road objects to determine any "real world" attribution, such as speed or capacity.

Adding geometry information to simpleNetwork.xml shown in Listing F.3.2.1 transforms it into withGeometry.xml shown in Listing F.3.2.3

Listing F.3.2.3 withGeometry.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Topology xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app networkExamples.xsd">
  <gml:Node gml:id="n1">
    <gml:pointProperty>
      <gml:Point>
        <gml:coordinates>10, 10</gml:coordinates>
```

```
    </gml:Point>
  </gml:pointProperty>
</gml:Node>
<gml:Node gml:id="n2">
  <gml:pointProperty>
    <gml:Point>
      <gml:coordinates>20, 10</gml:coordinates>
    </gml:Point>
  </gml:pointProperty>
</gml:Node>
<gml:Node gml:id="n3">
  <gml:pointProperty>
    <gml:Point>
      <gml:coordinates>30, 10</gml:coordinates>
    </gml:Point>
  </gml:pointProperty>
</gml:Node>
<gml:Node gml:id="n4">
  <gml:pointProperty>
    <gml:Point>
      <gml:coordinates>20, 20</gml:coordinates>
    </gml:Point>
  </gml:pointProperty>
</gml:Node>
<gml:Edge gml:id="e1">
  <gml:directedNode orientation="-" xlink:href="#n1"/>
  <gml:directedNode orientation="+" xlink:href="#n2"/>
  <gml:centerLineOf>
    <gml:LineString>
      <gml:coordinates>10,10 20,10</gml:coordinates>
    </gml:LineString>
  </gml:centerLineOf>
</gml:Edge>
<gml:Edge gml:id="e2">
  <gml:directedNode orientation="-" xlink:href="#n2"/>
  <gml:directedNode orientation="+" xlink:href="#n3"/>
  <gml:centerLineOf>
    <gml:LineString>
      <gml:coordinates>20,10 30,10</gml:coordinates>
    </gml:LineString>
```

```
    </gml:centerLineOf>
  </gml:Edge>
  <gml:Edge gml:id="e3">
    <gml:directedNode orientation="-" xlink:href="#n2"/>
    <gml:directedNode orientation="+" xlink:href="#n4"/>
    <gml:centerLineOf>
      <gml:LineString>
        <gml:coordinates>20,10 20,20</gml:coordinates>
      </gml:LineString>
    </gml:centerLineOf>
  </gml:Edge>
  <gml:Edge gml:id="e4">
    <gml:directedNode orientation="-" xlink:href="#n4"/>
    <gml:directedNode orientation="+" xlink:href="#n4"/>
    <gml:centerLineOf>
      <gml:LineString>
        <gml:coordinates>20,20 15,25 20,30 25,25 20,20</gml:coordinates>
      </gml:LineString>
    </gml:centerLineOf>
  </gml:Edge>
</Topology>
```

The only corresponding change required to convert Listing F.3.2.2 realWorld.xml into Listing F.3.2.4 realWorld2.xml that references withGeometry.xml shown in Listing F.3.2.3 is to change the base document specified in the xlink:href references to the topological elements:

Listing F.3.2.4 realWorld2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app networkExamples.xsd">
  <gml:boundedBy>
    <gml:Box>
      <gml:coordinates>0,0</gml:coordinates>
      <gml:coordinates>100,100</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <Town fid="t1">
```



```
      <gml:name>Bedford</gml:name>
      <gml:directedNode orientation="+" xlink:href="withGeometry.xml#n1"/>
    </Town>
  </gml:featureMember>
  <gml:featureMember>
    <Town fid="t2">
      <gml:name>Cambridge</gml:name>
      <gml:directedNode orientation="+" xlink:href="withGeometry.xml#n3"/>
    </Town>
  </gml:featureMember>
  <gml:featureMember>
    <Road fid="r1">
      <gml:name>A14</gml:name>
      <avSpeed>60</avSpeed>
      <gml:directedEdge orientation="+" xlink:href="withGeometry.xml#e1"/>
    </Road>
  </gml:featureMember>
  <gml:featureMember>
    <Road fid="r2">
      <gml:name>A428</gml:name>
      <avSpeed>45</avSpeed>
      <gml:directedEdge orientation="+" xlink:href="withGeometry.xml#e2"/>
    </Road>
  </gml:featureMember>
  <gml:featureMember>
    <Road fid="r3">
      <gml:name>B12</gml:name>
      <avSpeed>30</avSpeed>
      <gml:directedEdge orientation="+" xlink:href="withGeometry.xml#e3"/>
    </Road>
  </gml:featureMember>
  <gml:featureMember>
    <BusStop fid="b1">
      <gml:directedNode orientation="+" xlink:href="withGeometry.xml#n4"/>
    </BusStop>
  </gml:featureMember>
</gml:FeatureCollection>
```

Given a road network, the next question is how to represent two-way streets, as shown in Figure F.3.2.4.

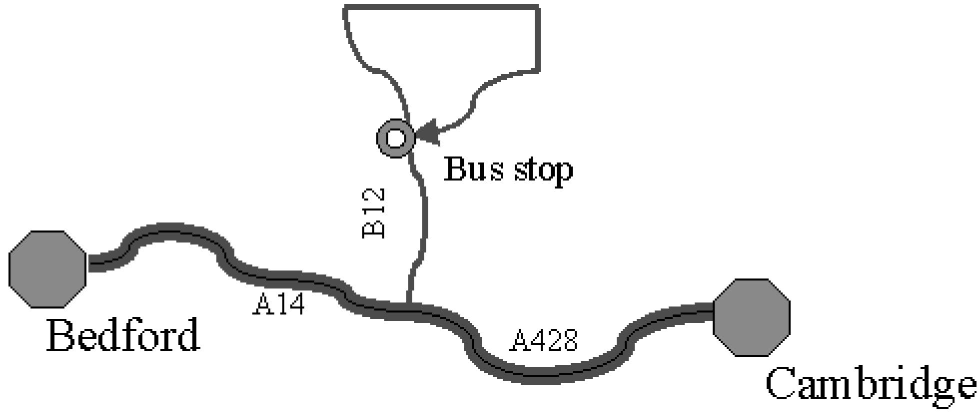


Figure F.3.2.4 – Two-way streets

The two possible real world situations are:

1. A true one-way road - this is shown in the example with the "unnamed" loop Road that uses edge e4. Here, the Road is related through a directed edge property to the actual edge.
2. A "dual carriageway" composed of two road carriageways, each of which is one-way.

In a "real life" application, the attribution for each direction on a dual carriageway may be different (in particular, such quantities as average speed may differ -- consider a road on a steep hill). We would therefore introduce an intermediate entity, Carriageway, which relates to a single directed edge. A bi-directional road would then relate to a pair of Carriageways (one for each direction), and a one-way road would relate to a single Carriageway. Listing F.3.2.5 shows the A- and B-roads decomposed in this way.

Listing F.3.2.5 twoWayStreets.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app networkExamples.xsd">
 <gml:boundedBy>
  <gml:Box>
   <gml:coordinates>0,0</gml:coordinates>
   <gml:coordinates>100,100</gml:coordinates>
  </gml:Box>
 </gml:boundedBy>
 <gml:featureMember>
```

```
  <Town fid="t1">
    <gml:name>Bedford</gml:name>
    <gml:directedNode orientation="+" xlink:href="withGeometryxml#n1"/>
  </Town>
</gml:featureMember>
<gml:featureMember>
  <Town fid="t2">
    <gml:name>Cambridge</gml:name>
    <gml:directedNode orientation="+" xlink:href="withGeometryxml#n3"/>
  </Town>
</gml:featureMember>
<gml:featureMember>
  <Road fid="r1a">
    <gml:name>A14E</gml:name>
    <avSpeed>60</avSpeed>
    <gml:directedEdge orientation="+" xlink:href="withGeometryxml#e1"/>
  </Road>
</gml:featureMember>
<gml:featureMember>
  <Road fid="r1b">
    <gml:name>A14W</gml:name>
    <avSpeed>55</avSpeed>
    <gml:directedEdge orientation="-" xlink:href="withGeometryxml#e1"/>
  </Road>
</gml:featureMember>
<gml:featureMember>
  <Road fid="r2a">
    <gml:name>A428E</gml:name>
    <avSpeed>45</avSpeed>
    <gml:directedEdge orientation="+" xlink:href="withGeometryxml#e2"/>
  </Road>
</gml:featureMember>
<gml:featureMember>
  <Road fid="r2b">
    <gml:name>A428W</gml:name>
    <avSpeed>50</avSpeed>
    <gml:directedEdge orientation="-" xlink:href="withGeometryxml#e2"/>
  </Road>
</gml:featureMember>
<gml:featureMember>
```

```
    <Road fid="r3a">
      <gml:name>B12E</gml:name>
      <avSpeed>30</avSpeed>
      <gml:directedEdge orientation="+" xlink:href="withGeometryxml#e3"/>
    </Road>
  </gml:featureMember>
  <gml:featureMember>
    <Road fid="r3b">
      <gml:name>B12W</gml:name>
      <avSpeed>30</avSpeed>
      <gml:directedEdge orientation="-" xlink:href="withGeometryxml#e3"/>
    </Road>
  </gml:featureMember>
  <gml:featureMember>
    <BusStop fid="b1">
      <gml:directedNode orientation="+" xlink:href="withGeometryxml#n4"/>
    </BusStop>
  </gml:featureMember>
</gml:FeatureCollection>
```

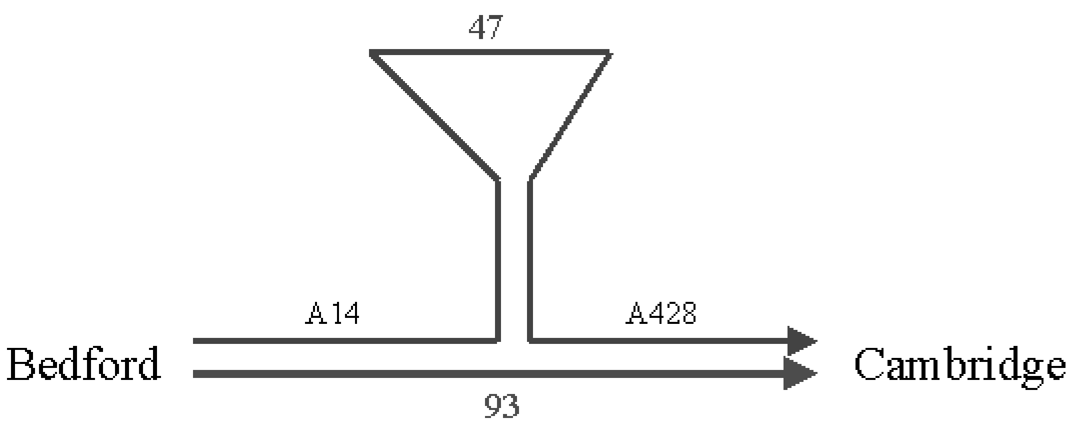


Figure F.3.2.5 – Bus Routes

Here we describe two bus routes - the first composed of the edge sequence {+e1,+e2} (route number 93) and the second of {+e1,+e3,+e4,-e3,+e2}.

One can deduce the Roads involved, and the Towns and BusStops visited, even calculate the timetable, by following back from the topology to the real world objects. Note that to represent a route unambiguously requires a list of directed edges - there are four interpretations of the set {+e1, -e1, +e2, -e2}, namely:

+e1 -> +e2 -> -e2 -> -e1

-e2 -> -e1 -> +e1 -> +e2

-e1 -> +e1 -> +e2 -> -e2

+e2 -> -e2 -> -e1 -> +e1

Obviously roadways are a special (single directed edge) version of this general rule. The GML data for our bus routes is in Listing F.3.2.6 busRoutes.xml.

Listing F.3.2.6 busRoutes.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app networkExamples.xsd">
  <gml:boundedBy>
    <gml:Box>
      <gml:coordinates>0,0</gml:coordinates>
      <gml:coordinates>100,100</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <BusRoute fid="route1">
      <number>93</number>
      <gml:directedEdge orientation="+" xlink:href="withGeometry.xml#e1"/>
      <gml:directedEdge orientation="+" xlink:href="withGeometry.xml#e2"/>
    </BusRoute>
  </gml:featureMember>
  <gml:featureMember>
    <BusRoute fid="route2">
      <number>47</number>
      <gml:directedEdge orientation="+" xlink:href="withGeometry.xml#e1"/>
      <gml:directedEdge orientation="+" xlink:href="withGeometry.xml#e3"/>
      <gml:directedEdge orientation="+" xlink:href="withGeometry.xml#e4"/>
      <gml:directedEdge orientation="-" xlink:href="withGeometry.xml#e3"/>
      <gml:directedEdge orientation="+" xlink:href="withGeometry.xml#e2"/>
    </BusRoute>
  </gml:featureMember>
</gml:FeatureCollection>
```

Listing F.3.2.7 networkExamples.xsd

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

```
<schema targetNamespace="http://www.opengis.net/app" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron" xmlns:app="http://www.opengis.net/app"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
 elementFormDefault="qualified" version="3.0">
  <import namespace="http://www.opengis.net/gml" schemaLocation="../../base/gml.xsd"/>
  <!-- =========== simple network example ================ -->
  <complexType name="TopologyArrayAssociationType">
    <annotation>
      <documentation>Restrict contents to Topologies</documentation>
    </annotation>
    <complexContent>
      <restriction base="gml:ArrayAssociationType">
        <sequence>
          <element ref="gml:_Topology" minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
      </restriction>
    </complexContent>
  </complexType>
  <element name="Topology" type="app:TopologyArrayAssociationType">
    <annotation>
      <documentation>Topology Array containing Nodes, Edges, Faces and/or TopoSolids</documentation>
    </annotation>
  </element>
  <!-- =========== real world example ================ -->
  <element name="Town" type="app:TownType" substitutionGroup="gml:_Feature"/>
  <complexType name="TownType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:directedNode"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <element name="Road" type="app:RoadType" substitutionGroup="gml:_Feature"/>
  <complexType name="RoadType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element name="avSpeed" type="positiveInteger"/>
```

```
        <element ref="gml:directedEdge"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<element name="BusStop" type="app:BusStopType" substitutionGroup="gml:_Feature"/>
<complexType name="BusStopType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element ref="gml:directedNode"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
<!-- ========== bus routes example =============== -->
<element name="BusRoute" type="app:BusRouteType" substitutionGroup="gml:_Feature"/>
<complexType name="BusRouteType">
  <complexContent>
    <extension base="gml:AbstractFeatureType">
      <sequence>
        <element name="number" type="positiveInteger"/>
        <element ref="gml:directedEdge" maxOccurs="unbounded"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>
</schema>
```

F.3.3 The Administrative Districts Example

Figure F.3.3.1 shows a manifold of nodes, edges and faces. The purpose is to abstract the adjacency and containment relationships for later use by a set of Real World Objects.

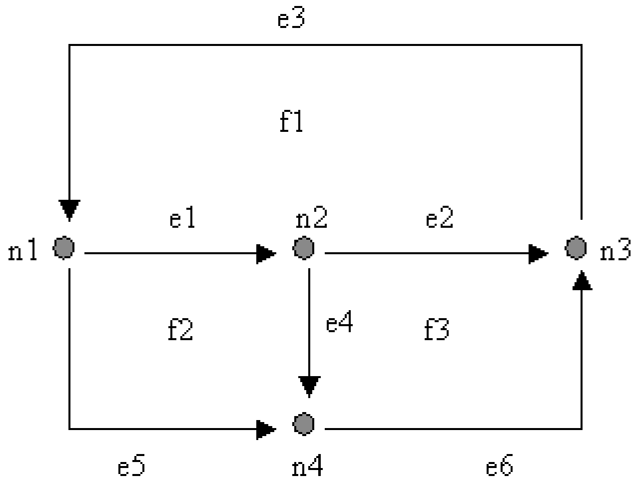


Figure F.3.3.1 – Simple topological manifold

Just as edges are composed from a pair of directed nodes, faces are composed from a set of directed edges which can be traversed to determine the face boundary. Listing F.3.3.1 simpleManifold.xml shows the GML data encoding for Figure F.3.3.1.

Listing F.3.3.1 simpleManifold.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<Topology xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app planarExamples.xsd">
  <gml:Node gml:id="n1">
    <gml:pointProperty>
      <gml:Point>
        <gml:coordinates>10, 10</gml:coordinates>
      </gml:Point>
    </gml:pointProperty>
  </gml:Node>
  <gml:Node gml:id="n2">
    <gml:pointProperty>
      <gml:Point>
        <gml:coordinates>20, 10</gml:coordinates>
      </gml:Point>
    </gml:pointProperty>
  </gml:Node>
```

```
<gml:Node gml:id="n3">
  <gml:pointProperty>
    <gml:Point>
      <gml:coordinates>30, 10</gml:coordinates>
    </gml:Point>
  </gml:pointProperty>
</gml:Node>
<gml:Node gml:id="n4">
  <gml:pointProperty>
    <gml:Point>
      <gml:coordinates>20, 0</gml:coordinates>
    </gml:Point>
  </gml:pointProperty>
</gml:Node>
<gml:Edge gml:id="e1">
  <gml:directedNode orientation="-" xlink:href="#n1"/>
  <gml:directedNode orientation="+" xlink:href="#n2"/>
  <gml:centerLineOf>
    <gml:LineString>
      <gml:coordinates>10,10 20,10</gml:coordinates>
    </gml:LineString>
  </gml:centerLineOf>
</gml:Edge>
<gml:Edge gml:id="e2">
  <gml:directedNode orientation="-" xlink:href="#n2"/>
  <gml:directedNode orientation="+" xlink:href="#n3"/>
  <gml:centerLineOf>
    <gml:LineString>
      <gml:coordinates>20,10 30,10</gml:coordinates>
    </gml:LineString>
  </gml:centerLineOf>
</gml:Edge>
<gml:Edge gml:id="e3">
  <gml:directedNode orientation="-" xlink:href="#n3"/>
  <gml:directedNode orientation="+" xlink:href="#n1"/>
  <gml:centerLineOf>
    <gml:LineString>
      <gml:coordinates>30,10 30,20 10,20 10,10</gml:coordinates>
    </gml:LineString>
  </gml:centerLineOf>
```

```
    </gml:Edge>
    <gml:Edge gml:id="e4">
      <gml:directedNode orientation="-" xlink:href="#n2"/>
      <gml:directedNode orientation="+" xlink:href="#n4"/>
      <gml:centerLineOf>
        <gml:LineString>
          <gml:coordinates>20,10 20,0</gml:coordinates>
        </gml:LineString>
      </gml:centerLineOf>
    </gml:Edge>
    <gml:Edge gml:id="e5">
      <gml:directedNode orientation="-" xlink:href="#n1"/>
      <gml:directedNode orientation="+" xlink:href="#n4"/>
      <gml:centerLineOf>
        <gml:LineString>
          <gml:coordinates>10,10 10,0 20,0</gml:coordinates>
        </gml:LineString>
      </gml:centerLineOf>
    </gml:Edge>
    <gml:Edge gml:id="e6">
      <gml:directedNode orientation="-" xlink:href="#n4"/>
      <gml:directedNode orientation="+" xlink:href="#n3"/>
      <gml:centerLineOf>
        <gml:LineString>
          <gml:coordinates>20,0 30,0 30,10</gml:coordinates>
        </gml:LineString>
      </gml:centerLineOf>
    </gml:Edge>
    <gml:Face gml:id="f1">
      <gml:directedEdge orientation="+" xlink:href="#e1"/>
      <gml:directedEdge orientation="+" xlink:href="#e2"/>
      <gml:directedEdge orientation="+" xlink:href="#e3"/>
      <gml:surfaceProperty>
        <gml:Polygon>
          <gml:exterior>
            <gml:LinearRing>
              <gml:coordinates>10,10 20,10 30,10 30,20 10,20 10,10</gml:coordinates>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
```

```
    </gml:surfaceProperty>
   </gml:Face>
   <gml:Face gml:id="f2">
    <gml:directedEdge orientation="+" xlink:href="#e5"/>
    <gml:directedEdge orientation="-" xlink:href="#e4"/>
    <gml:directedEdge orientation="-" xlink:href="#e1"/>
    <gml:surfaceProperty>
     <gml:Polygon>
      <gml:exterior>
       <gml:LinearRing>
        <gml:coordinates>10,10 10,0 20,0 20,10 10,10</gml:coordinates>
       </gml:LinearRing>
      </gml:exterior>
     </gml:Polygon>
    </gml:surfaceProperty>
   </gml:Face>
   <gml:Face gml:id="f3">
    <gml:directedEdge orientation="+" xlink:href="#e6"/>
    <gml:directedEdge orientation="-" xlink:href="#e2"/>
    <gml:directedEdge orientation="+" xlink:href="#e4"/>
    <gml:surfaceProperty>
     <gml:Polygon>
      <gml:exterior>
       <gml:LinearRing>
        <gml:coordinates>20,0 30,0 30,10 20,10 20,0</gml:coordinates>
       </gml:LinearRing>
      </gml:exterior>
     </gml:Polygon>
    </gml:surfaceProperty>
   </gml:Face>
  </Topology>
```

The face primitives can be used to define some Real World Objects such as a series of Administrative Districts. The smallest Administrative District in our example is the Ward. The manifold then corresponds to three Wards as you can see in Figure F.3.3.2 Administrative Districts.

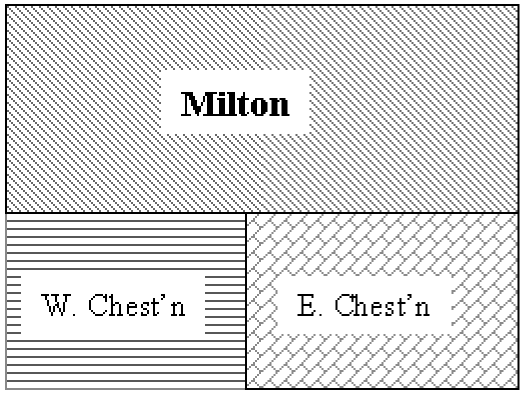


Figure F.3.3.2 – Administrative Districts

The encoding for these Administrative Districts is shown in Listing F.3.3.2 adminDistricts.xml.

Listing F.3.3.2 adminDistricts.xml

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

<gml:FeatureCollection xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"

xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app planarExamples.xsd">

 <gml:boundedBy>

  <gml:Box>

   <gml:coordinates>0,0</gml:coordinates>

   <gml:coordinates>100,100</gml:coordinates>

  </gml:Box>

 </gml:boundedBy>

 <gml:featureMember>

  <Ward fid="w1">

   <gml:name>Milton</gml:name>

   <gml:directedTopoSurface orientation="+">

    <gml:TopoSurface>

     <gml:directedFace orientation="+" xlink:href="simpleManifold.xml#f1"/>

    </gml:TopoSurface>

   </gml:directedTopoSurface>

  </Ward>

 </gml:featureMember>

 <gml:featureMember>

  <Ward fid="w2">
```

```
      <gml:name>W. Chesterton</gml:name>
      <gml:directedTopoSurface orientation="+">
        <gml:TopoSurface>
          <gml:directedFace orientation="+" xlink:href="simpleManifold.xml#f2"/>
        </gml:TopoSurface>
      </gml:directedTopoSurface>
    </Ward>
  </gml:featureMember>
  <gml:featureMember>
    <Ward fid="w3">
      <gml:name>F. Chesterton</gml:name>
      <gml:directedTopoSurface orientation="+">
        <gml:TopoSurface>
          <gml:directedFace orientation="+" xlink:href="simpleManifold.xml#f3"/>
        </gml:TopoSurface>
      </gml:directedTopoSurface>
    </Ward>
  </gml:featureMember>
</gml:FeatureCollection>
```

It is common for Administrative Districts to overlap. For example, a District Council or County may encompass many Wards, as shown in Figure F.3.3.3.

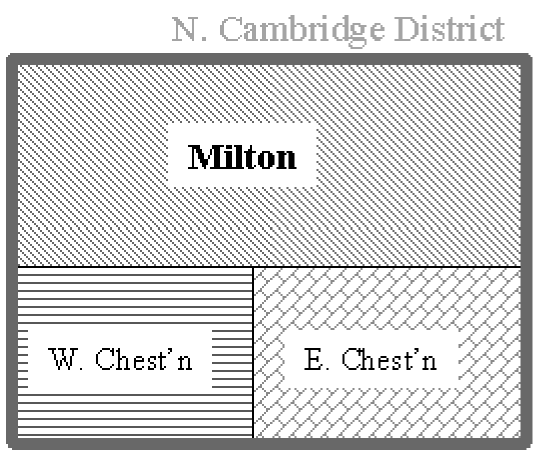


Figure F.3.3.3 – Overlapping Administrative Districts

Listing F.3.3.3 adminOverlap.xml below shows the encoding for the N. Cambridge District in terms of the same faces used to encode the Wards it contains.

Listing F.3.3.3 adminOverlap.xml


```xml
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/app planarExamples.xsd">
  <gml:boundedBy>
    <gml:Box>
      <gml:coordinates>0,0</gml:coordinates>
      <gml:coordinates>100,100</gml:coordinates>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <District fid="d1">
      <gml:name>N. Cambridge</gml:name>
      <gml:directedTopoSurface orientation="+">
        <gml:TopoSurface>
          <gml:directedFace orientation="+" xlink:href="simpleManifold.xml#f1"/>
          <gml:directedFace orientation="+" xlink:href="simpleManifold.xml#f2"/>
          <gml:directedFace orientation="+" xlink:href="simpleManifold.xml#f3"/>
        </gml:TopoSurface>
      </gml:directedTopoSurface>
    </District>
  </gml:featureMember>
</gml:FeatureCollection>
```


Within a manifold, individual edges may constitute any number of line features and part of the boundary of any number of area features. For example, the dividing line between two real world area features may be defined to be a particular stretch of a road or river. Figure F.3.3.4 Shared Boundaries shows Road A14 that is also the boundary between Milton and the West and East Chesterton Wards.

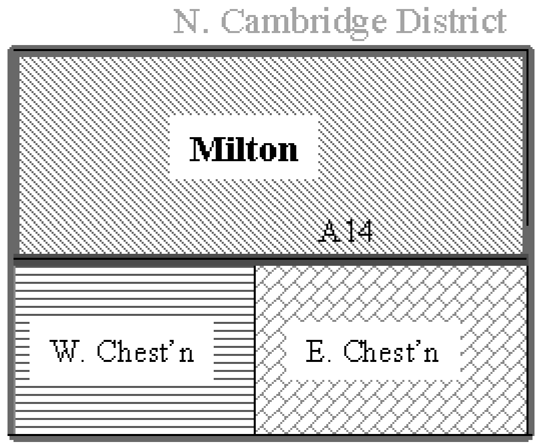


Figure F.3.3.3 – Shared Boundaries

Listing F.3.3.3 topoCurve.xml shows the GML encoding for Road A14 as a TopoCurve composed of directed Edges.

Listing F.3.3.3 topoCurve.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:FeatureCollection xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app planarExamples.xsd">
 <gml:boundedBy>
  <gml:Box>
   <gml:coordinates>0,0</gml:coordinates>
   <gml:coordinates>100,100</gml:coordinates>
  </gml:Box>
 </gml:boundedBy>
 <gml:featureMember>
  <Road fid="r11">
   <gml:name>A14</gml:name>
   <gml:directedTopoCurve orientation="+">
    <gml:TopoCurve>
     <gml:directedEdge orientation="+" xlink:href="simpleManifold.xml#e1"/>
     <gml:directedEdge orientation="+" xlink:href="simpleManifold.xml#e2"/>
    </gml:TopoCurve>
   </gml:directedTopoCurve>
  </Road>
 </gml:featureMember>
```

```
</gml:FeatureCollection>
```

F.3.4 The Lake Example

In general, faces are not simply composed of the edges, which separate them. They may contain isolated or dangling edges or isolated nodes. Note that the face f1 in Figure F.3.4.1 appears on both sides of edges e2 and e3. The face is therefore considered to use these edges in both orientations. Edge e2 is considered part of the boundary of face f1 because it shares node n1 with edge e1 that is part of the boundary of f1. Edge e3 and node n3 are considered to be contained within face f1 but isolated because then are not part of the boundary of f1.

Faces may also completely enclose other faces. Consider an island (area feature) in a lake. In this case, the edges comprising the boundary of the island also form part of the boundary of the lake traversed in the opposite direction.

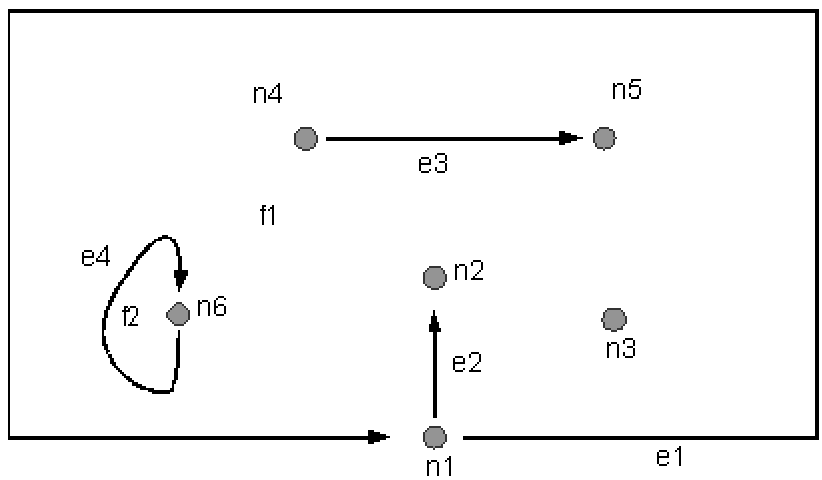


Figure F.3.4.1 – Faces and Containment

The topologies shown in Figure F.3.4.1 are encoded without geometry as shown in Listing F.3.4.1 faceContain.xml. Edge e3 is referenced twice as a directed edge with opposite orientations, as is done for dangling edge e2, where the semantics of the opposite orientation inclusions in 2D indicates that the edge has the same co-bounding face on each side, except that it shares no node with any bounding edge of the face.

Node n3 is referenced as an isolatedInFaceProperty of face f1. Nodes n4 and n5 are not referenced as isolatedProperties of face f1 because they bound edge e3, which is referenced as a directedEdge of face f1.

Listing F.3.4.1 faceContain.xml

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



```xml
<Topology xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app planarExamples.xsd">
 <gml:Node gml:id="n1"/>
 <gml:Node gml:id="n2"/>
 <gml:Node gml:id="n3"/>
 <gml:Node gml:id="n4"/>
 <gml:Node gml:id="n5"/>
 <gml:Node gml:id="n6"/>
 <gml:Edge gml:id="e1">
  <gml:directedNode orientation="-" xlink:href="#n1"/>
  <gml:directedNode orientation="+" xlink:href="#n1"/>
 </gml:Edge>
 <gml:Edge gml:id="e2">
  <gml:directedNode orientation="-" xlink:href="#n1"/>
  <gml:directedNode orientation="+" xlink:href="#n2"/>
 </gml:Edge>
 <gml:Edge gml:id="e3">
  <gml:directedNode orientation="-" xlink:href="#n4"/>
  <gml:directedNode orientation="+" xlink:href="#n5"/>
 </gml:Edge>
 <gml:Edge gml:id="e4">
  <gml:directedNode orientation="-" xlink:href="#n6"/>
  <gml:directedNode orientation="+" xlink:href="#n6"/>
 </gml:Edge>
<gml:Face gml:id="f1">
 <gml:isolatedInFaceProperty xlink:href="#n3"/>
 <gml:directedEdge orientation="+" xlink:href="#e1"/>
 <gml:directedEdge orientation="+" xlink:href="#e4"/>
 <gml:directedEdge orientation="+" xlink:href="#e2"/>
 <gml:directedEdge orientation="-" xlink:href="#e2"/>
 <gml:directedEdge orientation="+" xlink:href="#e3"/>
 <gml:directedEdge orientation="-" xlink:href="#e3"/>
</gml:Face>
 <gml:Face gml:id="f2">
  <gml:directedEdge orientation="-" xlink:href="#e4"/>
 </gml:Face>
</Topology>
```

To make the example more concrete, the topology above may be used as shown in Figure F.3.4.2 to realize a lake (f1) with a pier (dangling edge e2), a mooring (isolated edge e3), a buoy (isolated node n3), and an island (face f2).

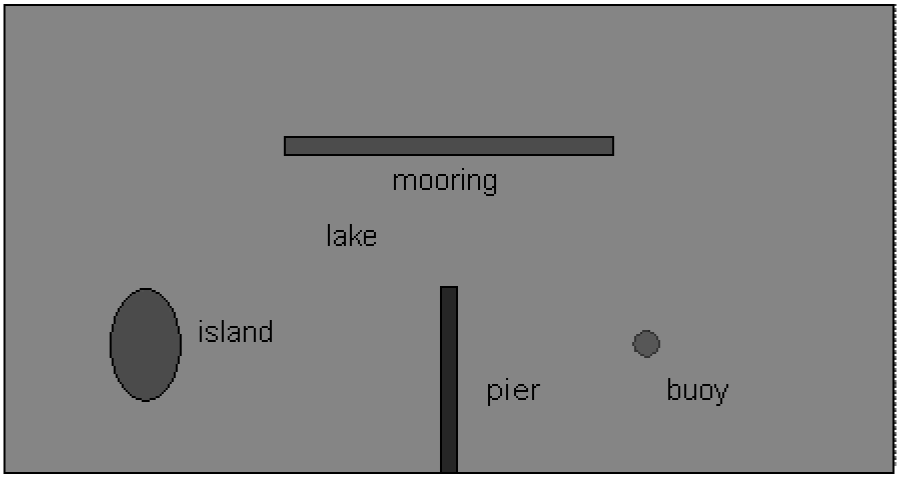


Figure F.3.4.2 – Lake

The lake and its contained real world objects may be encoded in GML as shown in Listing F.3.4.2 using the topologies from Listing F.3.4.1.

Listing F.3.4.2 lakeFace.xml

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

<gml:FeatureCollection xmlns="http://www.opengis.net/app" xmlns:sch="http://www.ascc.net/xml/schematron"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"

xmlns:app="http://www.opengis.net/app" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/app planarExamples.xsd">

  <gml:boundedBy>

    <gml:Box>

      <gml:coordinates>0,0</gml:coordinates>

      <gml:coordinates>100,100</gml:coordinates>

    </gml:Box>

  </gml:boundedBy>

  <gml:featureMember>

    <Lake fid="w1">

      <gml:directedFace orientation="+" xlink:href="faceContain.xml#f1"/>

    </Lake>

  </gml:featureMember>

  <gml:featureMember>

    <Island fid="i1">

      <gml:directedFace orientation="+" xlink:href="faceContain.xml#f2"/>
```

```
    </Island>
   </gml:featureMember>
   <gml:featureMember>
    <Pier fid="p1">
      <gml:directedEdge orientation="+" xlink:href="faceContain.xml#e2"/>
    </Pier>
   </gml:featureMember>
   <gml:featureMember>
    <Mooring>
      <gml:directedEdge xlink:href="faceContain.xml#e3"/>
    </Mooring>
   </gml:featureMember>
   <gml:featureMember>
    <Buoy>
      <gml:directedNode xlink:href="faceContain.xml#n3"/>
    </Buoy>
   </gml:featureMember>
</gml:FeatureCollection>
```

F.3.5 The Parcel & Building Example

The exampleParcels.xsd schema show in listing F.3.5.1 defines types used to show how area features may share boundaries by using composite curves. The gml:surfaceProperty is the only new geometry property element explicitly referenced in exampleRoad.xsd.

Listing F.3.5.1: exampleParcels.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/examples" xmlns:ex="http://www.opengis.net/examples"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:gml="http://www.opengis.net/gml"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" version="0.0">
  <import namespace="http://www.opengis.net/gml" schemaLocation="../../base/feature.xsd"/>
  <element name="District" type="ex:DistrictType" substitutionGroup="gml:FeatureCollection"/>
  <element name="Parcel" type="ex:ParcelType" substitutionGroup="gml:_Feature"/>
  <element name="Building" type="ex:BuildingType" substitutionGroup="gml:_Feature"/>
  <complexType name="DistrictType">
    <complexContent>
      <extension base="gml:FeatureCollectionType"/>
    </complexContent>
  </complexType>
  <complexType name="ParcelType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:surfaceProperty"/>
        </sequence>
      </extension>
    </complexContent>
```

```
  </complexType>
  <complexType name="BuildingType">
    <complexContent>
      <extension base="gml:AbstractFeatureType">
        <sequence>
          <element ref="gml:surfaceProperty"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</schema>
```

However, gml:surfaceProperty contains or points to a element substitutable for gml:_Surface, in this case gml:Polygon elements for both parcels and the building. The polygon boundaries (the example uses surfaces with only exterior boundaries) are described by gml:Ring elements. A ring is structurally like a closed composite curve and consists of one or more curves (elements substituable for gml:_Curve). In this example only gml:LineString is used. The important part is that curves like “c5” or “c1”, which are part of the boundary of more than one surface, are shared by the features and not represented redundantly.

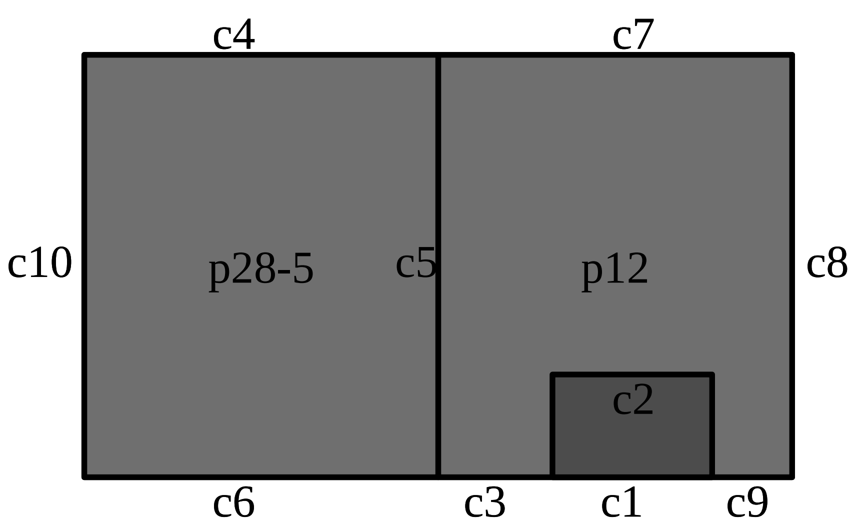


Figure F.3.5.3 – Parcels and Building Example

Listing F.3.5.2: exampleParcels.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<ex:District xmlns:ex="http://www.opengis.net/examples" xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/examples
exampleParcels.xsd ">
  <description>Example of two adjacent parcels and a building on a parcel sharing their boundary
geometry.</description>
  <name>Parcels</name>
```

```
<boundedBy>
  <Envelope srsName="somelistofcrs.xml#1234">
    <pos>0 0</pos>
    <pos>50 50</pos>
  </Envelope>
</boundedBy>
<featureMember>
  <ex:Parcel gml:id="p28-5">
    <surfaceProperty>
      <Polygon srsName="somelistofcrs.xml#1234">
        <exterior>
          <Ring>
            <curveMember>
              <LineString gml:id="c10">
                <coordinates>...</coordinates>
              </LineString>
            </curveMember>
            <curveMember>
              <LineString gml:id="c4">
                <coordinates>...</coordinates>
              </LineString>
            </curveMember>
            <curveMember>
              <LineString gml:id="c5">
                <coordinates>...</coordinates>
              </LineString>
            </curveMember>
            <curveMember>
              <LineString gml:id="c6">
                <coordinates>...</coordinates>
              </LineString>
            </curveMember>
          </Ring>
        </exterior>
      </Polygon>
    </surfaceProperty>
  </ex:Parcel>
</featureMember>
<featureMember>
  <ex:Parcel gml:id="p12">
    <surfaceProperty>
      <Polygon srsName="somelistofcrs.xml#1234">
        <exterior>
          <Ring>
            <curveMember>
              <OrientableCurve gml:id="c5m" orientation="-">
                <baseCurve xlink:href="#c5"/>
              </OrientableCurve>
            </curveMember>
            <curveMember>
              <LineString gml:id="c7">
                <coordinates>...</coordinates>
              </LineString>
            </curveMember>
            <curveMember>
              <LineString gml:id="c8">
                <coordinates>...</coordinates>
              </LineString>
            </curveMember>
            <curveMember>
              <LineString gml:id="c9">
                <coordinates>...</coordinates>
              </LineString>
            </curveMember>
```

```
              <curveMember>
                <LineString gml:id="c1">
                  <coordinates>...</coordinates>
                </LineString>
              </curveMember>
              <curveMember>
                <LineString gml:id="c3">
                  <coordinates>...</coordinates>
                </LineString>
              </curveMember>
            </Ring>
          </exterior>
        </Polygon>
      </surfaceProperty>
    </ex:Parcel>
  </featureMember>
  <featureMember>
    <ex:Building gml:id="building">
      <surfaceProperty>
        <Polygon srsName="somelistofcrs.xml#1234">
          <exterior>
            <Ring>
              <curveMember xlink:href="#c1"/>
              <curveMember>
                <LineString gml:id="c2">
                  <coordinates>...</coordinates>
                </LineString>
              </curveMember>
            </Ring>
          </exterior>
        </Polygon>
      </surfaceProperty>
    </ex:Building>
  </featureMember>
</ex:District>
```

Annex F
(Informative)

Guidelines for Subsetting GML Schemas

F.1 General

An automated approach is recommended for subsetting GML schemas. This annex contains an informative XSLT reference implementation of a GML schema subset tool. The tool consists of three XSLT stylesheets in the "xslt" subdirectory of the "applications" subdirectory of the gml3.0 directory structure:

- gml3.0
  - applications
    - xslt
  - base
  - smil
  - wfs
  - xlink

The three stylesheets are show in clauses F.2, F.3 and F.4 below.

To create a GML subset schema using this tool:

1. Transform feature.xsd using depends.xslt and an XSLT processor to produce gml.dep, shown in clause F.5 below, which lists the dependencies among all GML global types and elements.

2. If the XSLT processor you are using cannot pass parameters to a stylesheet being processed, edit gmlSubset.xslt, and change the "wanted" parameter to contain a comma separated list (with a trailing comma) of the namespace-qualified global types and elements you want in your GML subset schema. For example, change

```
<xsl:param name="wanted">,</xsl:param>
```

to

```
<xsl:param name="wanted">gml:featureProperty,gml:lineStringProperty,gml:polygonProperty,</xsl:param>
```

3. Transform gml.dep using gmlSubset.xslt, a parameter named "wanted" set to a comma separated list (with a trailing comma) of the namespace qualified global types and elements you want in your GML subset schema, and an XSLT

processor to produce gmlSubset.xsd, which will contain the global types and elements specified in the “wanted” parameter and all of the global types and elements on which they directly or indirectly use.

4. If your “wanted” list did not include or depend on any elements or types from the namespaces named “smil20” or smil20lang, you may wish to edit the generated schema to remove the following namespace definitions from the schema element:

   xmlns:smil20="http://www.w3.org/2001/SMIL20/"

   xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"

   xmlns:x="http://www.w3.org/XML/1998/namespace"

5.

The generated gmlSubset.xsd will include imports for the namespaces named “xlink”, “smil20” and “smil20lang” if your “wanted” list included or depended on any elements or types from the corresponding namespaces. Otherwise, it is a stand-alone GML subset schema that conforms to the requirements in clause 7.14 for GML profiles.

F.2 depends.xslt

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xlink="http://www.w3.org/1999/xlink">
    <!-- ============================================================
    This stylesheet is designed to be used on gml.xsd to produce gml.dep
    for use by the gml schema subset utility gmlSubset.xslt to produce a specialized
    gmlSubset.xsd that contains only the specified types and elements, and the types
    and elements on which they depend.
    ============================================================-->
    <xsl:output method="xml" encoding="UTF-8" indent="yes"/>
    <xsl:include href="utility.xslt"/>
    <!-- NEWLINE = &#xA; -->
    <xsl:param
name="schemas">gml.xsd,observation.xsd,dynamicFeature.xsd,coverage.xsd,topology.xsd,defaultStyle.xsd,coord
inateReferenceSystems.xsd,feature.xsd,valueObjects.xsd,grids.xsd,geometryComplexes.xsd,datums.xsd,coordina
teSystems.xsd,coordinateOperations.xsd,geometryAggregates.xsd,referenceSystems.xsd,dataQuality.xsd,geomet
ryPrimitives.xsd,geometryBasic2d.xsd,direction.xsd,geometryBasic0d1d.xsd,measures.xsd,temporal.xsd,units.xsd,
dictionary.xsd,gmlBase.xsd,basicTypes.xsd,</xsl:param>
    <xsl:param name="allSchemas">
        <xsl:call-template name="getUniqueSchemaList">
            <xsl:with-param name="list" select="$schemas"/>
            <xsl:with-param name="usePre"></xsl:with-param>
        </xsl:call-template>
    </xsl:param>
    <xsl:template match="/">
        <xsl:param name="docName">gml.xsd</xsl:param>
        <xsl:param name="top" select="true()"/>
        <xsl:param name="tns" select="//xsd:schema/@targetNamespace"/>
        <xsl:param name="vers" select="//xsd:schema/@version"/>
        <xsl:variable name="ltns">
            <xsl:for-each select="//xsd:schema/namespace::*">
                <xsl:if test="local-name() != 'targetNamespace' and string() = $tns">
```

```
                        <xsl:value-of select="local-name()"/>
                    </xsl:if>
                </xsl:for-each>
            </xsl:variable>
            <xsl:variable name="tnsp">
                <xsl:choose>
                    <xsl:when test="$ltns = ''">
                        <xsl:call-template name="getTargetNameSpacePrefix">
                            <xsl:with-param name="list" select="$tns"/>
                        </xsl:call-template>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:value-of select="$ltns"/>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:variable>
            <xsl:text>&#xA;</xsl:text>
            <xsl:choose>
                <xsl:when test="$top">
                    <xsl:text disable-output-escaping="yes">&lt;depends version="</xsl:text><xsl:value-of
select="$vers"/><xsl:text disable-output-escaping="yes">"&gt;</xsl:text>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:for-each select="/xsd:schema">
                        <xsl:for-each select="xsd:complexType | xsd:group | xsd:simpleType | xsd:element |
xsd:attribute | xsd:attributeGroup">
                            <xsl:variable name="type" select="local-name()"/>
                            <xsl:choose>
                                <xsl:when test="$type = 'complexType' ">
                                    <xsl:call-template name="complexType">
                                        <xsl:with-param name="docName" select="$docName"/>
                                        <xsl:with-param name="targetNamespace" select="$tnsp"/>
                                    </xsl:call-template>
                                </xsl:when>
                                <xsl:when test="$type = 'group' ">
                                    <xsl:call-template name="complexType">
                                        <xsl:with-param name="docName" select="$docName"/>
                                        <xsl:with-param name="targetNamespace" select="$tnsp"/>
                                    </xsl:call-template>
                                </xsl:when>
                                <xsl:when test="$type = 'simpleType' ">
                                    <xsl:call-template name="simpleType">
                                        <xsl:with-param name="docName" select="$docName"/>
                                        <xsl:with-param name="targetNamespace" select="$tnsp"/>
                                    </xsl:call-template>
                                </xsl:when>
                                <xsl:when test="$type = 'element' ">
                                    <xsl:call-template name="globalElement">
                                        <xsl:with-param name="docName" select="$docName"/>
                                        <xsl:with-param name="targetNamespace" select="$tnsp"/>
                                    </xsl:call-template>
                                </xsl:when>
                                <xsl:when test="$type = 'attribute' ">
                                    <xsl:call-template name="globalAtt">
                                        <xsl:with-param name="docName" select="$docName"/>
                                        <xsl:with-param name="targetNamespace" select="$tnsp"/>
                                    </xsl:call-template>
                                </xsl:when>
                                <xsl:when test="$type = 'attributeGroup' ">
                                    <xsl:call-template name="globalAtt">
                                        <xsl:with-param name="docName" select="$docName"/>
                                        <xsl:with-param name="targetNamespace" select="$tnsp"/>
                                    </xsl:call-template>
                                </xsl:when>
```

```
                              <xsl:otherwise/>
                         </xsl:choose>
                    </xsl:for-each>
               </xsl:for-each>
          </xsl:otherwise>
     </xsl:choose>
     <xsl:if test="$top">
          <xsl:call-template name="dependSchemas">
               <xsl:with-param name="list" select="$allSchemas"/>
          </xsl:call-template>
          <xsl:text disable-output-escaping="yes">&#xA;&lt;/depends&gt;&#xA;</xsl:text>
     </xsl:if>
</xsl:template>
<!-- ========================================================== -->
<xsl:template name="complexType">
     <xsl:param name="docName"/>
     <xsl:param name="targetNamespace"/>
     <xsl:variable name="name" select="@name"/>
     <xsl:if test="$name">
          <xsl:element name="def">
               <xsl:attribute name="name"><xsl:value-of select="$targetNamespace"/>:<xsl:value-of
select="$name"/></xsl:attribute>
               <xsl:attribute name="doc"><xsl:value-of select="$docName"/></xsl:attribute>
               <xsl:variable name="uses">
                    <xsl:apply-templates select="./xsd:complexContent|./xsd:simpleContent"/>
                    <xsl:call-template name="EltAndAtt"/>
               </xsl:variable>
               <!-- USES <xsl:value-of select="$uses"/> -->
               <xsl:call-template name="writeUses">
                    <xsl:with-param name="list" select="$uses"/>
               </xsl:call-template>
          </xsl:element>
     </xsl:if>
</xsl:template>
<!-- ========================================================== -->
<xsl:template match="xsd:complexContent">
     <xsl:for-each select="descendant::xsd:extension">
          <xsl:value-of select="@base"/>
          <xsl:text>?extension|</xsl:text>
     </xsl:for-each>
     <xsl:for-each select="descendant::xsd:restriction">
          <xsl:value-of select="@base"/>
          <xsl:text>?restriction|</xsl:text>
     </xsl:for-each>
</xsl:template>
<!-- ========================================================== -->
<xsl:template match="xsd:simpleContent">
     <xsl:for-each select="descendant::xsd:extension">
          <xsl:value-of select="@base"/>
          <xsl:text>?extension|</xsl:text>
     </xsl:for-each>
     <xsl:for-each select="descendant::xsd:restriction">
          <xsl:value-of select="@base"/>
          <xsl:text>?restriction|</xsl:text>
     </xsl:for-each>
</xsl:template>
<!-- ========================================================== -->
<xsl:template name="EltAndAtt">
     <xsl:for-each select="descendant::xsd:element | descendant::xsd:group | descendant::xsd:attribute |
descendant::xsd:attributeGroup">
          <xsl:variable name="name" select="@type | @ref"/>
          <xsl:if test="$name and contains($name,':')">
               <xsl:value-of select="$name"/>
               <xsl:text>|</xsl:text>
```

```
            </xsl:if>
        </xsl:for-each>
    </xsl:template>
    <!-- ============================================================ -->
    <xsl:template name="simpleType">
        <xsl:param name="docName"/>
        <xsl:param name="targetNamespace"/>
        <xsl:variable name="name" select="@name"/>
        <xsl:if test="$name">
            <xsl:element name="def">
                <xsl:attribute name="name"><xsl:value-of select="$targetNamespace"/>:<xsl:value-of
select="$name"/></xsl:attribute>
                <xsl:attribute name="doc"><xsl:value-of select="$docName"/></xsl:attribute>
                <!-- SIMPLE <xsl:copy-of select="."/>-->
                <xsl:variable name="uses">
                    <xsl:for-each select="xsd:union">
                    <!-- UNION <xsl:value-of select="@memberTypes"/> -->
                        <xsl:variable name="members" select="@memberTypes"/>
                        <xsl:if test="$members">
                            <xsl:value-of select="translate($members,' ','|')"/>
                            <xsl:text>|</xsl:text>
                        </xsl:if>
                    </xsl:for-each>
                    <xsl:for-each select="xsd:list">
                        <xsl:variable name="items" select="@itemType"/>
                        <xsl:if test="$items">
                            <xsl:value-of select="$items"/>
                            <xsl:text>|</xsl:text>
                        </xsl:if>
                    </xsl:for-each>
                </xsl:variable>
                 <!-- USES <xsl:value-of select="$uses"/> -->
                <xsl:call-template name="writeUses">
                    <xsl:with-param name="list" select="$uses"/>
                </xsl:call-template>
            </xsl:element>
        </xsl:if>
    </xsl:template>
    <!-- ============================================================ -->
    <xsl:template name="globalElement">
        <xsl:param name="docName"/>
        <xsl:param name="targetNamespace"/>
        <xsl:variable name="name" select="@name"/>
        <xsl:if test="$name">
            <xsl:element name="def">
                <xsl:attribute name="name"><xsl:value-of select="$targetNamespace"/>:<xsl:value-of
select="$name"/></xsl:attribute>
                <xsl:attribute name="doc"><xsl:value-of select="$docName"/></xsl:attribute>
                <xsl:variable name="uses">
                    <xsl:variable name="type" select="@type"/>
                    <xsl:if test="$type and contains($type,':')">
                        <xsl:value-of select="$type"/>
                        <xsl:text>|</xsl:text>
                    </xsl:if>
                    <xsl:variable name="sub" select="@substitutionGroup"/>
                    <xsl:if test="$sub">
                        <xsl:value-of select="$sub"/>
                        <xsl:text>|</xsl:text>
                    </xsl:if>
                </xsl:variable>
                <!-- USES <xsl:value-of select="$uses"/> -->
                <xsl:call-template name="writeUses">
                    <xsl:with-param name="list" select="$uses"/>
                </xsl:call-template>
```

```
            </xsl:element>
        </xsl:if>
    </xsl:template>
    <!-- ============================================================ -->
    <xsl:template name="globalAtt">
        <xsl:param name="docName"/>
        <xsl:param name="targetNamespace"/>
        <xsl:variable name="name" select="@name"/>
        <xsl:if test="$name">
            <xsl:element name="def">
                <xsl:attribute name="name"><xsl:value-of select="$targetNamespace"/>:<xsl:value-of
select="$name"/></xsl:attribute>
                <xsl:attribute name="doc"><xsl:value-of select="$docName"/></xsl:attribute>
                <xsl:variable name="uses">
                    <xsl:variable name="type" select="@type"/>
                    <xsl:if test="$type and contains($type,':')">
                        <xsl:value-of select="$type"/>
                        <xsl:text>|</xsl:text>
                    </xsl:if>
                    <xsl:call-template name="EltAndAtt"/>
                </xsl:variable>
                <!-- USES <xsl:value-of select="$uses"/> -->
                <xsl:call-template name="writeUses">
                    <xsl:with-param name="list" select="$uses"/>
                </xsl:call-template>
            </xsl:element>
        </xsl:if>
    </xsl:template>
    <!-- ============================================================ -->
    <xsl:template name="writeUses">
        <xsl:param name="list"/>
        <xsl:if test="$list != ''">
            <xsl:variable name="first" select="substring-before($list, '|')"/>
            <xsl:variable name="eor" select="substring-after($first, '?')"/>
            <xsl:variable name="use">
                <xsl:choose>
                    <xsl:when test="contains($first, '?')">
                        <xsl:value-of select="substring-before($first, '?')"/>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:value-of select="$first"/>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:variable>
            <xsl:variable name="testp">
                <xsl:value-of select="$use"/>
                <xsl:text>|</xsl:text>
            </xsl:variable>
            <xsl:variable name="testq">
                <xsl:value-of select="$use"/>
                <xsl:text>?</xsl:text>
            </xsl:variable>
            <xsl:variable name="rest" select="substring-after($list, '|')"/>
            <xsl:choose>
                <xsl:when test="contains($rest, $testp)"/>
                <xsl:when test="contains($rest, $testq)"/>
                <xsl:when test="$use = ''"/>
                <xsl:otherwise>
                    <xsl:element name="uses">
                        <xsl:attribute name="name"><xsl:value-of select="$use"/></xsl:attribute>
                        <xsl:if test="$eor != ''">
                            <xsl:attribute name="derivation"><xsl:value-of select="$eor"/> </xsl:attribute>
                        </xsl:if>
                    </xsl:element>
```

```
                </xsl:otherwise>
            </xsl:choose>
            <xsl:call-template name="writeUses">
                <xsl:with-param name="list" select="$rest"/>
            </xsl:call-template>
        </xsl:if>
    </xsl:template>
    <!-- ==================================================== -->
    <xsl:template name="dependSchemas">
        <xsl:param name="list"/>
        <xsl:if test="$list != ''">
            <xsl:variable name="first" select="substring-before($list, ',')"/>
            <xsl:variable name="rest" select="substring-after($list, ',')"/>
            <xsl:apply-templates select="document($first, /)">
                <xsl:with-param name="docName" select="$first"/>
                <xsl:with-param name="top" select="false()"/>
            </xsl:apply-templates>
            <xsl:choose>
                <xsl:when test="contains($rest,',')">
                    <xsl:call-template name="dependSchemas">
                        <xsl:with-param name="list" select="$rest"/>
                    </xsl:call-template>
                </xsl:when>
                <xsl:otherwise/>
            </xsl:choose>
        </xsl:if>
    </xsl:template>
    <!-- ========================================================== -->
    <!-- ========================================================== -->
</xsl:stylesheet>
```


F.3 gmlSubset.xslt

```
<xsl:stylesheet version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:xlink="http://www.w3.org/1999/xlink">
    <xsl:output method="xml" encoding="UTF-8" indent="yes"/>
    <!-- ============================================================
    This stylesheet is designed to be used on gml.dep (produced from
    gml.xsd by depends.xslt) to produce a specialized gmlSubset.xsd that
    contains only the types and elements specified in the "wanted" parameter,
    and the types and elements on which they depend.  Note that the type and
    element items in the "wanted" parameter must include namespace prefixes,
    and that they must be separated by commas, including a trailing comma after
    the last item.
    ============================================================-->
    <xsl:include href="utility.xslt"/>
    <xsl:param name="baseUri" select="document('../../base/gml.xsd')"/>
    <!-- sample1 <xsl:param
name="wanted">gml:featureProperty,gml:lineStringProperty,gml:polygonProperty,</xsl:param> -->
    <!-- sample2 <xsl:param name="wanted">gml:GeographicCRS,gml:_Coverage,gml:track,</xsl:param> -->
    <!-- sample3 <xsl:param
name="wanted">gml:_FeatureCollection,gml:ItemStyleDescriptorType,gml:FeatureConstraintType,</xsl:param> --
>
    <xsl:param
name="wanted">gml:metaDataProperty,gml:_association,gml:members,gml:Array,gml:curveProperty,gml:LineStrin
g,gml:LinearRing,gml:exterior,gml:interior,gml:surfaceMember,gml:surfaceProperty,gml:multiSurfaceProperty,gml:
directedNode,gml:directedEdge,gml:directedFace,gml:IsolatedProperty,gml:featureProperty,gml:featureMembers,g
ml:_FeatureCollection,gml:featureMember,gml:BaseStyleDescriptorType,</xsl:param>
    <xsl:template match="/">
        <xsl:variable name="wantedList">
            <xsl:call-template name="getWantedList">
                <xsl:with-param name="list" select="$wanted"/>
                <xsl:with-param name="from">BEGIN</xsl:with-param>
                <xsl:with-param name="depth">0</xsl:with-param>
            </xsl:call-template>
        </xsl:variable>
        <xsl:variable name="vers"  select="//depends/@version"/>
        <schema targetNamespace="http://www.opengis.net/gml" xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:sch="http://www.ascc.net/xml/schematron" xmlns:gml="http://www.opengis.net/gml"
xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:smil20="http://www.w3.org/2001/SMIL20/"
xmlns:smil20lang="http://www.w3.org/2001/SMIL20/Language"
xmlns:x="http://www.w3.org/XML/1998/namespace" elementFormDefault="qualified" version="{$vers}">
            <annotation>
                <documentation>GML Subset schema for <xsl:value-of select="$wanted"/>  written by
gmlSubset.xslt. </documentation>
            </annotation>

            <xsl:if test="contains($wantedList,'xlink:')">
                <import namespace="http://www.w3.org/1999/xlink" schemaLocation="../../xlink/xlinks.xsd"/>
            </xsl:if>
            <xsl:if test="contains($wantedList,'smil20:')">
                <import namespace="http://www.w3.org/2001/SMIL20/" schemaLocation="../../smil/smil20.xsd"/>
            </xsl:if>
            <xsl:if test="contains($wantedList,'smil20lang:')">
                <import namespace="http://www.w3.org/2001/SMIL20/Language"
schemaLocation="../../smil/smil20-language.xsd"/>
            </xsl:if>
```



```
        <xsl:call-template name="writeWantedList">
            <xsl:with-param name="list" select="$wantedList"/>
        </xsl:call-template>
    </schema>
</xsl:template>
<!-- ============================================================ -->
<xsl:template name="getDocName">
    <xsl:param name="wanted"/>
    <xsl:for-each select="//depends/def[@name=$wanted]">
        <xsl:value-of select="@doc"/>
    </xsl:for-each>
</xsl:template>
<!-- ============================================================ -->
<xsl:template name="getUses">
    <xsl:param name="wanted"/>
    <xsl:for-each select="//depends/def[@name=$wanted]">
        <xsl:for-each select="uses">
            <xsl:value-of select="@name"/>
            <xsl:text>,</xsl:text>
        </xsl:for-each>
    </xsl:for-each>
</xsl:template>
<!-- ============================================================ -->
<xsl:template name="writeWanted">
    <xsl:param name="wanted"/>
    <xsl:choose>
        <xsl:when test="contains($wanted,'xlink:') or contains($wanted,'wfs:') or contains($wanted,'smil20')">
            <!-- XLINK <xsl:value-of select="$wanted"/> -->
        </xsl:when>
        <xsl:otherwise>
            <!-- OTHER <xsl:value-of select="$wanted"/> -->
            <xsl:variable name="docName">
                <xsl:call-template name="getDocName">
                    <xsl:with-param name="wanted" select="$wanted"/>
                </xsl:call-template>
            </xsl:variable>
            <xsl:variable name="localName">
                <xsl:call-template name="removePrefix">
                    <xsl:with-param name="name" select="$wanted"/>
                    <xsl:with-param name="pre">:</xsl:with-param>
                </xsl:call-template>
            </xsl:variable>
            <xsl:call-template name="Separator"/>
            <xsl:for-each select="document($docName,$baseUri)">
                <xsl:for-each select="//xsd:schema/*[@name = $localName]">
                    <xsl:copy-of select="."/>
                </xsl:for-each>
            </xsl:for-each>
        </xsl:otherwise>
    </xsl:choose>
</xsl:template>
<!-- ============================================================ -->
<xsl:template name="writeWantedList">
    <xsl:param name="list"/>
    <xsl:if test="$list != ''">
        <xsl:variable name="first" select="substring-before($list, ',')"/>
        <xsl:variable name="rest" select="substring-after($list, ',')"/>
        <xsl:call-template name="writeWanted">
            <xsl:with-param name="wanted" select="$first"/>
        </xsl:call-template>
        <xsl:if test="contains($rest,',')">
            <xsl:call-template name="writeWantedList">
                <xsl:with-param name="list" select="$rest"/>
            </xsl:call-template>
```

```
            </xsl:if>
        </xsl:if>
    </xsl:template>
    <!-- ============================================================ -->
    <xsl:template name="getWantedList">
        <xsl:param name="list"/>
        <xsl:param name="seen"/>
        <xsl:param name="from"/>
        <xsl:param name="depth"/>
        <xsl:if test="$list != ''">
            <xsl:variable name="first" select="substring-before($list, ',')"/>
            <xsl:variable name="firstSep" select="concat($first,',')"/>
            <xsl:variable name="rest" select="substring-after($list, ',')"/>
            <xsl:choose>
                <xsl:when test="contains($seen,$firstSep)">
                    <xsl:call-template name="getWantedList">
                        <xsl:with-param name="list" select="$rest"/>
                        <xsl:with-param name="seen" select="$seen"/>
                        <xsl:with-param name="from">REST</xsl:with-param>
                        <xsl:with-param name="depth" select="$depth + 1"/>
                    </xsl:call-template>
                </xsl:when>
                <xsl:otherwise>
                    <xsl:value-of select="$firstSep"/>
                    <xsl:variable name="usesList">
                        <xsl:call-template name="getUses">
                            <xsl:with-param name="wanted" select="$first"/>
                        </xsl:call-template>
                    </xsl:variable>
                    <xsl:variable name="toDo" select="concat($usesList,$rest)"/>
                    <xsl:variable name="nowSeen" select="concat($seen,$firstSep)"/>
                    <xsl:call-template name="getWantedList">
                        <xsl:with-param name="list" select="$toDo"/>
                        <xsl:with-param name="seen" select="$nowSeen"/>
                        <xsl:with-param name="from">USES</xsl:with-param>
                        <xsl:with-param name="depth" select="$depth + 1"/>
                    </xsl:call-template>
                </xsl:otherwise>
            </xsl:choose>
        </xsl:if>
    </xsl:template>
    <!-- ============================================================ -->
</xsl:stylesheet>
```


F.4 utility.xslt

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
	<xsl:output method="xml" encoding="UTF-8" indent="yes"/>
	<!-- ====================================================== -->
	<xsl:template name="getTargetNameSpacePrefix">
		<xsl:param name="list"/>
		<xsl:if test="$list != ''">
			<xsl:variable name="first" select="substring-before($list, '/')"/>
			<xsl:variable name="rest" select="substring-after($list, '/')"/>
			<xsl:choose>
				<xsl:when test="contains($rest,'/')">
					<xsl:call-template name="getTargetNameSpacePrefix">
						<xsl:with-param name="list" select="$rest"/>
					</xsl:call-template>
				</xsl:when>
				<xsl:when test="$rest =''">
					<xsl:value-of select="$first"/>
				</xsl:when>
				<xsl:otherwise>
					<xsl:value-of select="$rest"/>
				</xsl:otherwise>
			</xsl:choose>
		</xsl:if>
	</xsl:template>
	<!-- ====================================================== -->
	<xsl:template name="getPathPrefix">
		<xsl:param name="file"/>
		<xsl:if test="contains($file,'/')">
			<xsl:variable name="pre" select="substring-before($file,'/')"/>
			<xsl:variable name="suf" select="substring-after($file,'/')"/>
			<xsl:choose>
				<xsl:when test="contains($suf,'/')">
					<xsl:value-of select="$pre"/><xsl:text>/</xsl:text>
					<xsl:call-template name="getPathPrefix">
						<xsl:with-param name="file" select="$suf"/>
					</xsl:call-template>
				</xsl:when>
				<xsl:otherwise>
					<xsl:variable name="path">
						<xsl:call-template name="removeSuffix">
							<xsl:with-param name="name" select="$file"/>
							<xsl:with-param name="suf" select="$suf"/>
						</xsl:call-template>
					</xsl:variable>
					<xsl:value-of select="$path"/>
				</xsl:otherwise>
			</xsl:choose>
		</xsl:if>
	</xsl:template>
	<!-- ====================================================== -->
	<xsl:template name="removePrefix">
		<xsl:param name="name"/>
		<xsl:param name="pre"/>
		<xsl:variable name="npName">
			<xsl:choose>
				<xsl:when test="contains($name,$pre)">
```

```
                <xsl:value-of select="substring-after($name,$pre)"/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="$name"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:value-of select="$npName"/>
</xsl:template>
<!-- =================================================== -->
<xsl:template name="removeSuffix">
    <xsl:param name="name"/>
    <xsl:param name="suf"/>
    <xsl:variable name="nsName">
        <xsl:choose>
            <xsl:when test="contains($name,$suf)">
                <xsl:value-of select="substring-before($name,$suf)"/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="$name"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:value-of select="$nsName"/>
</xsl:template>
<!-- =================================================== -->
<xsl:template name="lowerLeading">
    <xsl:param name="name"/>
    <xsl:variable name="ch1" select="substring($name, 1, 1)"/>
    <xsl:variable name="lc1"
select="translate($ch1,'ABCDEFGHIJKLMNOPQRSTUVWXYZ','abcdefghijklmnopqrstuvwxyz')"/>
    <xsl:value-of select="concat($lc1, substring($name, 2))"/>
</xsl:template>
<!-- =================================================== -->
<xsl:template name="uniqueList">
    <xsl:param name="list"/>
    <xsl:param name="sep"/>
    <xsl:param name="seen"/>
    <xsl:param name="pre">../base/</xsl:param>
    <xsl:if test="$list != ''">
        <xsl:variable name="first" select="substring-before($list, $sep)"/>
        <xsl:variable name="firstSep" select="concat($first,$sep)"/>
        <xsl:variable name="rest" select="substring-after($list, $sep)"/>
        <xsl:choose>
            <xsl:when test="contains($seen,$firstSep)">
                <xsl:call-template name="uniqueList">
                    <xsl:with-param name="list" select="$rest"/>
                    <xsl:with-param name="sep" select="$sep"/>
                    <xsl:with-param name="seen" select="$seen"/>
                </xsl:call-template>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="$firstSep"/>
                <xsl:variable name="nowSeen" select="concat($seen, $firstSep)"/>
                <xsl:call-template name="uniqueList">
                    <xsl:with-param name="list" select="$rest"/>
                    <xsl:with-param name="sep" select="$sep"/>
                    <xsl:with-param name="seen" select="$nowSeen"/>
                </xsl:call-template>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:if>
</xsl:template>
<!-- =================================================== -->
```



```
<xsl:template name="getIncludedDocs">
    <xsl:param name="docName"/>
    <xsl:param name="usePre"/>
    <xsl:param name="seenList"/>
    <xsl:param name="sep">,</xsl:param>
    <xsl:value-of select="$docName"/>
    <xsl:text>,</xsl:text>
    <xsl:variable name="pathPre">
        <xsl:call-template name="getPathPrefix">
            <xsl:with-param name="file" select="$docName"/>
        </xsl:call-template>
    </xsl:variable>
    <xsl:variable name="callPathPre">
        <xsl:choose>
            <xsl:when test="$pathPre = '' or $pathPre = './' ">
                <xsl:value-of select="$usePre"/>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="$pathPre"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:for-each select="document($docName, / )">
        <xsl:for-each select="//xsd:include | //xsd:import">
            <xsl:variable name="iDoc" select="@schemaLocation"/>
            <xsl:variable name="iPathPre">
                <xsl:call-template name="getPathPrefix">
                    <xsl:with-param name="file" select="$iDoc"/>
                </xsl:call-template>
            </xsl:variable>
            <xsl:variable name="iDocSuf">
                <xsl:call-template name="removePrefix">
                    <xsl:with-param name="name" select="$iDoc"/>
                    <xsl:with-param name="pre" select="$iPathPre"/>
                </xsl:call-template>
            </xsl:variable>
            <xsl:variable name="usePathPre">
                <xsl:choose>
                    <xsl:when test="$iPathPre = '' or $iPathPre = './' ">
                        <xsl:value-of select="$callPathPre"/>
                    </xsl:when>
                    <xsl:otherwise>
                        <xsl:value-of select="$iPathPre"/>
                    </xsl:otherwise>
                </xsl:choose>
            </xsl:variable>
            <xsl:variable name="uDoc">
                <xsl:value-of select="concat($usePathPre,$iDocSuf)"/>
            </xsl:variable>
            <xsl:variable name="uDocSep">
                <xsl:value-of select="concat($uDoc,$sep)"/>
            </xsl:variable>
            <xsl:if test="not(contains($seenList,$uDocSep))">
                <xsl:variable name="seenListPlus" select="concat($seenList,$uDocSep)"/>
                <xsl:call-template name="getIncludedDocs">
                    <xsl:with-param name="docName" select="$uDoc"/>
                    <xsl:with-param name="usePre" select="$usePathPre"/>
                    <xsl:with-param name="seenList" select="$seenListPlus"/>
                </xsl:call-template>
            </xsl:if>
        </xsl:for-each>
    </xsl:for-each>
</xsl:template>
<!-- ================================================== -->
```

```
    <xsl:template name="getDocumentList">
        <xsl:param name="list"/>
        <xsl:param name="seenList"/>
        <xsl:param name="usePre"/>
        <xsl:if test="$list != ''">
            <xsl:variable name="first" select="substring-before($list, ',')"/>
            <xsl:variable name="rest" select="substring-after($list, ',')"/>
            <xsl:variable name="included">
                <xsl:call-template name="getIncludedDocs">
                    <xsl:with-param name="docName" select="$first"/>
                    <xsl:with-param name="usePre" select="$usePre"/>
                    <xsl:with-param name="seenList" select="$seenList"/>
                </xsl:call-template>
            </xsl:variable>
            <xsl:value-of select="$included"/>
            <xsl:variable name="seenListIncluded" select="concat($seenList,$included)"/>
            <xsl:if test="contains($rest,',')">
                <xsl:call-template name="getDocumentList">
                    <xsl:with-param name="list" select="$rest"/>
                    <xsl:with-param name="seenList" select="$seenListIncluded"/>
                    <xsl:with-param name="usePre" select="$usePre"/>
                </xsl:call-template>
            </xsl:if>
        </xsl:if>
    </xsl:template>
    <!-- ==================================================== -->
    <xsl:template name="getUniqueSchemaList">
        <xsl:param name="list"/>
        <xsl:param name="usePre"/>
        <xsl:variable name="allSchemas">
            <xsl:call-template name="getDocumentList">
                <xsl:with-param name="list" select="$list"/>
                <xsl:with-param name="usePre" select="$usePre"/>
            </xsl:call-template>
        </xsl:variable>
        <xsl:variable name="uniqueSchemas">
            <xsl:call-template name="uniqueList">
                <xsl:with-param name="list" select="$allSchemas"/>
                <xsl:with-param name="sep">,</xsl:with-param>
            </xsl:call-template>
        </xsl:variable>
        <xsl:value-of select="$uniqueSchemas"/>
    </xsl:template>
    <!-- ========================================================= -->
     <xsl:template name="Separator">
   <xsl:param name="comment" select="'=========='"/>
   <xsl:text>&#xA;</xsl:text>
   <xsl:text disable-output-escaping="yes">
      &lt;!-- =================</xsl:text><xsl:value-of select="$comment"/><xsl:text disable-output-
escaping="yes">==================== --&gt;
  </xsl:text>
    <!-- ========================================================= -->
  </xsl:template>


</xsl:stylesheet>
```

F.5 gml.dep


```xml
<?xml version="1.0" encoding="UTF-8"?>
<depends version="3.0c4">

<def name="gml:_TimeSlice" doc="dynamicFeature.xsd">
  <uses name="gml:AbstractTimeSliceType"/>
  <uses name="gml:_GML"/>
</def>
<def name="gml:dataSource" doc="dynamicFeature.xsd">
  <uses name="gml:StringOrRefType"/>
</def>
<def name="gml:AbstractTimeSliceType" doc="dynamicFeature.xsd">
  <uses name="gml:AbstractGMLType" derivation="extension"/>
  <uses name="gml:timeStamp"/>
  <uses name="gml:dataSource"/>
</def>
<def name="gml:MovingObjectStatus" doc="dynamicFeature.xsd">
  <uses name="gml:MovingObjectStatusType"/>
  <uses name="gml:_TimeSlice"/>
</def>
<def name="gml:status" doc="dynamicFeature.xsd">
  <uses name="gml:StringOrRefType"/>
</def>
<def name="gml:MovingObjectStatusType" doc="dynamicFeature.xsd">
  <uses name="gml:AbstractTimeSliceType" derivation="extension"/>
  <uses name="gml:location"/>
  <uses name="gml:DirectionPropertyType"/>
  <uses name="gml:MeasureType"/>
  <uses name="gml:status"/>
</def>
<def name="gml:history" doc="dynamicFeature.xsd">
  <uses name="gml:HistoryPropertyType"/>
</def>
<def name="gml:HistoryPropertyType" doc="dynamicFeature.xsd">
  <uses name="gml:_TimeSlice"/>
</def>
<def name="gml:track" doc="dynamicFeature.xsd">
  <uses name="gml:TrackType"/>
  <uses name="gml:history"/>
</def>
<def name="gml:TrackType" doc="dynamicFeature.xsd">
  <uses name="gml:HistoryPropertyType" derivation="restriction"/>
  <uses name="gml:MovingObjectStatus"/>
</def>
<def name="gml:dynamicProperties" doc="dynamicFeature.xsd">
  <uses name="gml:timeStamp"/>
  <uses name="gml:history"/>
  <uses name="gml:dataSource"/>
</def>
<def name="gml:DynamicFeatureType" doc="dynamicFeature.xsd">
  <uses name="gml:AbstractFeatureType" derivation="extension"/>
  <uses name="gml:dynamicProperties"/>
</def>
<def name="gml:DynamicFeatureCollectionType" doc="dynamicFeature.xsd">
```

```
  <uses name="gml:FeatureCollectionType" derivation="extension"/>
  <uses name="gml:dynamicProperties"/>
</def>

<def name="gml:_TimeObject" doc="temporal.xsd">
  <uses name="gml:AbstractTimeType"/>
  <uses name="gml:_GML"/>
</def>
<def name="gml:AbstractTimeType" doc="temporal.xsd">
  <uses name="gml:AbstractGMLType" derivation="extension"/>
</def>
<def name="gml:_TimePrimitive" doc="temporal.xsd">
  <uses name="gml:TimePrimitiveType"/>
  <uses name="gml:_TimeObject"/>
</def>
<def name="gml:TimePrimitiveType" doc="temporal.xsd">
  <uses name="gml:AbstractTimeType" derivation="extension"/>
</def>
<def name="gml:TimeInstant" doc="temporal.xsd">
  <uses name="gml:TimeInstantType"/>
  <uses name="gml:_TimePrimitive"/>
</def>
<def name="gml:TimeInstantType" doc="temporal.xsd">
  <uses name="gml:TimePrimitiveType" derivation="extension"/>
  <uses name="gml:timePosition"/>
</def>
<def name="gml:begin" doc="temporal.xsd">
  <uses name="gml:TimeInstantPropertyType"/>
</def>
<def name="gml:end" doc="temporal.xsd">
  <uses name="gml:TimeInstantPropertyType"/>
</def>
<def name="gml:TimeInstantPropertyType" doc="temporal.xsd">
  <uses name="gml:TimeInstant"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:TimePeriod" doc="temporal.xsd">
  <uses name="gml:TimePeriodType"/>
  <uses name="gml:_TimePrimitive"/>
</def>
<def name="gml:TimePeriodType" doc="temporal.xsd">
  <uses name="gml:TimePrimitiveType" derivation="extension"/>
  <uses name="gml:begin"/>
  <uses name="gml:end"/>
  <uses name="gml:_duration"/>
</def>
<def name="gml:_duration" doc="temporal.xsd">
  <uses name="gml:TimeDurationType"/>
</def>
<def name="gml:TimeDurationType" doc="temporal.xsd">
  <uses name="duration"/>
  <uses name="decimal"/>
</def>
<def name="gml:duration" doc="temporal.xsd">
  <uses name="gml:_duration"/>
</def>
<def name="gml:TemporalPositionType" doc="temporal.xsd">
  <uses name="dateTime"/>
  <uses name="date"/>
  <uses name="gYearMonth"/>
```

```
  <uses name="gYear"/>
  <uses name="anyURI"/>
  <uses name="decimal"/>
</def>
<def name="gml:timePosition" doc="temporal.xsd">
  <uses name="gml:TimePositionType"/>
</def>
<def name="gml:TimePositionType" doc="temporal.xsd">
  <uses name="gml:TemporalPositionType" derivation="extension"/>
  <uses name="gml:TimeIndeterminateValueType"/>
</def>
<def name="gml:TimeIndeterminateValueType" doc="temporal.xsd"/>
<def name="gml:interval" doc="temporal.xsd">
  <uses name="gml:TimeIntervalLengthType"/>
  <uses name="gml:_duration"/>
</def>
<def name="gml:TimeIntervalLengthType" doc="temporal.xsd">
  <uses name="decimal" derivation="extension"/>
  <uses name="gml:TimeUnitType"/>
</def>
<def name="gml:TimeUnitType" doc="temporal.xsd"/>
<def name="gml:timePrimitiveProperty" doc="temporal.xsd">
  <uses name="gml:TimePrimitivePropertyType"/>
</def>
<def name="gml:timeStamp" doc="temporal.xsd">
  <uses name="gml:TimePrimitivePropertyType"/>
  <uses name="gml:timePrimitiveProperty"/>
</def>
<def name="gml:TimePrimitivePropertyType" doc="temporal.xsd">
  <uses name="gml:_TimePrimitive"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:TimeReferenceSystem" doc="temporal.xsd">
  <uses name="gml:TimeReferenceSystemType"/>
  <uses name="gml:_GML"/>
</def>
<def name="gml:TimeReferenceSystemType" doc="temporal.xsd">
  <uses name="gml:DefinitionType" derivation="extension"/>
</def>
<def name="gml:TimeCoordinateSystem" doc="temporal.xsd">
  <uses name="gml:TimeCoordinateSystemType"/>
  <uses name="gml:TimeReferenceSystem"/>
</def>
<def name="gml:TimeCoordinateSystemType" doc="temporal.xsd">
  <uses name="gml:TimeReferenceSystemType" derivation="extension"/>
  <uses name="gml:TimeInstantPropertyType"/>
  <uses name="gml:UnitOfMeasureType"/>
  <uses name="gml:SignType"/>
</def>
<def name="gml:TimeOrdinalReferenceSystem" doc="temporal.xsd">
  <uses name="gml:TimeOrdinalReferenceSystemType"/>
  <uses name="gml:TimeReferenceSystem"/>
</def>
<def name="gml:TimeOrdinalReferenceSystemType" doc="temporal.xsd">
  <uses name="gml:TimeReferenceSystemType" derivation="extension"/>
  <uses name="gml:TimeOrdinalReferenceSystemMemberType"/>
</def>
<def name="gml:TimeOrdinalReferenceSystemMemberType" doc="temporal.xsd">
  <uses name="gml:TimeOrdinalEra"/>
</def>
```



```
<def name="gml:TimeOrdinalEra" doc="temporal.xsd">
  <uses name="gml:TimeOrdinalEraType"/>
  <uses name="gml:_GML"/>
</def>
<def name="gml:TimeOrdinalEraType" doc="temporal.xsd">
  <uses name="gml:AbstractGMLType" derivation="extension"/>
  <uses name="gml:begin"/>
  <uses name="gml:end"/>
  <uses name="gml:TimeOrdinalReferenceSystemMemberType"/>
</def>

<def name="gml:unitOfMeasure" doc="units.xsd">
  <uses name="gml:UnitOfMeasureType"/>
</def>
<def name="gml:UnitOfMeasureType" doc="units.xsd"/>
<def name="gml:UnitDefinition" doc="units.xsd">
  <uses name="gml:UnitDefinitionType"/>
  <uses name="gml:Definition"/>
</def>
<def name="gml:UnitDefinitionType" doc="units.xsd">
  <uses name="gml:DefinitionType" derivation="extension"/>
  <uses name="gml:quantityType"/>
  <uses name="gml:catalogSymbol"/>
</def>
<def name="gml:BaseUnit" doc="units.xsd">
  <uses name="gml:BaseUnitType"/>
  <uses name="gml:UnitDefinition"/>
</def>
<def name="gml:BaseUnitType" doc="units.xsd">
  <uses name="gml:UnitDefinitionType" derivation="extension"/>
  <uses name="gml:ReferenceType"/>
</def>
<def name="gml:DerivedUnit" doc="units.xsd">
  <uses name="gml:DerivedUnitType"/>
  <uses name="gml:UnitDefinition"/>
</def>
<def name="gml:DerivedUnitType" doc="units.xsd">
  <uses name="gml:UnitDefinitionType" derivation="extension"/>
  <uses name="gml:unitDerivation"/>
</def>
<def name="gml:ConventionalUnit" doc="units.xsd">
  <uses name="gml:ConventionalUnitType"/>
  <uses name="gml:UnitDefinition"/>
</def>
<def name="gml:ConventionalUnitType" doc="units.xsd">
  <uses name="gml:UnitDefinitionType" derivation="extension"/>
  <uses name="gml:conversionToPreferredUnit"/>
  <uses name="gml:roughConversionToPreferredUnit"/>
  <uses name="gml:unitDerivation"/>
</def>
<def name="gml:quantityType" doc="units.xsd">
  <uses name="gml:StringOrRefType"/>
</def>
<def name="gml:catalogSymbol" doc="units.xsd">
  <uses name="gml:CodeType"/>
</def>
<def name="gml:unitDerivation" doc="units.xsd">
  <uses name="gml:UnitDerivationType"/>
</def>
<def name="gml:UnitDerivationType" doc="units.xsd">
```



```
  <uses name="gml:unitTerm"/>
</def>
<def name="gml:unitTerm" doc="units.xsd">
  <uses name="gml:UnitTermType"/>
</def>
<def name="gml:UnitTermType" doc="units.xsd">
  <uses name="gml:UnitOfMeasureType" derivation="extension"/>
</def>
<def name="gml:conversionToPreferredUnit" doc="units.xsd">
  <uses name="gml:ConversionToPreferredUnitType"/>
</def>
<def name="gml:roughConversionToPreferredUnit" doc="units.xsd">
  <uses name="gml:ConversionToPreferredUnitType"/>
</def>
<def name="gml:ConversionToPreferredUnitType" doc="units.xsd">
  <uses name="gml:UnitOfMeasureType" derivation="extension"/>
  <uses name="gml:factor"/>
  <uses name="gml:formula"/>
</def>
<def name="gml:factor" doc="units.xsd"/>
<def name="gml:formula" doc="units.xsd">
  <uses name="gml:FormulaType"/>
</def>
<def name="gml:FormulaType" doc="units.xsd"/>

<def name="gml:Definition" doc="dictionary.xsd">
  <uses name="gml:DefinitionType"/>
  <uses name="gml:_GML"/>
</def>
<def name="gml:DefinitionType" doc="dictionary.xsd">
  <uses name="gml:AbstractGMLType" derivation="restriction"/>
  <uses name="gml:metaDataProperty"/>
  <uses name="gml:description"/>
  <uses name="gml:name"/>
  <uses name="gml:id"/>
</def>
<def name="gml:Dictionary" doc="dictionary.xsd">
  <uses name="gml:DictionaryType"/>
  <uses name="gml:Definition"/>
</def>
<def name="gml:DefinitionCollection" doc="dictionary.xsd">
  <uses name="gml:DictionaryType"/>
  <uses name="gml:Definition"/>
</def>
<def name="gml:DictionaryType" doc="dictionary.xsd">
  <uses name="gml:DefinitionType" derivation="extension"/>
  <uses name="gml:dictionaryEntry"/>
  <uses name="gml:indirectEntry"/>
</def>
<def name="gml:dictionaryEntry" doc="dictionary.xsd">
  <uses name="gml:DictionaryEntryType"/>
</def>
<def name="gml:definitionMember" doc="dictionary.xsd">
  <uses name="gml:DictionaryEntryType"/>
  <uses name="gml:dictionaryEntry"/>
</def>
<def name="gml:DictionaryEntryType" doc="dictionary.xsd">
  <uses name="gml:Definition"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
```

```
<def name="gml:indirectEntry" doc="dictionary.xsd">
  <uses name="gml:IndirectEntryType"/>
</def>
<def name="gml:IndirectEntryType" doc="dictionary.xsd">
  <uses name="gml:DefinitionProxy"/>
</def>
<def name="gml:DefinitionProxy" doc="dictionary.xsd">
  <uses name="gml:DefinitionProxyType"/>
  <uses name="gml:Definition"/>
</def>
<def name="gml:DefinitionProxyType" doc="dictionary.xsd">
  <uses name="gml:DefinitionType" derivation="extension"/>
  <uses name="gml:definitionRef"/>
</def>
<def name="gml:definitionRef" doc="dictionary.xsd">
  <uses name="gml:ReferenceType"/>
</def>

<def name="gml:_Object" doc="gmlBase.xsd"/>
<def name="gml:AbstractGMLType" doc="gmlBase.xsd">
  <uses name="gml:metaDataProperty"/>
  <uses name="gml:description"/>
  <uses name="gml:name"/>
  <uses name="gml:id"/>
</def>
<def name="gml:_GML" doc="gmlBase.xsd">
  <uses name="gml:AbstractGMLType"/>
  <uses name="gml:_Object"/>
</def>
<def name="gml:BagType" doc="gmlBase.xsd">
  <uses name="gml:AbstractGMLType" derivation="extension"/>
  <uses name="gml:member"/>
  <uses name="gml:members"/>
</def>
<def name="gml:Bag" doc="gmlBase.xsd">
  <uses name="gml:BagType"/>
  <uses name="gml:_GML"/>
</def>
<def name="gml:ArrayType" doc="gmlBase.xsd">
  <uses name="gml:AbstractGMLType" derivation="extension"/>
  <uses name="gml:members"/>
</def>
<def name="gml:Array" doc="gmlBase.xsd">
  <uses name="gml:ArrayType"/>
  <uses name="gml:_GML"/>
</def>
<def name="gml:AbstractMetaDataType" doc="gmlBase.xsd">
  <uses name="gml:id"/>
</def>
<def name="gml:_MetaData" doc="gmlBase.xsd">
  <uses name="gml:AbstractMetaDataType"/>
  <uses name="gml:_Object"/>
</def>
<def name="gml:GenericMetaDataType" doc="gmlBase.xsd">
  <uses name="gml:AbstractMetaDataType" derivation="extension"/>
</def>
<def name="gml:GenericMetaData" doc="gmlBase.xsd">
  <uses name="gml:GenericMetaDataType"/>
  <uses name="gml:_MetaData"/>
</def>
```

```
<def name="gml:_property" doc="gmlBase.xsd"/>
<def name="gml:AssociationType" doc="gmlBase.xsd">
  <uses name="gml:_Object"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_association" doc="gmlBase.xsd">
  <uses name="gml:AssociationType"/>
</def>
<def name="gml:_strictAssociation" doc="gmlBase.xsd">
  <uses name="gml:AssociationType"/>
</def>
<def name="gml:member" doc="gmlBase.xsd">
  <uses name="gml:AssociationType"/>
</def>
<def name="gml:ReferenceType" doc="gmlBase.xsd">
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_reference" doc="gmlBase.xsd">
  <uses name="gml:ReferenceType"/>
</def>
<def name="gml:ArrayAssociationType" doc="gmlBase.xsd">
  <uses name="gml:_Object"/>
</def>
<def name="gml:members" doc="gmlBase.xsd">
  <uses name="gml:ArrayAssociationType"/>
</def>
<def name="gml:MetaDataPropertyType" doc="gmlBase.xsd">
  <uses name="gml:_MetaData"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:metaDataProperty" doc="gmlBase.xsd">
  <uses name="gml:MetaDataPropertyType"/>
</def>
<def name="gml:id" doc="gmlBase.xsd"/>
<def name="gml:remoteSchema" doc="gmlBase.xsd"/>
<def name="gml:AssociationAttributeGroup" doc="gmlBase.xsd">
  <uses name="xlink:simpleLink"/>
  <uses name="gml:remoteSchema"/>
</def>
<def name="gml:name" doc="gmlBase.xsd">
  <uses name="gml:CodeType"/>
</def>
<def name="gml:StringOrRefType" doc="gmlBase.xsd">
  <uses name="string" derivation="extension"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:description" doc="gmlBase.xsd">
  <uses name="gml:StringOrRefType"/>
</def>

<def name="gml:NullEnumeration" doc="basicTypes.xsd"/>
<def name="gml:NullType" doc="basicTypes.xsd">
  <uses name="gml:NullEnumeration"/>
  <uses name="anyURI"/>
</def>
<def name="gml:Null" doc="basicTypes.xsd">
  <uses name="gml:NullType"/>
</def>
<def name="gml:booleanOrNull" doc="basicTypes.xsd">
  <uses name="gml:NullEnumeration"/>
```



```
  <uses name="boolean"/>
  <uses name="anyURI"/>
</def>
<def name="gml:booleanOrNullList" doc="basicTypes.xsd">
  <uses name="gml:booleanOrNull"/>
</def>
<def name="gml:booleanList" doc="basicTypes.xsd">
  <uses name="boolean"/>
</def>
<def name="gml:stringOrNull" doc="basicTypes.xsd">
  <uses name="gml:NullEnumeration"/>
  <uses name="string"/>
  <uses name="anyURI"/>
</def>
<def name="gml:NameOrNull" doc="basicTypes.xsd">
  <uses name="gml:NullEnumeration"/>
  <uses name="Name"/>
  <uses name="anyURI"/>
</def>
<def name="gml:NameOrNullList" doc="basicTypes.xsd">
  <uses name="gml:NameOrNull"/>
</def>
<def name="gml:NameList" doc="basicTypes.xsd">
  <uses name="Name"/>
</def>
<def name="gml:doubleOrNull" doc="basicTypes.xsd">
  <uses name="gml:NullEnumeration"/>
  <uses name="double"/>
  <uses name="anyURI"/>
</def>
<def name="gml:doubleOrNullList" doc="basicTypes.xsd">
  <uses name="gml:doubleOrNull"/>
</def>
<def name="gml:doubleList" doc="basicTypes.xsd">
  <uses name="double"/>
</def>
<def name="gml:integerOrNull" doc="basicTypes.xsd">
  <uses name="gml:NullEnumeration"/>
  <uses name="integer"/>
  <uses name="anyURI"/>
</def>
<def name="gml:integerOrNullList" doc="basicTypes.xsd">
  <uses name="gml:integerOrNull"/>
</def>
<def name="gml:integerList" doc="basicTypes.xsd">
  <uses name="integer"/>
</def>
<def name="gml:CodeType" doc="basicTypes.xsd">
  <uses name="string" derivation="extension"/>
</def>
<def name="gml:CodeListType" doc="basicTypes.xsd">
  <uses name="gml:NameList" derivation="extension"/>
</def>
<def name="gml:CodeOrNullListType" doc="basicTypes.xsd">
  <uses name="gml:NameOrNullList" derivation="extension"/>
</def>
<def name="gml:MeasureType" doc="basicTypes.xsd">
  <uses name="double" derivation="extension"/>
</def>
<def name="gml:MeasureListType" doc="basicTypes.xsd">
```

```
  <uses name="gml:doubleList" derivation="extension"/>
</def>
<def name="gml:MeasureOrNullListType" doc="basicTypes.xsd">
  <uses name="gml:doubleOrNullList" derivation="extension"/>
</def>
<def name="gml:CoordinatesType" doc="basicTypes.xsd">
  <uses name="string" derivation="extension"/>
</def>
<def name="gml:SignType" doc="basicTypes.xsd"/>

<def name="xlink:href" doc="../xlink/xlinks.xsd"/>
<def name="xlink:role" doc="../xlink/xlinks.xsd"/>
<def name="xlink:arcrole" doc="../xlink/xlinks.xsd"/>
<def name="xlink:title" doc="../xlink/xlinks.xsd"/>
<def name="xlink:show" doc="../xlink/xlinks.xsd"/>
<def name="xlink:actuate" doc="../xlink/xlinks.xsd"/>
<def name="xlink:label" doc="../xlink/xlinks.xsd"/>
<def name="xlink:from" doc="../xlink/xlinks.xsd"/>
<def name="xlink:to" doc="../xlink/xlinks.xsd"/>
<def name="xlink:simpleLink" doc="../xlink/xlinks.xsd">
  <uses name="xlink:href"/>
  <uses name="xlink:role"/>
  <uses name="xlink:arcrole"/>
  <uses name="xlink:title"/>
  <uses name="xlink:show"/>
  <uses name="xlink:actuate"/>
</def>
<def name="xlink:extendedLink" doc="../xlink/xlinks.xsd">
  <uses name="xlink:role"/>
  <uses name="xlink:title"/>
</def>
<def name="xlink:locatorLink" doc="../xlink/xlinks.xsd">
  <uses name="xlink:href"/>
  <uses name="xlink:role"/>
  <uses name="xlink:title"/>
  <uses name="xlink:label"/>
</def>
<def name="xlink:arcLink" doc="../xlink/xlinks.xsd">
  <uses name="xlink:arcrole"/>
  <uses name="xlink:title"/>
  <uses name="xlink:show"/>
  <uses name="xlink:actuate"/>
  <uses name="xlink:from"/>
  <uses name="xlink:to"/>
</def>
<def name="xlink:resourceLink" doc="../xlink/xlinks.xsd">
  <uses name="xlink:role"/>
  <uses name="xlink:title"/>
  <uses name="xlink:label"/>
</def>
<def name="xlink:titleLink" doc="../xlink/xlinks.xsd"/>
<def name="xlink:emptyLink" doc="../xlink/xlinks.xsd"/>

<def name="gml:AbstractFeatureType" doc="feature.xsd">
  <uses name="gml:AbstractGMLType" derivation="extension"/>
  <uses name="gml:boundedBy"/>
  <uses name="gml:location"/>
</def>
<def name="gml:_Feature" doc="feature.xsd">
  <uses name="gml:AbstractFeatureType"/>
```



```
  <uses name="gml:_GML"/>
</def>
<def name="gml:FeaturePropertyType" doc="feature.xsd">
  <uses name="gml:_Feature"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:featureMember" doc="feature.xsd">
  <uses name="gml:FeaturePropertyType"/>
</def>
<def name="gml:featureProperty" doc="feature.xsd">
  <uses name="gml:FeaturePropertyType"/>
</def>
<def name="gml:FeatureArrayPropertyType" doc="feature.xsd">
  <uses name="gml:_Feature"/>
</def>
<def name="gml:featureMembers" doc="feature.xsd">
  <uses name="gml:FeatureArrayPropertyType"/>
</def>
<def name="gml:BoundedFeatureType" doc="feature.xsd">
  <uses name="gml:AbstractFeatureType" derivation="restriction"/>
  <uses name="gml:metaDataProperty"/>
  <uses name="gml:description"/>
  <uses name="gml:name"/>
  <uses name="gml:boundedBy"/>
  <uses name="gml:location"/>
</def>
<def name="gml:AbstractFeatureCollectionType" doc="feature.xsd">
  <uses name="gml:BoundedFeatureType" derivation="extension"/>
  <uses name="gml:featureMember"/>
  <uses name="gml:featureMembers"/>
</def>
<def name="gml:_FeatureCollection" doc="feature.xsd">
  <uses name="gml:AbstractFeatureCollectionType"/>
  <uses name="gml:_Feature"/>
</def>
<def name="gml:FeatureCollectionType" doc="feature.xsd">
  <uses name="gml:AbstractFeatureCollectionType" derivation="extension"/>
</def>
<def name="gml:FeatureCollection" doc="feature.xsd">
  <uses name="gml:FeatureCollectionType"/>
  <uses name="gml:_Feature"/>
</def>
<def name="gml:boundingShape" doc="feature.xsd">
  <uses name="gml:Envelope"/>
  <uses name="gml:Null"/>
</def>
<def name="gml:BoundingShapeType" doc="feature.xsd">
  <uses name="gml:boundingShape"/>
</def>
<def name="gml:boundedBy" doc="feature.xsd">
  <uses name="gml:BoundingShapeType"/>
</def>
<def name="gml:EnvelopeWithTimePeriodType" doc="feature.xsd">
  <uses name="gml:EnvelopeType" derivation="extension"/>
  <uses name="gml:timePosition"/>
</def>
<def name="gml:EnvelopeWithTimePeriod" doc="feature.xsd">
  <uses name="gml:EnvelopeWithTimePeriodType"/>
  <uses name="gml:Envelope"/>
</def>
```

```
<def name="gml:Box" doc="feature.xsd">
  <uses name="gml:EnvelopeType"/>
  <uses name="gml:Envelope"/>
</def>
<def name="gml:LocationKeyWord" doc="feature.xsd">
  <uses name="gml:CodeType"/>
</def>
<def name="gml:LocationString" doc="feature.xsd">
  <uses name="gml:StringOrRefType"/>
</def>
<def name="gml:locator" doc="feature.xsd">
  <uses name="gml:_Geometry"/>
  <uses name="gml:LocationKeyWord"/>
  <uses name="gml:LocationString"/>
</def>
<def name="gml:LocationPropertyType" doc="feature.xsd">
  <uses name="gml:locator"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:location" doc="feature.xsd">
  <uses name="gml:LocationPropertyType"/>
  <uses name="gml:_property"/>
</def>
<def name="gml:PriorityLocationPropertyType" doc="feature.xsd">
  <uses name="gml:LocationPropertyType" derivation="extension"/>
</def>
<def name="gml:priorityLocation" doc="feature.xsd">
  <uses name="gml:PriorityLocationPropertyType"/>
  <uses name="gml:location"/>
</def>
<def name="gml:centerOf" doc="feature.xsd">
  <uses name="gml:PointPropertyType"/>
  <uses name="gml:pointProperty"/>
</def>
<def name="gml:position" doc="feature.xsd">
  <uses name="gml:PointPropertyType"/>
  <uses name="gml:pointProperty"/>
</def>
<def name="gml:edgeOf" doc="feature.xsd">
  <uses name="gml:CurvePropertyType"/>
  <uses name="gml:curveProperty"/>
</def>
<def name="gml:centerLineOf" doc="feature.xsd">
  <uses name="gml:CurvePropertyType"/>
  <uses name="gml:curveProperty"/>
</def>
<def name="gml:extentOf" doc="feature.xsd">
  <uses name="gml:SurfacePropertyType"/>
  <uses name="gml:surfaceProperty"/>
</def>
<def name="gml:coverage" doc="feature.xsd">
  <uses name="gml:SurfacePropertyType"/>
  <uses name="gml:surfaceProperty"/>
</def>

<def name="gml:AbstractSurfaceType" doc="geometryBasic2d.xsd">
  <uses name="gml:AbstractGeometricPrimitiveType" derivation="extension"/>
</def>
<def name="gml:_Surface" doc="geometryBasic2d.xsd">
  <uses name="gml:AbstractSurfaceType"/>
```



```
  <uses name="gml:_GeometricPrimitive"/>
</def>
<def name="gml:SurfacePropertyType" doc="geometryBasic2d.xsd">
  <uses name="gml:_Surface"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:surfaceProperty" doc="geometryBasic2d.xsd">
  <uses name="gml:SurfacePropertyType"/>
</def>
<def name="gml:SurfaceArrayPropertyType" doc="geometryBasic2d.xsd">
  <uses name="gml:_Surface"/>
</def>
<def name="gml:surfaceArrayProperty" doc="geometryBasic2d.xsd">
  <uses name="gml:SurfaceArrayPropertyType"/>
</def>
<def name="gml:PolygonType" doc="geometryBasic2d.xsd">
  <uses name="gml:AbstractSurfaceType" derivation="extension"/>
  <uses name="gml:exterior"/>
  <uses name="gml:interior"/>
</def>
<def name="gml:Polygon" doc="geometryBasic2d.xsd">
  <uses name="gml:PolygonType"/>
  <uses name="gml:_Surface"/>
</def>
<def name="gml:AbstractRingType" doc="geometryBasic2d.xsd">
  <uses name="gml:AbstractGeometryType" derivation="extension"/>
</def>
<def name="gml:_Ring" doc="geometryBasic2d.xsd">
  <uses name="gml:AbstractRingType"/>
  <uses name="gml:_Geometry"/>
</def>
<def name="gml:AbstractRingPropertyType" doc="geometryBasic2d.xsd">
  <uses name="gml:_Ring"/>
</def>
<def name="gml:exterior" doc="geometryBasic2d.xsd">
  <uses name="gml:AbstractRingPropertyType"/>
</def>
<def name="gml:interior" doc="geometryBasic2d.xsd">
  <uses name="gml:AbstractRingPropertyType"/>
</def>
<def name="gml:outerBoundaryIs" doc="geometryBasic2d.xsd">
  <uses name="gml:AbstractRingPropertyType"/>
  <uses name="gml:exterior"/>
</def>
<def name="gml:innerBoundaryIs" doc="geometryBasic2d.xsd">
  <uses name="gml:AbstractRingPropertyType"/>
  <uses name="gml:interior"/>
</def>
<def name="gml:LinearRingType" doc="geometryBasic2d.xsd">
  <uses name="gml:AbstractRingType" derivation="extension"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coordinates"/>
  <uses name="gml:coord"/>
</def>
<def name="gml:LinearRing" doc="geometryBasic2d.xsd">
  <uses name="gml:LinearRingType"/>
  <uses name="gml:_Ring"/>
</def>
<def name="gml:LinearRingPropertyType" doc="geometryBasic2d.xsd">
```

```
  <uses name="gml:LinearRing"/>
</def>
<def name="gml:PolygonPropertyType" doc="geometryBasic2d.xsd">
  <uses name="gml:Polygon"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:polygonProperty" doc="geometryBasic2d.xsd">
  <uses name="gml:PolygonPropertyType"/>
</def>

<def name="gml:AbstractGeometryType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:AbstractGMLType" derivation="extension"/>
</def>
<def name="gml:_Geometry" doc="geometryBasic0d1d.xsd">
  <uses name="gml:AbstractGeometryType"/>
  <uses name="gml:_GML"/>
</def>
<def name="gml:GeometryPropertyType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:_Geometry"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:GeometryArrayPropertyType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:_Geometry"/>
</def>
<def name="gml:AbstractGeometricPrimitiveType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:AbstractGeometryType" derivation="extension"/>
</def>
<def name="gml:_GeometricPrimitive" doc="geometryBasic0d1d.xsd">
  <uses name="gml:AbstractGeometricPrimitiveType"/>
  <uses name="gml:_Geometry"/>
</def>
<def name="gml:GeometricPrimitivePropertyType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:_GeometricPrimitive"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:PointType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:AbstractGeometricPrimitiveType" derivation="extension"/>
  <uses name="gml:pos"/>
  <uses name="gml:coordinates"/>
  <uses name="gml:coord"/>
</def>
<def name="gml:Point" doc="geometryBasic0d1d.xsd">
  <uses name="gml:PointType"/>
  <uses name="gml:_GeometricPrimitive"/>
</def>
<def name="gml:PointPropertyType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:Point"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:pointProperty" doc="geometryBasic0d1d.xsd">
  <uses name="gml:PointPropertyType"/>
</def>
<def name="gml:PointArrayPropertyType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:Point"/>
</def>
<def name="gml:pointArrayProperty" doc="geometryBasic0d1d.xsd">
  <uses name="gml:PointArrayPropertyType"/>
</def>
<def name="gml:AbstractCurveType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:AbstractGeometricPrimitiveType" derivation="extension"/>
```

```
</def>
<def name="gml:_Curve" doc="geometryBasic0d1d.xsd">
  <uses name="gml:AbstractCurveType"/>
  <uses name="gml:_GeometricPrimitive"/>
</def>
<def name="gml:CurvePropertyType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:_Curve"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:curveProperty" doc="geometryBasic0d1d.xsd">
  <uses name="gml:CurvePropertyType"/>
</def>
<def name="gml:CurveArrayPropertyType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:_Curve"/>
</def>
<def name="gml:curveArrayProperty" doc="geometryBasic0d1d.xsd">
  <uses name="gml:CurveArrayPropertyType"/>
</def>
<def name="gml:LineStringType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:AbstractCurveType" derivation="extension"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coord"/>
  <uses name="gml:coordinates"/>
</def>
<def name="gml:LineString" doc="geometryBasic0d1d.xsd">
  <uses name="gml:LineStringType"/>
  <uses name="gml:_Curve"/>
</def>
<def name="gml:DirectPositionType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:doubleList" derivation="extension"/>
</def>
<def name="gml:pos" doc="geometryBasic0d1d.xsd">
  <uses name="gml:DirectPositionType"/>
</def>
<def name="gml:VectorType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:DirectPositionType" derivation="restriction"/>
</def>
<def name="gml:vector" doc="geometryBasic0d1d.xsd">
  <uses name="gml:VectorType"/>
</def>
<def name="gml:pointRep" doc="geometryBasic0d1d.xsd">
  <uses name="gml:PointPropertyType"/>
</def>
<def name="gml:coordinates" doc="geometryBasic0d1d.xsd">
  <uses name="gml:CoordinatesType"/>
</def>
<def name="gml:EnvelopeType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:AbstractGeometryType" derivation="extension"/>
  <uses name="gml:coord"/>
  <uses name="gml:pos"/>
  <uses name="gml:coordinates"/>
</def>
<def name="gml:Envelope" doc="geometryBasic0d1d.xsd">
  <uses name="gml:EnvelopeType"/>
  <uses name="gml:_Geometry"/>
</def>
<def name="gml:coord" doc="geometryBasic0d1d.xsd">
  <uses name="gml:CoordType"/>
</def>
```

```
<def name="gml:CoordType" doc="geometryBasic0d1d.xsd"/>
<def name="gml:LineStringPropertyType" doc="geometryBasic0d1d.xsd">
  <uses name="gml:LineString"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:lineStringProperty" doc="geometryBasic0d1d.xsd">
  <uses name="gml:LineStringPropertyType"/>
</def>

<def name="gml:measure" doc="measures.xsd">
  <uses name="gml:MeasureType"/>
</def>
<def name="gml:LengthType" doc="measures.xsd">
  <uses name="gml:MeasureType" derivation="extension"/>
</def>
<def name="gml:ScaleType" doc="measures.xsd">
  <uses name="gml:MeasureType" derivation="extension"/>
</def>
<def name="gml:TimeType" doc="measures.xsd">
  <uses name="gml:MeasureType" derivation="extension"/>
</def>
<def name="gml:GridLengthType" doc="measures.xsd">
  <uses name="gml:MeasureType" derivation="extension"/>
</def>
<def name="gml:AreaType" doc="measures.xsd">
  <uses name="gml:MeasureType" derivation="extension"/>
</def>
<def name="gml:VolumeType" doc="measures.xsd">
  <uses name="gml:MeasureType" derivation="extension"/>
</def>
<def name="gml:VelocityType" doc="measures.xsd">
  <uses name="gml:MeasureType" derivation="extension"/>
</def>
<def name="gml:AngleChoiceType" doc="measures.xsd">
  <uses name="gml:angle"/>
  <uses name="gml:dmsAngle"/>
</def>
<def name="gml:angle" doc="measures.xsd">
  <uses name="gml:AngleType"/>
</def>
<def name="gml:AngleType" doc="measures.xsd">
  <uses name="gml:MeasureType" derivation="extension"/>
</def>
<def name="gml:dmsAngle" doc="measures.xsd">
  <uses name="gml:DMSAngleType"/>
</def>
<def name="gml:DMSAngleType" doc="measures.xsd">
  <uses name="gml:degrees"/>
  <uses name="gml:decimalMinutes"/>
  <uses name="gml:minutes"/>
  <uses name="gml:seconds"/>
</def>
<def name="gml:degrees" doc="measures.xsd">
  <uses name="gml:DegreesType"/>
</def>
<def name="gml:DegreesType" doc="measures.xsd">
  <uses name="gml:DegreeValueType" derivation="extension"/>
  <uses name="string" derivation="restriction"/>
  <uses name="gml:SignType" derivation="restriction"/>
</def>
```



```
<def name="gml:DegreeValueType" doc="measures.xsd"/>
<def name="gml:decimalMinutes" doc="measures.xsd">
  <uses name="gml:DecimalMinutesType"/>
</def>
<def name="gml:DecimalMinutesType" doc="measures.xsd"/>
<def name="gml:minutes" doc="measures.xsd">
  <uses name="gml:ArcMinutesType"/>
</def>
<def name="gml:ArcMinutesType" doc="measures.xsd"/>
<def name="gml:seconds" doc="measures.xsd">
  <uses name="gml:ArcSecondsType"/>
</def>
<def name="gml:ArcSecondsType" doc="measures.xsd"/>

<def name="gml:DirectionPropertyType" doc="direction.xsd">
  <uses name="gml:DirectionVector"/>
  <uses name="gml:CompassPoint"/>
  <uses name="gml:CodeType"/>
  <uses name="gml:StringOrRefType"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:direction" doc="direction.xsd">
  <uses name="gml:DirectionPropertyType"/>
</def>
<def name="gml:DirectionVectorType" doc="direction.xsd">
  <uses name="gml:vector"/>
  <uses name="gml:AngleType"/>
</def>
<def name="gml:DirectionVector" doc="direction.xsd">
  <uses name="gml:DirectionVectorType"/>
</def>
<def name="gml:CompassPointEnumeration" doc="direction.xsd"/>
<def name="gml:CompassPoint" doc="direction.xsd">
  <uses name="gml:CompassPointEnumeration"/>
</def>

<def name="gml:AbstractTopologyType" doc="topology.xsd">
  <uses name="gml:AbstractGMLType" derivation="extension"/>
</def>
<def name="gml:_Topology" doc="topology.xsd">
  <uses name="gml:AbstractTopologyType"/>
  <uses name="gml:_Object"/>
</def>
<def name="gml:AbstractTopoPrimitiveType" doc="topology.xsd">
  <uses name="gml:AbstractTopologyType" derivation="extension"/>
  <uses name="gml:isolated"/>
  <uses name="gml:container"/>
</def>
<def name="gml:_TopoPrimitive" doc="topology.xsd">
  <uses name="gml:AbstractTopoPrimitiveType"/>
  <uses name="gml:_Topology"/>
</def>
<def name="gml:IsolatedPropertyType" doc="topology.xsd">
  <uses name="gml:Node"/>
  <uses name="gml:Edge"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:isolated" doc="topology.xsd">
  <uses name="gml:IsolatedPropertyType"/>
</def>
```

```
<def name="gml:ContainerPropertyType" doc="topology.xsd">
  <uses name="gml:Face"/>
  <uses name="gml:TopoSolid"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:container" doc="topology.xsd">
  <uses name="gml:ContainerPropertyType"/>
</def>
<def name="gml:NodeType" doc="topology.xsd">
  <uses name="gml:AbstractTopoPrimitiveType" derivation="extension"/>
  <uses name="gml:directedEdge"/>
  <uses name="gml:pointProperty"/>
</def>
<def name="gml:Node" doc="topology.xsd">
  <uses name="gml:NodeType"/>
  <uses name="gml:_TopoPrimitive"/>
</def>
<def name="gml:DirectedNodePropertyType" doc="topology.xsd">
  <uses name="gml:Node"/>
  <uses name="gml:SignType"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:directedNode" doc="topology.xsd">
  <uses name="gml:DirectedNodePropertyType"/>
</def>
<def name="gml:EdgeType" doc="topology.xsd">
  <uses name="gml:AbstractTopoPrimitiveType" derivation="extension"/>
  <uses name="gml:directedNode"/>
  <uses name="gml:directedFace"/>
  <uses name="gml:curveProperty"/>
</def>
<def name="gml:Edge" doc="topology.xsd">
  <uses name="gml:EdgeType"/>
  <uses name="gml:_TopoPrimitive"/>
</def>
<def name="gml:DirectedEdgePropertyType" doc="topology.xsd">
  <uses name="gml:Edge"/>
  <uses name="gml:SignType"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:directedEdge" doc="topology.xsd">
  <uses name="gml:DirectedEdgePropertyType"/>
</def>
<def name="gml:FaceType" doc="topology.xsd">
  <uses name="gml:AbstractTopoPrimitiveType" derivation="extension"/>
  <uses name="gml:directedEdge"/>
  <uses name="gml:directedTopoSolid"/>
  <uses name="gml:surfaceProperty"/>
</def>
<def name="gml:Face" doc="topology.xsd">
  <uses name="gml:FaceType"/>
  <uses name="gml:_TopoPrimitive"/>
</def>
<def name="gml:DirectedFacePropertyType" doc="topology.xsd">
  <uses name="gml:Face"/>
  <uses name="gml:SignType"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:directedFace" doc="topology.xsd">
  <uses name="gml:DirectedFacePropertyType"/>
```

```
</def>
<def name="gml:TopoSolidType" doc="topology.xsd">
  <uses name="gml:AbstractTopoPrimitiveType" derivation="extension"/>
  <uses name="gml:directedFace"/>
</def>
<def name="gml:TopoSolid" doc="topology.xsd">
  <uses name="gml:TopoSolidType"/>
  <uses name="gml:_TopoPrimitive"/>
</def>
<def name="gml:DirectedTopoSolidPropertyType" doc="topology.xsd">
  <uses name="gml:TopoSolid"/>
  <uses name="gml:SignType"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:directedTopoSolid" doc="topology.xsd">
  <uses name="gml:DirectedTopoSolidPropertyType"/>
</def>
<def name="gml:TopoPointType" doc="topology.xsd">
  <uses name="gml:directedNode"/>
</def>
<def name="gml:TopoPoint" doc="topology.xsd">
  <uses name="gml:TopoPointType"/>
</def>
<def name="gml:TopoPointPropertyType" doc="topology.xsd">
  <uses name="gml:TopoPoint"/>
</def>
<def name="gml:topoPointProperty" doc="topology.xsd">
  <uses name="gml:TopoPointPropertyType"/>
</def>
<def name="gml:TopoCurveType" doc="topology.xsd">
  <uses name="gml:directedEdge"/>
</def>
<def name="gml:TopoCurve" doc="topology.xsd">
  <uses name="gml:TopoCurveType"/>
</def>
<def name="gml:TopoCurvePropertyType" doc="topology.xsd">
  <uses name="gml:TopoCurve"/>
</def>
<def name="gml:topoCurveProperty" doc="topology.xsd">
  <uses name="gml:TopoCurvePropertyType"/>
</def>
<def name="gml:TopoSurfaceType" doc="topology.xsd">
  <uses name="gml:directedFace"/>
</def>
<def name="gml:TopoSurface" doc="topology.xsd">
  <uses name="gml:TopoSurfaceType"/>
</def>
<def name="gml:TopoSurfacePropertyType" doc="topology.xsd">
  <uses name="gml:TopoSurface"/>
</def>
<def name="gml:topoSurfaceProperty" doc="topology.xsd">
  <uses name="gml:TopoSurfacePropertyType"/>
</def>
<def name="gml:TopoVolumeType" doc="topology.xsd">
  <uses name="gml:directedTopoSolid"/>
</def>
<def name="gml:TopoVolume" doc="topology.xsd">
  <uses name="gml:TopoVolumeType"/>
</def>
<def name="gml:TopoVolumePropertyType" doc="topology.xsd">
```

```
  <uses name="gml:TopoVolume"/>
</def>
<def name="gml:topoVolumeProperty" doc="topology.xsd">
  <uses name="gml:TopoVolumePropertyType"/>
</def>
<def name="gml:TopoComplexType" doc="topology.xsd">
  <uses name="gml:AbstractTopologyType" derivation="extension"/>
  <uses name="gml:maximalComplex"/>
  <uses name="gml:superComplex"/>
  <uses name="gml:subComplex"/>
  <uses name="gml:topoPrimitiveMember"/>
  <uses name="gml:topoPrimitiveMembers"/>
</def>
<def name="gml:TopoComplex" doc="topology.xsd">
  <uses name="gml:TopoComplexType"/>
  <uses name="gml:_Topology"/>
</def>
<def name="gml:TopoComplexMemberType" doc="topology.xsd">
  <uses name="gml:TopoComplex"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:topoComplexProperty" doc="topology.xsd">
  <uses name="gml:TopoComplexMemberType"/>
  <uses name="gml:_Topology"/>
</def>
<def name="gml:subComplex" doc="topology.xsd">
  <uses name="gml:TopoComplexMemberType"/>
</def>
<def name="gml:superComplex" doc="topology.xsd">
  <uses name="gml:TopoComplexMemberType"/>
</def>
<def name="gml:maximalComplex" doc="topology.xsd">
  <uses name="gml:TopoComplexMemberType"/>
</def>
<def name="gml:topoPrimitiveMemberType" doc="topology.xsd">
  <uses name="gml:_TopoPrimitive"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:topoPrimitiveMember" doc="topology.xsd">
  <uses name="gml:topoPrimitiveMemberType"/>
</def>
<def name="gml:TopoPrimitiveArrayAssociationType" doc="topology.xsd">
  <uses name="gml:ArrayAssociationType" derivation="restriction"/>
  <uses name="gml:_TopoPrimitive"/>
</def>
<def name="gml:topoPrimitiveMembers" doc="topology.xsd">
  <uses name="gml:TopoPrimitiveArrayAssociationType"/>
  <uses name="gml:members"/>
</def>

<def name="gml:CompositeCurveType" doc="geometryComplexes.xsd">
  <uses name="gml:AbstractCurveType" derivation="extension"/>
  <uses name="gml:curveMember"/>
</def>
<def name="gml:CompositeCurve" doc="geometryComplexes.xsd">
  <uses name="gml:CompositeCurveType"/>
  <uses name="gml:_Curve"/>
</def>
<def name="gml:CompositeSurfaceType" doc="geometryComplexes.xsd">
  <uses name="gml:AbstractSurfaceType" derivation="extension"/>
```

```
  <uses name="gml:surfaceMember"/>
</def>
<def name="gml:CompositeSurface" doc="geometryComplexes.xsd">
  <uses name="gml:CompositeSurfaceType"/>
  <uses name="gml:_Surface"/>
</def>
<def name="gml:CompositeSolidType" doc="geometryComplexes.xsd">
  <uses name="gml:AbstractSolidType" derivation="extension"/>
  <uses name="gml:solidMember"/>
</def>
<def name="gml:CompositeSolid" doc="geometryComplexes.xsd">
  <uses name="gml:CompositeSolidType"/>
  <uses name="gml:_Solid"/>
</def>
<def name="gml:GeometricComplexType" doc="geometryComplexes.xsd">
  <uses name="gml:AbstractGeometryType" derivation="extension"/>
  <uses name="gml:GeometricPrimitivePropertyType"/>
</def>
<def name="gml:GeometricComplex" doc="geometryComplexes.xsd">
  <uses name="gml:GeometricComplexType"/>
  <uses name="gml:_Geometry"/>
</def>
<def name="gml:GeometricComplexPropertyType" doc="geometryComplexes.xsd">
  <uses name="gml:GeometricComplex"/>
  <uses name="gml:CompositeCurve"/>
  <uses name="gml:CompositeSurface"/>
  <uses name="gml:CompositeSolid"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>

<def name="gml:AbstractGeometricAggregateType" doc="geometryAggregates.xsd">
  <uses name="gml:AbstractGeometryType" derivation="extension"/>
</def>
<def name="gml:_GeometricAggregate" doc="geometryAggregates.xsd">
  <uses name="gml:AbstractGeometricAggregateType"/>
  <uses name="gml:_Geometry"/>
</def>
<def name="gml:MultiGeometryType" doc="geometryAggregates.xsd">
  <uses name="gml:AbstractGeometricAggregateType" derivation="extension"/>
  <uses name="gml:geometryMember"/>
  <uses name="gml:geometryMembers"/>
</def>
<def name="gml:MultiGeometry" doc="geometryAggregates.xsd">
  <uses name="gml:MultiGeometryType"/>
  <uses name="gml:_GeometricAggregate"/>
</def>
<def name="gml:MultiGeometryPropertyType" doc="geometryAggregates.xsd">
  <uses name="gml:_GeometricAggregate"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:multiGeometryProperty" doc="geometryAggregates.xsd">
  <uses name="gml:MultiGeometryPropertyType"/>
</def>
<def name="gml:MultiPointType" doc="geometryAggregates.xsd">
  <uses name="gml:AbstractGeometricAggregateType" derivation="extension"/>
  <uses name="gml:pointMember"/>
  <uses name="gml:pointMembers"/>
</def>
<def name="gml:MultiPoint" doc="geometryAggregates.xsd">
  <uses name="gml:MultiPointType"/>
```

```
  <uses name="gml:_GeometricAggregate"/>
</def>
<def name="gml:MultiPointPropertyType" doc="geometryAggregates.xsd">
  <uses name="gml:MultiPoint"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:multiPointProperty" doc="geometryAggregates.xsd">
  <uses name="gml:MultiPointPropertyType"/>
</def>
<def name="gml:MultiCurveType" doc="geometryAggregates.xsd">
  <uses name="gml:AbstractGeometricAggregateType" derivation="extension"/>
  <uses name="gml:curveMember"/>
  <uses name="gml:curveMembers"/>
</def>
<def name="gml:MultiCurve" doc="geometryAggregates.xsd">
  <uses name="gml:MultiCurveType"/>
  <uses name="gml:_GeometricAggregate"/>
</def>
<def name="gml:MultiCurvePropertyType" doc="geometryAggregates.xsd">
  <uses name="gml:MultiCurve"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:multiCurveProperty" doc="geometryAggregates.xsd">
  <uses name="gml:MultiCurvePropertyType"/>
</def>
<def name="gml:MultiSurfaceType" doc="geometryAggregates.xsd">
  <uses name="gml:AbstractGeometricAggregateType" derivation="extension"/>
  <uses name="gml:surfaceMember"/>
  <uses name="gml:surfaceMembers"/>
</def>
<def name="gml:MultiSurface" doc="geometryAggregates.xsd">
  <uses name="gml:MultiSurfaceType"/>
  <uses name="gml:_GeometricAggregate"/>
</def>
<def name="gml:MultiSurfacePropertyType" doc="geometryAggregates.xsd">
  <uses name="gml:MultiSurface"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:multiSurfaceProperty" doc="geometryAggregates.xsd">
  <uses name="gml:MultiSurfacePropertyType"/>
</def>
<def name="gml:MultiSolidType" doc="geometryAggregates.xsd">
  <uses name="gml:AbstractGeometricAggregateType" derivation="extension"/>
  <uses name="gml:solidMember"/>
  <uses name="gml:solidMembers"/>
</def>
<def name="gml:MultiSolid" doc="geometryAggregates.xsd">
  <uses name="gml:MultiSolidType"/>
  <uses name="gml:_GeometricAggregate"/>
</def>
<def name="gml:MultiSolidPropertyType" doc="geometryAggregates.xsd">
  <uses name="gml:MultiSolid"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:multiSolidProperty" doc="geometryAggregates.xsd">
  <uses name="gml:MultiSolidPropertyType"/>
</def>
<def name="gml:MultiLineString" doc="geometryAggregates.xsd">
  <uses name="gml:MultiLineStringType"/>
  <uses name="gml:_GeometricAggregate"/>
```

```
</def>
<def name="gml:MultiPolygon" doc="geometryAggregates.xsd">
  <uses name="gml:MultiPolygonType"/>
  <uses name="gml:_GeometricAggregate"/>
</def>
<def name="gml:MultiLineStringType" doc="geometryAggregates.xsd">
  <uses name="gml:AbstractGeometricAggregateType" derivation="extension"/>
  <uses name="gml:lineStringMember"/>
</def>
<def name="gml:MultiLineStringPropertyType" doc="geometryAggregates.xsd">
  <uses name="gml:MultiLineString"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:MultiPolygonType" doc="geometryAggregates.xsd">
  <uses name="gml:AbstractGeometricAggregateType" derivation="extension"/>
  <uses name="gml:polygonMember"/>
</def>
<def name="gml:MultiPolygonPropertyType" doc="geometryAggregates.xsd">
  <uses name="gml:MultiPolygon"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:geometryMember" doc="geometryAggregates.xsd">
  <uses name="gml:GeometryPropertyType"/>
</def>
<def name="gml:geometryMembers" doc="geometryAggregates.xsd">
  <uses name="gml:GeometryArrayPropertyType"/>
</def>
<def name="gml:pointMember" doc="geometryAggregates.xsd">
  <uses name="gml:PointPropertyType"/>
</def>
<def name="gml:pointMembers" doc="geometryAggregates.xsd">
  <uses name="gml:PointArrayPropertyType"/>
</def>
<def name="gml:curveMembers" doc="geometryAggregates.xsd">
  <uses name="gml:CurveArrayPropertyType"/>
</def>
<def name="gml:surfaceMember" doc="geometryAggregates.xsd">
  <uses name="gml:SurfacePropertyType"/>
</def>
<def name="gml:surfaceMembers" doc="geometryAggregates.xsd">
  <uses name="gml:SurfaceArrayPropertyType"/>
</def>
<def name="gml:solidMember" doc="geometryAggregates.xsd">
  <uses name="gml:SolidPropertyType"/>
</def>
<def name="gml:solidMembers" doc="geometryAggregates.xsd">
  <uses name="gml:SolidArrayPropertyType"/>
</def>
<def name="gml:multiCenterOf" doc="geometryAggregates.xsd">
  <uses name="gml:MultiPointPropertyType"/>
  <uses name="gml:multiPointProperty"/>
</def>
<def name="gml:multiPosition" doc="geometryAggregates.xsd">
  <uses name="gml:MultiPointPropertyType"/>
  <uses name="gml:multiPointProperty"/>
</def>
<def name="gml:multiCenterLineOf" doc="geometryAggregates.xsd">
  <uses name="gml:MultiCurvePropertyType"/>
  <uses name="gml:multiCurveProperty"/>
</def>
```

```
<def name="gml:multiEdgeOf" doc="geometryAggregates.xsd">
  <uses name="gml:MultiCurvePropertyType"/>
  <uses name="gml:multiCurveProperty"/>
</def>
<def name="gml:multiCoverage" doc="geometryAggregates.xsd">
  <uses name="gml:MultiSurfacePropertyType"/>
  <uses name="gml:multiSurfaceProperty"/>
</def>
<def name="gml:multiExtentOf" doc="geometryAggregates.xsd">
  <uses name="gml:MultiSurfacePropertyType"/>
  <uses name="gml:multiSurfaceProperty"/>
</def>
<def name="gml:multiLocation" doc="geometryAggregates.xsd">
  <uses name="gml:MultiPointPropertyType"/>
  <uses name="gml:multiPointProperty"/>
</def>
<def name="gml:lineStringMember" doc="geometryAggregates.xsd">
  <uses name="gml:LineStringPropertyType"/>
</def>
<def name="gml:polygonMember" doc="geometryAggregates.xsd">
  <uses name="gml:PolygonPropertyType"/>
</def>

<def name="gml:CurveType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractCurveType" derivation="extension"/>
  <uses name="gml:segments"/>
</def>
<def name="gml:Curve" doc="geometryPrimitives.xsd">
  <uses name="gml:CurveType"/>
  <uses name="gml:_Curve"/>
</def>
<def name="gml:baseCurve" doc="geometryPrimitives.xsd">
  <uses name="gml:CurvePropertyType"/>
</def>
<def name="gml:OrientableCurveType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractCurveType" derivation="extension"/>
  <uses name="gml:baseCurve"/>
  <uses name="gml:SignType"/>
</def>
<def name="gml:OrientableCurve" doc="geometryPrimitives.xsd">
  <uses name="gml:OrientableCurveType"/>
  <uses name="gml:_Curve"/>
</def>
<def name="gml:AbstractCurveSegmentType" doc="geometryPrimitives.xsd"/>
<def name="gml:_CurveSegment" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractCurveSegmentType"/>
</def>
<def name="gml:CurveSegmentArrayPropertyType" doc="geometryPrimitives.xsd">
  <uses name="gml:_CurveSegment"/>
</def>
<def name="gml:segments" doc="geometryPrimitives.xsd">
  <uses name="gml:CurveSegmentArrayPropertyType"/>
</def>
<def name="gml:LineStringSegmentType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractCurveSegmentType" derivation="extension"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coordinates"/>
  <uses name="gml:CurveInterpolationType"/>
</def>
```

```
<def name="gml:LineStringSegment" doc="geometryPrimitives.xsd">
  <uses name="gml:LineStringSegmentType"/>
  <uses name="gml:_CurveSegment"/>
</def>
<def name="gml:ArcStringType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractCurveSegmentType" derivation="extension"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coordinates"/>
  <uses name="gml:CurveInterpolationType"/>
</def>
<def name="gml:ArcString" doc="geometryPrimitives.xsd">
  <uses name="gml:ArcStringType"/>
  <uses name="gml:_CurveSegment"/>
</def>
<def name="gml:ArcType" doc="geometryPrimitives.xsd">
  <uses name="gml:ArcStringType" derivation="restriction"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coordinates"/>
</def>
<def name="gml:Arc" doc="geometryPrimitives.xsd">
  <uses name="gml:ArcType"/>
  <uses name="gml:ArcString"/>
</def>
<def name="gml:CircleType" doc="geometryPrimitives.xsd">
  <uses name="gml:ArcType" derivation="extension"/>
</def>
<def name="gml:Circle" doc="geometryPrimitives.xsd">
  <uses name="gml:CircleType"/>
  <uses name="gml:Arc"/>
</def>
<def name="gml:ArcStringByBulgeType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractCurveSegmentType" derivation="extension"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coordinates"/>
  <uses name="gml:VectorType"/>
  <uses name="gml:CurveInterpolationType"/>
</def>
<def name="gml:ArcStringByBulge" doc="geometryPrimitives.xsd">
  <uses name="gml:ArcStringByBulgeType"/>
  <uses name="gml:_CurveSegment"/>
</def>
<def name="gml:ArcByBulgeType" doc="geometryPrimitives.xsd">
  <uses name="gml:ArcStringByBulgeType" derivation="restriction"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coordinates"/>
  <uses name="gml:VectorType"/>
</def>
<def name="gml:ArcByBulge" doc="geometryPrimitives.xsd">
  <uses name="gml:ArcByBulgeType"/>
  <uses name="gml:ArcStringByBulge"/>
</def>
<def name="gml:ArcByCenterPointType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractCurveSegmentType" derivation="extension"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coordinates"/>
```



```
  <uses name="gml:LengthType"/>
  <uses name="gml:AngleType"/>
  <uses name="gml:CurveInterpolationType"/>
</def>
<def name="gml:ArcByCenterPoint" doc="geometryPrimitives.xsd">
  <uses name="gml:ArcByCenterPointType"/>
  <uses name="gml:_CurveSegment"/>
</def>
<def name="gml:CircleByCenterPointType" doc="geometryPrimitives.xsd">
  <uses name="gml:ArcByCenterPointType" derivation="extension"/>
</def>
<def name="gml:CircleByCenterPoint" doc="geometryPrimitives.xsd">
  <uses name="gml:CircleByCenterPointType"/>
  <uses name="gml:ArcByCenterPoint"/>
</def>
<def name="gml:CubicSplineType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractCurveSegmentType" derivation="extension"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coordinates"/>
  <uses name="gml:VectorType"/>
  <uses name="gml:CurveInterpolationType"/>
</def>
<def name="gml:CubicSpline" doc="geometryPrimitives.xsd">
  <uses name="gml:CubicSplineType"/>
  <uses name="gml:_CurveSegment"/>
</def>
<def name="gml:KnotType" doc="geometryPrimitives.xsd"/>
<def name="gml:KnotPropertyType" doc="geometryPrimitives.xsd">
  <uses name="gml:KnotType"/>
</def>
<def name="gml:BSplineType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractCurveSegmentType" derivation="extension"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coordinates"/>
  <uses name="gml:KnotPropertyType"/>
  <uses name="gml:CurveInterpolationType"/>
  <uses name="gml:KnotTypesType"/>
</def>
<def name="gml:BSpline" doc="geometryPrimitives.xsd">
  <uses name="gml:BSplineType"/>
  <uses name="gml:_CurveSegment"/>
</def>
<def name="gml:BezierType" doc="geometryPrimitives.xsd">
  <uses name="gml:BSplineType" derivation="restriction"/>
  <uses name="gml:pos"/>
  <uses name="gml:pointRep"/>
  <uses name="gml:coordinates"/>
  <uses name="gml:KnotPropertyType"/>
  <uses name="gml:CurveInterpolationType"/>
  <uses name="gml:KnotTypesType"/>
</def>
<def name="gml:Bezier" doc="geometryPrimitives.xsd">
  <uses name="gml:BezierType"/>
  <uses name="gml:BSpline"/>
</def>
<def name="gml:SurfaceType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractSurfaceType" derivation="extension"/>
  <uses name="gml:patches"/>
```

```
</def>
<def name="gml:Surface" doc="geometryPrimitives.xsd">
  <uses name="gml:SurfaceType"/>
  <uses name="gml:_Surface"/>
</def>
<def name="gml:baseSurface" doc="geometryPrimitives.xsd">
  <uses name="gml:SurfacePropertyType"/>
</def>
<def name="gml:OrientableSurfaceType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractSurfaceType" derivation="extension"/>
  <uses name="gml:baseSurface"/>
  <uses name="gml:SignType"/>
</def>
<def name="gml:OrientableSurface" doc="geometryPrimitives.xsd">
  <uses name="gml:OrientableSurfaceType"/>
  <uses name="gml:_Surface"/>
</def>
<def name="gml:AbstractSurfacePatchType" doc="geometryPrimitives.xsd"/>
<def name="gml:_SurfacePatch" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractSurfacePatchType"/>
</def>
<def name="gml:SurfacePatchArrayPropertyType" doc="geometryPrimitives.xsd">
  <uses name="gml:_SurfacePatch"/>
</def>
<def name="gml:patches" doc="geometryPrimitives.xsd">
  <uses name="gml:SurfacePatchArrayPropertyType"/>
</def>
<def name="gml:PolygonPatchType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractSurfacePatchType" derivation="extension"/>
  <uses name="gml:exterior"/>
  <uses name="gml:interior"/>
  <uses name="gml:SurfaceInterpolationType"/>
</def>
<def name="gml:PolygonPatch" doc="geometryPrimitives.xsd">
  <uses name="gml:PolygonPatchType"/>
  <uses name="gml:_SurfacePatch"/>
</def>
<def name="gml:TriangleType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractSurfacePatchType" derivation="extension"/>
  <uses name="gml:exterior"/>
  <uses name="gml:SurfaceInterpolationType"/>
</def>
<def name="gml:Triangle" doc="geometryPrimitives.xsd">
  <uses name="gml:TriangleType"/>
  <uses name="gml:_SurfacePatch"/>
</def>
<def name="gml:RectangleType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractSurfacePatchType" derivation="extension"/>
  <uses name="gml:exterior"/>
  <uses name="gml:SurfaceInterpolationType"/>
</def>
<def name="gml:Rectangle" doc="geometryPrimitives.xsd">
  <uses name="gml:RectangleType"/>
  <uses name="gml:_SurfacePatch"/>
</def>
<def name="gml:curveMember" doc="geometryPrimitives.xsd">
  <uses name="gml:CurvePropertyType"/>
</def>
<def name="gml:RingType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractRingType" derivation="extension"/>
```

```
  <uses name="gml:curveMember"/>
</def>
<def name="gml:Ring" doc="geometryPrimitives.xsd">
  <uses name="gml:RingType"/>
  <uses name="gml:_Ring"/>
</def>
<def name="gml:RingPropertyType" doc="geometryPrimitives.xsd">
  <uses name="gml:Ring"/>
</def>
<def name="gml:AbstractSolidType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractGeometricPrimitiveType" derivation="extension"/>
</def>
<def name="gml:_Solid" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractSolidType"/>
  <uses name="gml:_GeometricPrimitive"/>
</def>
<def name="gml:SolidPropertyType" doc="geometryPrimitives.xsd">
  <uses name="gml:_Solid"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:solidProperty" doc="geometryPrimitives.xsd">
  <uses name="gml:SolidPropertyType"/>
</def>
<def name="gml:SolidArrayPropertyType" doc="geometryPrimitives.xsd">
  <uses name="gml:_Solid"/>
</def>
<def name="gml:solidArrayProperty" doc="geometryPrimitives.xsd">
  <uses name="gml:SolidArrayPropertyType"/>
</def>
<def name="gml:SolidType" doc="geometryPrimitives.xsd">
  <uses name="gml:AbstractSolidType" derivation="extension"/>
  <uses name="gml:SurfacePropertyType"/>
</def>
<def name="gml:Solid" doc="geometryPrimitives.xsd">
  <uses name="gml:SolidType"/>
  <uses name="gml:_Solid"/>
</def>
<def name="gml:CurveInterpolationType" doc="geometryPrimitives.xsd"/>
<def name="gml:SurfaceInterpolationType" doc="geometryPrimitives.xsd"/>
<def name="gml:KnotTypesType" doc="geometryPrimitives.xsd"/>

<def name="gml:AbstractCoverageType" doc="coverage.xsd">
  <uses name="gml:AbstractFeatureType" derivation="extension"/>
  <uses name="gml:domainSet"/>
  <uses name="gml:rangeSet"/>
  <uses name="gml:coverageFunction"/>
</def>
<def name="gml:_Coverage" doc="coverage.xsd">
  <uses name="gml:AbstractCoverageType"/>
  <uses name="gml:_Feature"/>
</def>
<def name="gml:DomainSetType" doc="coverage.xsd">
  <uses name="gml:_Geometry"/>
  <uses name="gml:_TimeObject"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:domainSet" doc="coverage.xsd">
  <uses name="gml:DomainSetType"/>
</def>
<def name="gml:RangeSetType" doc="coverage.xsd">
```



```
  <uses name="gml:ValueArray"/>
  <uses name="gml:_ScalarValueList"/>
  <uses name="gml:DataBlock"/>
  <uses name="gml:File"/>
</def>
<def name="gml:rangeSet" doc="coverage.xsd">
  <uses name="gml:RangeSetType"/>
</def>
<def name="gml:CoverageFunctionType" doc="coverage.xsd">
  <uses name="gml:MappingRule"/>
  <uses name="gml:GridFunction"/>
</def>
<def name="gml:coverageFunction" doc="coverage.xsd">
  <uses name="gml:CoverageFunctionType"/>
</def>
<def name="gml:DataBlockType" doc="coverage.xsd">
  <uses name="gml:rangeParameters"/>
  <uses name="gml:tupleList"/>
</def>
<def name="gml:DataBlock" doc="coverage.xsd">
  <uses name="gml:DataBlockType"/>
</def>
<def name="gml:tupleList" doc="coverage.xsd">
  <uses name="gml:CoordinatesType"/>
</def>
<def name="gml:FileType" doc="coverage.xsd">
  <uses name="gml:rangeParameters"/>
  <uses name="gml:FileValueModelType"/>
</def>
<def name="gml:File" doc="coverage.xsd">
  <uses name="gml:FileType"/>
</def>
<def name="gml:FileValueModelType" doc="coverage.xsd"/>
<def name="gml:RangeParametersType" doc="coverage.xsd">
  <uses name="gml:_Value"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:rangeParameters" doc="coverage.xsd">
  <uses name="gml:RangeParametersType"/>
</def>
<def name="gml:MappingRule" doc="coverage.xsd">
  <uses name="gml:StringOrRefType"/>
</def>
<def name="gml:GridFunctionType" doc="coverage.xsd">
  <uses name="gml:SequenceRuleType"/>
  <uses name="gml:integerList"/>
</def>
<def name="gml:GridFunction" doc="coverage.xsd">
  <uses name="gml:GridFunctionType"/>
</def>
<def name="gml:IndexMapType" doc="coverage.xsd">
  <uses name="gml:GridFunctionType" derivation="extension"/>
  <uses name="gml:integerList"/>
</def>
<def name="gml:IndexMap" doc="coverage.xsd">
  <uses name="gml:IndexMapType"/>
  <uses name="gml:GridFunction"/>
</def>
<def name="gml:SequenceRuleNames" doc="coverage.xsd"/>
<def name="gml:IncrementOrder" doc="coverage.xsd"/>
```



```
<def name="gml:SequenceRuleType" doc="coverage.xsd">
  <uses name="gml:SequenceRuleNames" derivation="extension"/>
  <uses name="gml:IncrementOrder"/>
</def>
<def name="gml:MultiPointCoverageType" doc="coverage.xsd">
  <uses name="gml:AbstractCoverageType" derivation="restriction"/>
  <uses name="gml:multiPointDomain"/>
  <uses name="gml:rangeSet"/>
  <uses name="gml:coverageFunction"/>
</def>
<def name="gml:MultiPointCoverage" doc="coverage.xsd">
  <uses name="gml:MultiPointCoverageType"/>
  <uses name="gml:_Coverage"/>
</def>
<def name="gml:MultiPointDomainType" doc="coverage.xsd">
  <uses name="gml:DomainSetType" derivation="restriction"/>
  <uses name="gml:MultiPoint"/>
</def>
<def name="gml:multiPointDomain" doc="coverage.xsd">
  <uses name="gml:MultiPointDomainType"/>
  <uses name="gml:domainSet"/>
</def>
<def name="gml:MultiSurfaceCoverageType" doc="coverage.xsd">
  <uses name="gml:AbstractCoverageType" derivation="restriction"/>
  <uses name="gml:multiSurfaceDomain"/>
  <uses name="gml:rangeSet"/>
  <uses name="gml:coverageFunction"/>
</def>
<def name="gml:MultiSurfaceCoverage" doc="coverage.xsd">
  <uses name="gml:MultiSurfaceCoverageType"/>
  <uses name="gml:_Coverage"/>
</def>
<def name="gml:MultiSurfaceDomainType" doc="coverage.xsd">
  <uses name="gml:DomainSetType" derivation="restriction"/>
  <uses name="gml:MultiSurface"/>
</def>
<def name="gml:multiSurfaceDomain" doc="coverage.xsd">
  <uses name="gml:MultiSurfaceDomainType"/>
  <uses name="gml:domainSet"/>
</def>
<def name="gml:GridCoverageType" doc="coverage.xsd">
  <uses name="gml:AbstractCoverageType" derivation="restriction"/>
  <uses name="gml:gridDomain"/>
  <uses name="gml:rangeSet"/>
  <uses name="gml:coverageFunction"/>
</def>
<def name="gml:GridCoverage" doc="coverage.xsd">
  <uses name="gml:GridCoverageType"/>
  <uses name="gml:_Coverage"/>
</def>
<def name="gml:GridDomainType" doc="coverage.xsd">
  <uses name="gml:DomainSetType" derivation="restriction"/>
  <uses name="gml:Grid"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:gridDomain" doc="coverage.xsd">
  <uses name="gml:GridDomainType"/>
  <uses name="gml:domainSet"/>
</def>
<def name="gml:RectifiedGridCoverageType" doc="coverage.xsd">
```



```
  <uses name="gml:AbstractCoverageType" derivation="restriction"/>
  <uses name="gml:rectifiedGridDomain"/>
  <uses name="gml:rangeSet"/>
  <uses name="gml:coverageFunction"/>
</def>
<def name="gml:RectifiedGridCoverage" doc="coverage.xsd">
  <uses name="gml:RectifiedGridCoverageType"/>
  <uses name="gml:_Coverage"/>
</def>
<def name="gml:RectifiedGridDomainType" doc="coverage.xsd">
  <uses name="gml:DomainSetType" derivation="restriction"/>
  <uses name="gml:RectifiedGrid"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:rectifiedGridDomain" doc="coverage.xsd">
  <uses name="gml:RectifiedGridDomainType"/>
  <uses name="gml:domainSet"/>
</def>

<def name="gml:Value" doc="valueObjects.xsd">
  <uses name="gml:_Value"/>
  <uses name="gml:_Geometry"/>
  <uses name="gml:_TimeObject"/>
  <uses name="gml:Null"/>
  <uses name="gml:measure"/>
</def>
<def name="gml:_Value" doc="valueObjects.xsd">
  <uses name="gml:_Object"/>
</def>
<def name="gml:_ScalarValue" doc="valueObjects.xsd">
  <uses name="gml:_Value"/>
</def>
<def name="gml:_ScalarValueList" doc="valueObjects.xsd">
  <uses name="gml:_Value"/>
</def>
<def name="gml:Boolean" doc="valueObjects.xsd">
  <uses name="gml:_ScalarValue"/>
</def>
<def name="gml:BooleanList" doc="valueObjects.xsd">
  <uses name="gml:booleanOrNullList"/>
  <uses name="gml:_ScalarValueList"/>
</def>
<def name="gml:Category" doc="valueObjects.xsd">
  <uses name="gml:CodeType"/>
  <uses name="gml:_ScalarValue"/>
</def>
<def name="gml:CategoryList" doc="valueObjects.xsd">
  <uses name="gml:CodeOrNullListType"/>
  <uses name="gml:_ScalarValueList"/>
</def>
<def name="gml:Quantity" doc="valueObjects.xsd">
  <uses name="gml:MeasureType"/>
  <uses name="gml:_ScalarValue"/>
</def>
<def name="gml:QuantityList" doc="valueObjects.xsd">
  <uses name="gml:MeasureOrNullListType"/>
  <uses name="gml:_ScalarValueList"/>
</def>
<def name="gml:Count" doc="valueObjects.xsd">
  <uses name="gml:_ScalarValue"/>
```

```
</def>
<def name="gml:CountList" doc="valueObjects.xsd">
  <uses name="gml:integerOrNullList"/>
  <uses name="gml:_ScalarValueList"/>
</def>
<def name="gml:CompositeValueType" doc="valueObjects.xsd">
  <uses name="gml:AbstractGMLType" derivation="extension"/>
  <uses name="gml:valueComponent"/>
  <uses name="gml:valueComponents"/>
</def>
<def name="gml:CompositeValue" doc="valueObjects.xsd">
  <uses name="gml:CompositeValueType"/>
  <uses name="gml:_Value"/>
</def>
<def name="gml:ValueArrayType" doc="valueObjects.xsd">
  <uses name="gml:CompositeValueType" derivation="extension"/>
  <uses name="gml:referenceSystem"/>
</def>
<def name="gml:ValueArray" doc="valueObjects.xsd">
  <uses name="gml:ValueArrayType"/>
  <uses name="gml:CompositeValue"/>
</def>
<def name="gml:referenceSystem" doc="valueObjects.xsd"/>
<def name="gml:QuantityExtentType" doc="valueObjects.xsd">
  <uses name="gml:MeasureOrNullListType" derivation="restriction"/>
</def>
<def name="gml:QuantityExtent" doc="valueObjects.xsd">
  <uses name="gml:QuantityExtentType"/>
  <uses name="gml:_Value"/>
</def>
<def name="gml:CategoryExtentType" doc="valueObjects.xsd">
  <uses name="gml:CodeOrNullListType" derivation="restriction"/>
</def>
<def name="gml:CategoryExtent" doc="valueObjects.xsd">
  <uses name="gml:CategoryExtentType"/>
  <uses name="gml:_Value"/>
</def>
<def name="gml:CountExtentType" doc="valueObjects.xsd"/>
<def name="gml:CountExtent" doc="valueObjects.xsd">
  <uses name="gml:CountExtentType"/>
  <uses name="gml:_Value"/>
</def>
<def name="gml:ValuePropertyType" doc="valueObjects.xsd">
  <uses name="gml:Value"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:valueProperty" doc="valueObjects.xsd">
  <uses name="gml:ValuePropertyType"/>
</def>
<def name="gml:valueComponent" doc="valueObjects.xsd">
  <uses name="gml:ValuePropertyType"/>
</def>
<def name="gml:ValueArrayPropertyType" doc="valueObjects.xsd">
  <uses name="gml:Value"/>
</def>
<def name="gml:valueComponents" doc="valueObjects.xsd">
  <uses name="gml:ValueArrayPropertyType"/>
</def>
<def name="gml:BooleanPropertyType" doc="valueObjects.xsd">
  <uses name="gml:ValuePropertyType" derivation="restriction"/>
```



```
  <uses name="gml:Boolean"/>
</def>
<def name="gml:CategoryPropertyType" doc="valueObjects.xsd">
  <uses name="gml:ValuePropertyType" derivation="restriction"/>
  <uses name="gml:Category"/>
</def>
<def name="gml:QuantityPropertyType" doc="valueObjects.xsd">
  <uses name="gml:ValuePropertyType" derivation="restriction"/>
  <uses name="gml:Quantity"/>
</def>
<def name="gml:CountPropertyType" doc="valueObjects.xsd">
  <uses name="gml:ValuePropertyType" derivation="restriction"/>
  <uses name="gml:Count"/>
</def>

<def name="gml:_ImplicitGeometry" doc="grids.xsd">
  <uses name="gml:AbstractGeometryType"/>
  <uses name="gml:_Geometry"/>
</def>
<def name="gml:GridType" doc="grids.xsd">
  <uses name="gml:AbstractGeometryType" derivation="extension"/>
  <uses name="gml:GridLimitsType"/>
</def>
<def name="gml:Grid" doc="grids.xsd">
  <uses name="gml:GridType"/>
  <uses name="gml:_ImplicitGeometry"/>
</def>
<def name="gml:GridLimitsType" doc="grids.xsd">
  <uses name="gml:GridEnvelopeType"/>
</def>
<def name="gml:GridEnvelopeType" doc="grids.xsd">
  <uses name="gml:integerList"/>
</def>
<def name="gml:RectifiedGridType" doc="grids.xsd">
  <uses name="gml:GridType" derivation="extension"/>
  <uses name="gml:PointPropertyType"/>
  <uses name="gml:VectorType"/>
</def>
<def name="gml:RectifiedGrid" doc="grids.xsd">
  <uses name="gml:RectifiedGridType"/>
  <uses name="gml:Grid"/>
</def>

<def name="gml:_CoordinateReferenceSystem" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractCoordinateReferenceSystemType"/>
  <uses name="gml:_CRS"/>
</def>
<def name="gml:AbstractCoordinateReferenceSystemType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractCRSType" derivation="extension"/>
</def>
<def name="gml:coordinateReferenceSystemRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CoordinateReferenceSystemRefType"/>
  <uses name="gml:crsRef"/>
</def>
<def name="gml:CoordinateReferenceSystemRefType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CRSRefType" derivation="restriction"/>
  <uses name="gml:_CoordinateReferenceSystem"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:CompoundCRS" doc="coordinateReferenceSystems.xsd">
```

```
  <uses name="gml:CompoundCRSType"/>
  <uses name="gml:_CRS"/>
</def>
<def name="gml:CompoundCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractCRSType" derivation="extension"/>
  <uses name="gml:CoordinateReferenceSystemRefType"/>
</def>
<def name="gml:compoundCRSRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CompoundCRSRefType"/>
  <uses name="gml:crsRef"/>
</def>
<def name="gml:CompoundCRSRefType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CRSRefType" derivation="restriction"/>
  <uses name="gml:CompoundCRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:GeographicCRS" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:GeographicCRSType"/>
  <uses name="gml:_CoordinateReferenceSystem"/>
</def>
<def name="gml:GeographicCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractCoordinateReferenceSystemType" derivation="extension"/>
  <uses name="gml:EllipsoidalCSRefType"/>
  <uses name="gml:GeodeticDatumRefType"/>
</def>
<def name="gml:geographicCRSRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:GeographicCRSRefType"/>
  <uses name="gml:coordinateReferenceSystemRef"/>
</def>
<def name="gml:GeographicCRSRefType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CoordinateReferenceSystemRefType" derivation="restriction"/>
  <uses name="gml:GeographicCRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:VerticalCRS" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:VerticalCRSType"/>
  <uses name="gml:_CoordinateReferenceSystem"/>
</def>
<def name="gml:VerticalCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractCoordinateReferenceSystemType" derivation="extension"/>
  <uses name="gml:GravityRelatedCSRefType"/>
  <uses name="gml:VerticalDatumRefType"/>
</def>
<def name="gml:verticalCRSRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:VerticalCRSRefType"/>
  <uses name="gml:coordinateReferenceSystemRef"/>
</def>
<def name="gml:VerticalCRSRefType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CoordinateReferenceSystemRefType" derivation="restriction"/>
  <uses name="gml:VerticalCRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:GeocentricCRS" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:GeocentricCRSType"/>
  <uses name="gml:_CoordinateReferenceSystem"/>
</def>
<def name="gml:GeocentricCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractCoordinateReferenceSystemType" derivation="extension"/>
  <uses name="gml:CartesianCSRefType"/>
  <uses name="gml:SphericalCSRefType"/>
```



```
  <uses name="gml:GeodeticDatumRefType"/>
</def>
<def name="gml:geocentricCRSRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:GeocentricCRSRefType"/>
  <uses name="gml:coordinateReferenceSystemRef"/>
</def>
<def name="gml:GeocentricCRSRefType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CoordinateReferenceSystemRefType" derivation="restriction"/>
  <uses name="gml:GeocentricCRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_GeneralDerivedCRS" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractGeneralDerivedCRSType"/>
  <uses name="gml:_CoordinateReferenceSystem"/>
</def>
<def name="gml:AbstractGeneralDerivedCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractCoordinateReferenceSystemType" derivation="extension"/>
  <uses name="gml:CoordinateReferenceSystemRefType"/>
  <uses name="gml:GeneralConversionRefType"/>
</def>
<def name="gml:generalDerivedCRSRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:GeneralDerivedCRSRefType"/>
  <uses name="gml:coordinateReferenceSystemRef"/>
</def>
<def name="gml:GeneralDerivedCRSRefType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CoordinateReferenceSystemRefType" derivation="restriction"/>
  <uses name="gml:_GeneralDerivedCRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:ProjectedCRS" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:ProjectedCRSType"/>
  <uses name="gml:_GeneralDerivedCRS"/>
</def>
<def name="gml:ProjectedCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractGeneralDerivedCRSType" derivation="extension"/>
  <uses name="gml:CartesianCSRefType"/>
</def>
<def name="gml:projectedCRSRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:ProjectedCRSRefType"/>
  <uses name="gml:generalDerivedCRSRef"/>
</def>
<def name="gml:ProjectedCRSRefType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:GeneralDerivedCRSRefType" derivation="restriction"/>
  <uses name="gml:ProjectedCRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:DerivedCRS" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:DerivedCRSType"/>
  <uses name="gml:_GeneralDerivedCRS"/>
</def>
<def name="gml:DerivedCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractGeneralDerivedCRSType" derivation="extension"/>
  <uses name="gml:CoordinateSystemRefType"/>
  <uses name="gml:derivedCRSType"/>
</def>
<def name="gml:derivedCRSRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:DerivedCRSRefType"/>
  <uses name="gml:generalDerivedCRSRef"/>
</def>
<def name="gml:DerivedCRSRefType" doc="coordinateReferenceSystems.xsd">
```

```
  <uses name="gml:GeneralDerivedCRSRefType" derivation="restriction"/>
  <uses name="gml:DerivedCRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:derivedCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:DerivedCRSTypeType"/>
</def>
<def name="gml:DerivedCRSTypeType" doc="coordinateReferenceSystems.xsd"/>
<def name="gml:EngineeringCRS" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:EngineeringCRSType"/>
  <uses name="gml:_CoordinateReferenceSystem"/>
</def>
<def name="gml:EngineeringCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractCoordinateReferenceSystemType" derivation="extension"/>
  <uses name="gml:CoordinateSystemRefType"/>
  <uses name="gml:EngineeringDatumRefType"/>
</def>
<def name="gml:engineeringCRSRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:EngineeringCRSRefType"/>
  <uses name="gml:coordinateReferenceSystemRef"/>
</def>
<def name="gml:EngineeringCRSRefType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CoordinateReferenceSystemRefType" derivation="restriction"/>
  <uses name="gml:EngineeringCRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:ImageCRS" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:ImageCRSType"/>
  <uses name="gml:_CoordinateReferenceSystem"/>
</def>
<def name="gml:ImageCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractCoordinateReferenceSystemType" derivation="extension"/>
  <uses name="gml:CartesianCSRefType"/>
  <uses name="gml:ObliqueCartesianCSRefType"/>
  <uses name="gml:ImageDatumRefType"/>
</def>
<def name="gml:imageCRSRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:ImageCRSRefType"/>
  <uses name="gml:coordinateReferenceSystemRef"/>
</def>
<def name="gml:ImageCRSRefType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CoordinateReferenceSystemRefType" derivation="restriction"/>
  <uses name="gml:ImageCRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:TemporalCRS" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:TemporalCRSType"/>
  <uses name="gml:_CoordinateReferenceSystem"/>
</def>
<def name="gml:TemporalCRSType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:AbstractCoordinateReferenceSystemType" derivation="extension"/>
  <uses name="gml:TemporalCSRefType"/>
  <uses name="gml:TemporalDatumRefType"/>
</def>
<def name="gml:temporalCRSRef" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:TemporalCRSRefType"/>
  <uses name="gml:coordinateReferenceSystemRef"/>
</def>
<def name="gml:TemporalCRSRefType" doc="coordinateReferenceSystems.xsd">
  <uses name="gml:CoordinateReferenceSystemRefType" derivation="restriction"/>
```

```
  <uses name="gml:TemporalCRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>

<def name="gml:CoordinateSystemAxis" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemAxisType"/>
  <uses name="gml:_CRSObject"/>
</def>
<def name="gml:CoordinateSystemAxisType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCRSObjectType" derivation="extension"/>
  <uses name="gml:IdentifierType"/>
</def>
<def name="gml:coordinateSystemAxisRef" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemAxisRefType"/>
</def>
<def name="gml:CoordinateSystemAxisRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemAxis"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_CoordinateSystem" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCoordinateSystemType"/>
  <uses name="gml:_CRSObject"/>
</def>
<def name="gml:AbstractCoordinateSystemType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCRSObjectType" derivation="extension"/>
  <uses name="gml:IdentifierType"/>
  <uses name="gml:CoordinateSystemAxisRefType"/>
</def>
<def name="gml:coordinateSystemRef" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType"/>
</def>
<def name="gml:CoordinateSystemRefType" doc="coordinateSystems.xsd">
  <uses name="gml:_CoordinateSystem"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:EllipsoidalCS" doc="coordinateSystems.xsd">
  <uses name="gml:EllipsoidalCSType"/>
  <uses name="gml:_CoordinateSystem"/>
</def>
<def name="gml:EllipsoidalCSType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCoordinateSystemType" derivation="extension"/>
</def>
<def name="gml:ellipsoidalCSRef" doc="coordinateSystems.xsd">
  <uses name="gml:EllipsoidalCSRefType"/>
  <uses name="gml:coordinateSystemRef"/>
</def>
<def name="gml:EllipsoidalCSRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType" derivation="restriction"/>
  <uses name="gml:EllipsoidalCS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:CartesianCS" doc="coordinateSystems.xsd">
  <uses name="gml:CartesianCSType"/>
  <uses name="gml:_CoordinateSystem"/>
</def>
<def name="gml:CartesianCSType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCoordinateSystemType" derivation="extension"/>
</def>
<def name="gml:cartesianCSRef" doc="coordinateSystems.xsd">
  <uses name="gml:CartesianCSRefType"/>
```



```
  <uses name="gml:coordinateSystemRef"/>
</def>
<def name="gml:CartesianCSRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType" derivation="restriction"/>
  <uses name="gml:CartesianCS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:GravityRelatedCS" doc="coordinateSystems.xsd">
  <uses name="gml:GravityRelatedCSType"/>
  <uses name="gml:_CoordinateSystem"/>
</def>
<def name="gml:GravityRelatedCSType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCoordinateSystemType" derivation="extension"/>
</def>
<def name="gml:gravityRelatedCSRef" doc="coordinateSystems.xsd">
  <uses name="gml:GravityRelatedCSRefType"/>
  <uses name="gml:coordinateSystemRef"/>
</def>
<def name="gml:GravityRelatedCSRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType" derivation="restriction"/>
  <uses name="gml:GravityRelatedCS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:TemporalCS" doc="coordinateSystems.xsd">
  <uses name="gml:TemporalCSType"/>
  <uses name="gml:_CoordinateSystem"/>
</def>
<def name="gml:TemporalCSType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCoordinateSystemType" derivation="extension"/>
</def>
<def name="gml:temporalCSRef" doc="coordinateSystems.xsd">
  <uses name="gml:TemporalCSRefType"/>
  <uses name="gml:coordinateSystemRef"/>
</def>
<def name="gml:TemporalCSRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType" derivation="restriction"/>
  <uses name="gml:TemporalCS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:LinearCS" doc="coordinateSystems.xsd">
  <uses name="gml:LinearCSType"/>
  <uses name="gml:_CoordinateSystem"/>
</def>
<def name="gml:LinearCSType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCoordinateSystemType" derivation="extension"/>
</def>
<def name="gml:linearCSRef" doc="coordinateSystems.xsd">
  <uses name="gml:LinearCSRefType"/>
  <uses name="gml:coordinateSystemRef"/>
</def>
<def name="gml:LinearCSRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType" derivation="restriction"/>
  <uses name="gml:LinearCS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:UserDefinedCS" doc="coordinateSystems.xsd">
  <uses name="gml:UserDefinedCSType"/>
  <uses name="gml:_CoordinateSystem"/>
</def>
<def name="gml:UserDefinedCSType" doc="coordinateSystems.xsd">
```

```
  <uses name="gml:AbstractCoordinateSystemType" derivation="extension"/>
</def>
<def name="gml:userDefinedCSRef" doc="coordinateSystems.xsd">
  <uses name="gml:UserDefinedCSRefType"/>
  <uses name="gml:coordinateSystemRef"/>
</def>
<def name="gml:UserDefinedCSRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType" derivation="restriction"/>
  <uses name="gml:UserDefinedCS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:SphericalCS" doc="coordinateSystems.xsd">
  <uses name="gml:SphericalCSType"/>
  <uses name="gml:_CoordinateSystem"/>
</def>
<def name="gml:SphericalCSType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCoordinateSystemType" derivation="extension"/>
</def>
<def name="gml:sphericalCSRef" doc="coordinateSystems.xsd">
  <uses name="gml:SphericalCSRefType"/>
  <uses name="gml:coordinateSystemRef"/>
</def>
<def name="gml:SphericalCSRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType" derivation="restriction"/>
  <uses name="gml:SphericalCS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:PolarCS" doc="coordinateSystems.xsd">
  <uses name="gml:PolarCSType"/>
  <uses name="gml:_CoordinateSystem"/>
</def>
<def name="gml:PolarCSType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCoordinateSystemType" derivation="extension"/>
</def>
<def name="gml:polarCSRef" doc="coordinateSystems.xsd">
  <uses name="gml:PolarCSRefType"/>
  <uses name="gml:coordinateSystemRef"/>
</def>
<def name="gml:PolarCSRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType" derivation="restriction"/>
  <uses name="gml:PolarCS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:CylindricalCS" doc="coordinateSystems.xsd">
  <uses name="gml:CylindricalCSType"/>
  <uses name="gml:_CoordinateSystem"/>
</def>
<def name="gml:CylindricalCSType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCoordinateSystemType" derivation="extension"/>
</def>
<def name="gml:cylindricalCSRef" doc="coordinateSystems.xsd">
  <uses name="gml:CylindricalCSRefType"/>
  <uses name="gml:coordinateSystemRef"/>
</def>
<def name="gml:CylindricalCSRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType" derivation="restriction"/>
  <uses name="gml:CylindricalCS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:ObliqueCartesianCS" doc="coordinateSystems.xsd">
```

```
  <uses name="gml:ObliqueCartesianCSType"/>
  <uses name="gml:_CoordinateSystem"/>
</def>
<def name="gml:ObliqueCartesianCSType" doc="coordinateSystems.xsd">
  <uses name="gml:AbstractCoordinateSystemType" derivation="extension"/>
</def>
<def name="gml:obliqueCartesianCSRef" doc="coordinateSystems.xsd">
  <uses name="gml:ObliqueCartesianCSRefType"/>
  <uses name="gml:coordinateSystemRef"/>
</def>
<def name="gml:ObliqueCartesianCSRefType" doc="coordinateSystems.xsd">
  <uses name="gml:CoordinateSystemRefType" derivation="restriction"/>
  <uses name="gml:ObliqueCartesianCS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>

<def name="gml:_CRSObject" doc="referenceSystems.xsd">
  <uses name="gml:AbstractCRSObjectType"/>
  <uses name="gml:_Object"/>
</def>
<def name="gml:AbstractCRSObjectType" doc="referenceSystems.xsd">
  <uses name="gml:AbstractGMLType" derivation="restriction"/>
  <uses name="gml:metaDataProperty"/>
  <uses name="gml:id"/>
</def>
<def name="gml:_ReferenceSystem" doc="referenceSystems.xsd">
  <uses name="gml:AbstractReferenceSystemType"/>
  <uses name="gml:_CRSObject"/>
</def>
<def name="gml:AbstractReferenceSystemType" doc="referenceSystems.xsd">
  <uses name="gml:AbstractCRSObjectType" derivation="extension"/>
  <uses name="gml:ExtendedIdentifierType"/>
</def>
<def name="gml:referenceSystemRef" doc="referenceSystems.xsd">
  <uses name="gml:ReferenceSystemRefType"/>
</def>
<def name="gml:ReferenceSystemRefType" doc="referenceSystems.xsd">
  <uses name="gml:_ReferenceSystem"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_CRS" doc="referenceSystems.xsd">
  <uses name="gml:AbstractCRSType"/>
  <uses name="gml:_ReferenceSystem"/>
</def>
<def name="gml:AbstractCRSType" doc="referenceSystems.xsd">
  <uses name="gml:AbstractReferenceSystemType" derivation="extension"/>
</def>
<def name="gml:crsRef" doc="referenceSystems.xsd">
  <uses name="gml:CRSRefType"/>
  <uses name="gml:referenceSystemRef"/>
</def>
<def name="gml:CRSRefType" doc="referenceSystems.xsd">
  <uses name="gml:ReferenceSystemRefType" derivation="restriction"/>
  <uses name="gml:_CRS"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:IdentifierType" doc="referenceSystems.xsd">
  <uses name="gml:AliasType"/>
</def>
<def name="gml:AliasType" doc="referenceSystems.xsd"/>
```



```
<def name="gml:ExtendedIdentifierType" doc="referenceSystems.xsd">
  <uses name="gml:IdentifierType" derivation="extension"/>
  <uses name="gml:ExtentType"/>
</def>
<def name="gml:ExtentType" doc="referenceSystems.xsd">
  <uses name="gml:description"/>
  <uses name="gml:PolygonType"/>
  <uses name="gml:EnvelopeType"/>
  <uses name="gml:TimePeriodType"/>
</def>

<def name="gml:_Datum" doc="datums.xsd">
  <uses name="gml:AbstractDatumType"/>
  <uses name="gml:_CRSObject"/>
</def>
<def name="gml:AbstractDatumType" doc="datums.xsd">
  <uses name="gml:AbstractCRSObjectType" derivation="extension"/>
  <uses name="gml:ExtendedIdentifierType"/>
</def>
<def name="gml:datumRef" doc="datums.xsd">
  <uses name="gml:DatumRefType"/>
</def>
<def name="gml:DatumRefType" doc="datums.xsd">
  <uses name="gml:_Datum"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:EngineeringDatum" doc="datums.xsd">
  <uses name="gml:EngineeringDatumType"/>
  <uses name="gml:_Datum"/>
</def>
<def name="gml:EngineeringDatumType" doc="datums.xsd">
  <uses name="gml:AbstractDatumType" derivation="extension"/>
</def>
<def name="gml:engineeringDatumRef" doc="datums.xsd">
  <uses name="gml:EngineeringDatumRefType"/>
  <uses name="gml:datumRef"/>
</def>
<def name="gml:EngineeringDatumRefType" doc="datums.xsd">
  <uses name="gml:DatumRefType" derivation="restriction"/>
  <uses name="gml:EngineeringDatum"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:ImageDatum" doc="datums.xsd">
  <uses name="gml:ImageDatumType"/>
  <uses name="gml:_Datum"/>
</def>
<def name="gml:ImageDatumType" doc="datums.xsd">
  <uses name="gml:AbstractDatumType" derivation="extension"/>
  <uses name="gml:pixelInCell"/>
</def>
<def name="gml:imageDatumRef" doc="datums.xsd">
  <uses name="gml:ImageDatumRefType"/>
  <uses name="gml:datumRef"/>
</def>
<def name="gml:ImageDatumRefType" doc="datums.xsd">
  <uses name="gml:DatumRefType" derivation="restriction"/>
  <uses name="gml:ImageDatum"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:pixelInCell" doc="datums.xsd">
```

```
  <uses name="gml:PixelInCellType"/>
</def>
<def name="gml:PixelInCellType" doc="datums.xsd"/>
<def name="gml:VerticalDatum" doc="datums.xsd">
  <uses name="gml:VerticalDatumType"/>
  <uses name="gml:_Datum"/>
</def>
<def name="gml:VerticalDatumType" doc="datums.xsd">
  <uses name="gml:AbstractDatumType" derivation="extension"/>
  <uses name="gml:verticalDatumType"/>
</def>
<def name="gml:verticalDatumRef" doc="datums.xsd">
  <uses name="gml:VerticalDatumRefType"/>
  <uses name="gml:datumRef"/>
</def>
<def name="gml:VerticalDatumRefType" doc="datums.xsd">
  <uses name="gml:DatumRefType" derivation="restriction"/>
  <uses name="gml:VerticalDatum"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:verticalDatumType" doc="datums.xsd">
  <uses name="gml:VerticalDatumTypeType"/>
</def>
<def name="gml:VerticalDatumTypeType" doc="datums.xsd"/>
<def name="gml:TemporalDatumRestrictionType" doc="datums.xsd">
  <uses name="gml:AbstractDatumType" derivation="restriction"/>
  <uses name="gml:metaDataProperty"/>
  <uses name="gml:ExtendedIdentifierType"/>
  <uses name="gml:id"/>
</def>
<def name="gml:TemporalDatum" doc="datums.xsd">
  <uses name="gml:TemporalDatumType"/>
  <uses name="gml:_Datum"/>
</def>
<def name="gml:TemporalDatumType" doc="datums.xsd">
  <uses name="gml:TemporalDatumRestrictionType" derivation="extension"/>
</def>
<def name="gml:temporalDatumRef" doc="datums.xsd">
  <uses name="gml:TemporalDatumRefType"/>
  <uses name="gml:datumRef"/>
</def>
<def name="gml:TemporalDatumRefType" doc="datums.xsd">
  <uses name="gml:DatumRefType" derivation="restriction"/>
  <uses name="gml:TemporalDatum"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:GeodeticDatum" doc="datums.xsd">
  <uses name="gml:GeodeticDatumType"/>
  <uses name="gml:_Datum"/>
</def>
<def name="gml:GeodeticDatumType" doc="datums.xsd">
  <uses name="gml:AbstractDatumType" derivation="extension"/>
  <uses name="gml:PrimeMeridianRefType"/>
  <uses name="gml:EllipsoidRefType"/>
</def>
<def name="gml:geodeticDatumRef" doc="datums.xsd">
  <uses name="gml:GeodeticDatumRefType"/>
  <uses name="gml:datumRef"/>
</def>
<def name="gml:GeodeticDatumRefType" doc="datums.xsd">
```

```
  <uses name="gml:DatumRefType" derivation="restriction"/>
  <uses name="gml:GeodeticDatum"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:PrimeMeridian" doc="datums.xsd">
  <uses name="gml:PrimeMeridianType"/>
  <uses name="gml:_CRSObject"/>
</def>
<def name="gml:PrimeMeridianType" doc="datums.xsd">
  <uses name="gml:AbstractCRSObjectType" derivation="extension"/>
  <uses name="gml:IdentifierType"/>
  <uses name="gml:AngleChoiceType"/>
</def>
<def name="gml:primeMeridianRef" doc="datums.xsd">
  <uses name="gml:PrimeMeridianRefType"/>
</def>
<def name="gml:PrimeMeridianRefType" doc="datums.xsd">
  <uses name="gml:PrimeMeridian"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:Ellipsoid" doc="datums.xsd">
  <uses name="gml:EllipsoidType"/>
  <uses name="gml:_CRSObject"/>
</def>
<def name="gml:EllipsoidType" doc="datums.xsd">
  <uses name="gml:AbstractCRSObjectType" derivation="extension"/>
  <uses name="gml:IdentifierType"/>
  <uses name="gml:LengthType"/>
  <uses name="gml:SecondDefiningParameter"/>
</def>
<def name="gml:ellipsoidRef" doc="datums.xsd">
  <uses name="gml:EllipsoidRefType"/>
</def>
<def name="gml:EllipsoidRefType" doc="datums.xsd">
  <uses name="gml:Ellipsoid"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:SecondDefiningParameter" doc="datums.xsd">
  <uses name="gml:SecondDefiningParameterType"/>
</def>
<def name="gml:SecondDefiningParameterType" doc="datums.xsd">
  <uses name="gml:ScaleType"/>
  <uses name="gml:LengthType"/>
</def>

<def name="gml:_CoordinateOperation" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractCoordinateOperationType"/>
  <uses name="gml:_CRSObject"/>
</def>
<def name="gml:AbstractCoordinateOperationType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractCRSObjectType" derivation="extension"/>
  <uses name="gml:ExtendedIdentifierType"/>
  <uses name="gml:_PositionalAccuracy"/>
  <uses name="gml:CRSRefType"/>
</def>
<def name="gml:coordinateOperationRef" doc="coordinateOperations.xsd">
  <uses name="gml:CoordinateOperationRefType"/>
</def>
<def name="gml:CoordinateOperationRefType" doc="coordinateOperations.xsd">
  <uses name="gml:_CoordinateOperation"/>
```



```
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:ConcatenatedOperation" doc="coordinateOperations.xsd">
  <uses name="gml:ConcatenatedOperationType"/>
  <uses name="gml:_CoordinateOperation"/>
</def>
<def name="gml:ConcatenatedOperationType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractCoordinateOperationType" derivation="extension"/>
  <uses name="gml:SingleOperationRefType"/>
</def>
<def name="gml:concatenatedOperationRef" doc="coordinateOperations.xsd">
  <uses name="gml:ConcatenatedOperationRefType"/>
  <uses name="gml:coordinateOperationRef"/>
</def>
<def name="gml:ConcatenatedOperationRefType" doc="coordinateOperations.xsd">
  <uses name="gml:CoordinateOperationRefType" derivation="restriction"/>
  <uses name="gml:ConcatenatedOperation"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_SingleOperation" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractSingleOperationType"/>
  <uses name="gml:_CoordinateOperation"/>
</def>
<def name="gml:AbstractSingleOperationType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractCoordinateOperationType" derivation="extension"/>
</def>
<def name="gml:singleOperationRef" doc="coordinateOperations.xsd">
  <uses name="gml:SingleOperationRefType"/>
  <uses name="gml:coordinateOperationRef"/>
</def>
<def name="gml:SingleOperationRefType" doc="coordinateOperations.xsd">
  <uses name="gml:CoordinateOperationRefType" derivation="restriction"/>
  <uses name="gml:_SingleOperation"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:PassThroughOperation" doc="coordinateOperations.xsd">
  <uses name="gml:PassThroughOperationType"/>
  <uses name="gml:_SingleOperation"/>
</def>
<def name="gml:PassThroughOperationType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractSingleOperationType" derivation="extension"/>
  <uses name="gml:OperationRefType"/>
</def>
<def name="gml:passThroughOperationRef" doc="coordinateOperations.xsd">
  <uses name="gml:PassThroughOperationRefType"/>
  <uses name="gml:singleOperationRef"/>
</def>
<def name="gml:PassThroughOperationRefType" doc="coordinateOperations.xsd">
  <uses name="gml:SingleOperationRefType" derivation="restriction"/>
  <uses name="gml:PassThroughOperation"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_Operation" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractOperationType"/>
  <uses name="gml:_SingleOperation"/>
</def>
<def name="gml:AbstractOperationType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractSingleOperationType" derivation="extension"/>
</def>
<def name="gml:operationRef" doc="coordinateOperations.xsd">
```

```
  <uses name="gml:OperationRefType"/>
  <uses name="gml:singleOperationRef"/>
</def>
<def name="gml:OperationRefType" doc="coordinateOperations.xsd">
  <uses name="gml:SingleOperationRefType" derivation="restriction"/>
  <uses name="gml:_Operation"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_GeneralConversion" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractGeneralConversionType"/>
  <uses name="gml:_Operation"/>
</def>
<def name="gml:AbstractGeneralConversionType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractOperationType" derivation="restriction"/>
  <uses name="gml:metaDataProperty"/>
  <uses name="gml:ExtendedIdentifierType"/>
  <uses name="gml:AbstractPositionalAccuracyType"/>
  <uses name="gml:id"/>
</def>
<def name="gml:generalConversionRef" doc="coordinateOperations.xsd">
  <uses name="gml:GeneralConversionRefType"/>
  <uses name="gml:operationRef"/>
</def>
<def name="gml:GeneralConversionRefType" doc="coordinateOperations.xsd">
  <uses name="gml:OperationRefType" derivation="restriction"/>
  <uses name="gml:_GeneralConversion"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:Conversion" doc="coordinateOperations.xsd">
  <uses name="gml:ConversionType"/>
  <uses name="gml:_GeneralConversion"/>
</def>
<def name="gml:ConversionType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractGeneralConversionType" derivation="extension"/>
  <uses name="gml:OperationMethodRefType"/>
  <uses name="gml:ParameterValueType"/>
</def>
<def name="gml:conversionRef" doc="coordinateOperations.xsd">
  <uses name="gml:ConversionRefType"/>
  <uses name="gml:generalConversionRef"/>
</def>
<def name="gml:ConversionRefType" doc="coordinateOperations.xsd">
  <uses name="gml:GeneralConversionRefType" derivation="restriction"/>
  <uses name="gml:Conversion"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_GeneralTransformation" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractGeneralTransformationType"/>
  <uses name="gml:_Operation"/>
</def>
<def name="gml:AbstractGeneralTransformationType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractOperationType" derivation="restriction"/>
  <uses name="gml:metaDataProperty"/>
  <uses name="gml:ExtendedIdentifierType"/>
  <uses name="gml:AbstractPositionalAccuracyType"/>
  <uses name="gml:CRSRefType"/>
  <uses name="gml:id"/>
</def>
<def name="gml:generalTransformationRef" doc="coordinateOperations.xsd">
  <uses name="gml:GeneralTransformationRefType"/>
```

```
  <uses name="gml:operationRef"/>
</def>
<def name="gml:GeneralTransformationRefType" doc="coordinateOperations.xsd">
  <uses name="gml:OperationRefType" derivation="restriction"/>
  <uses name="gml:_GeneralTransformation"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:Transformation" doc="coordinateOperations.xsd">
  <uses name="gml:TransformationType"/>
  <uses name="gml:_GeneralTransformation"/>
</def>
<def name="gml:TransformationType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractGeneralTransformationType" derivation="extension"/>
  <uses name="gml:OperationMethodRefType"/>
  <uses name="gml:ParameterValueType"/>
</def>
<def name="gml:transformationRef" doc="coordinateOperations.xsd">
  <uses name="gml:TransformationRefType"/>
  <uses name="gml:generalTransformationRef"/>
</def>
<def name="gml:TransformationRefType" doc="coordinateOperations.xsd">
  <uses name="gml:GeneralTransformationRefType" derivation="restriction"/>
  <uses name="gml:Transformation"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_generalParameterValue" doc="coordinateOperations.xsd">
  <uses name="gml:GeneralParameterValueType"/>
</def>
<def name="gml:GeneralParameterValueType" doc="coordinateOperations.xsd"/>
<def name="gml:parameterValue" doc="coordinateOperations.xsd">
  <uses name="gml:ParameterValueType"/>
  <uses name="gml:_generalParameterValue"/>
</def>
<def name="gml:ParameterValueType" doc="coordinateOperations.xsd">
  <uses name="gml:GeneralParameterValueType" derivation="extension"/>
  <uses name="gml:MeasureType"/>
  <uses name="gml:DMSAngleType"/>
  <uses name="gml:MeasureListType"/>
  <uses name="gml:integerList"/>
  <uses name="gml:OperationParameterRefType"/>
</def>
<def name="gml:_GeneralOperationMethod" doc="coordinateOperations.xsd">
  <uses name="gml:GeneralOperationMethodType"/>
  <uses name="gml:_CRSObject"/>
</def>
<def name="gml:GeneralOperationMethodType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractCRSObjectType" derivation="extension"/>
  <uses name="gml:IdentifierType"/>
</def>
<def name="gml:generalOperationMethodRef" doc="coordinateOperations.xsd">
  <uses name="gml:GeneralOperationMethodRefType"/>
</def>
<def name="gml:GeneralOperationMethodRefType" doc="coordinateOperations.xsd">
  <uses name="gml:_GeneralOperationMethod"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:OperationMethod" doc="coordinateOperations.xsd">
  <uses name="gml:OperationMethodType"/>
  <uses name="gml:_GeneralOperationMethod"/>
</def>
```

```
<def name="gml:OperationMethodType" doc="coordinateOperations.xsd">
  <uses name="gml:GeneralOperationMethodRefType" derivation="extension"/>
  <uses name="gml:OperationParameterRefType"/>
</def>
<def name="gml:operationMethodRef" doc="coordinateOperations.xsd">
  <uses name="gml:OperationMethodRefType"/>
  <uses name="gml:generalOperationMethodRef"/>
</def>
<def name="gml:OperationMethodRefType" doc="coordinateOperations.xsd">
  <uses name="gml:GeneralOperationMethodRefType" derivation="restriction"/>
  <uses name="gml:OperationMethod"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:OperationParameter" doc="coordinateOperations.xsd">
  <uses name="gml:OperationParameterType"/>
  <uses name="gml:_CRSObject"/>
</def>
<def name="gml:OperationParameterType" doc="coordinateOperations.xsd">
  <uses name="gml:AbstractCRSObjectType" derivation="extension"/>
  <uses name="gml:IdentifierType"/>
</def>
<def name="gml:operationParameterRef" doc="coordinateOperations.xsd">
  <uses name="gml:OperationParameterRefType"/>
</def>
<def name="gml:OperationParameterRefType" doc="coordinateOperations.xsd">
  <uses name="gml:OperationParameter"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>

<def name="gml:_PositionalAccuracy" doc="dataQuality.xsd">
  <uses name="gml:AbstractPositionalAccuracyType"/>
</def>
<def name="gml:AbstractPositionalAccuracyType" doc="dataQuality.xsd"/>
<def name="gml:AbsoluteExternalPositionalAccuracy" doc="dataQuality.xsd">
  <uses name="gml:AbsoluteExternalPositionalAccuracyType"/>
  <uses name="gml:_PositionalAccuracy"/>
</def>
<def name="gml:AbsoluteExternalPositionalAccuracyType" doc="dataQuality.xsd">
  <uses name="gml:AbstractPositionalAccuracyType" derivation="extension"/>
  <uses name="gml:MeasureType"/>
</def>
<def name="gml:RelativeInternalPositionalAccuracy" doc="dataQuality.xsd">
  <uses name="gml:RelativeInternalPositionalAccuracyType"/>
  <uses name="gml:_PositionalAccuracy"/>
</def>
<def name="gml:RelativeInternalPositionalAccuracyType" doc="dataQuality.xsd">
  <uses name="gml:AbstractPositionalAccuracyType" derivation="extension"/>
  <uses name="gml:MeasureType"/>
</def>
<def name="gml:CovarianceMatrix" doc="dataQuality.xsd">
  <uses name="gml:CovarianceMatrixType"/>
  <uses name="gml:_PositionalAccuracy"/>
</def>
<def name="gml:CovarianceMatrixType" doc="dataQuality.xsd">
  <uses name="gml:AbstractPositionalAccuracyType" derivation="extension"/>
  <uses name="gml:unitOfMeasure"/>
  <uses name="gml:CovarianceElementType"/>
</def>
<def name="gml:CovarianceElementType" doc="dataQuality.xsd"/>
```

```
<def name="gml:using" doc="observation.xsd">
  <uses name="gml:FeaturePropertyType"/>
</def>
<def name="gml:target" doc="observation.xsd">
  <uses name="gml:TargetPropertyType"/>
</def>
<def name="gml:TargetPropertyType" doc="observation.xsd">
  <uses name="gml:_Feature"/>
  <uses name="gml:_Geometry"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:subject" doc="observation.xsd">
  <uses name="gml:FeaturePropertyType"/>
  <uses name="gml:target"/>
</def>
<def name="gml:resultOf" doc="observation.xsd">
  <uses name="gml:AssociationType"/>
</def>
<def name="gml:ObservationType" doc="observation.xsd">
  <uses name="gml:AbstractFeatureType" derivation="extension"/>
  <uses name="gml:timeStamp"/>
  <uses name="gml:using"/>
  <uses name="gml:target"/>
  <uses name="gml:resultOf"/>
</def>
<def name="gml:Observation" doc="observation.xsd">
  <uses name="gml:ObservationType"/>
  <uses name="gml:_Feature"/>
</def>
<def name="gml:DirectedObservationType" doc="observation.xsd">
  <uses name="gml:ObservationType" derivation="extension"/>
  <uses name="gml:direction"/>
</def>
<def name="gml:DirectedObservation" doc="observation.xsd">
  <uses name="gml:DirectedObservationType"/>
  <uses name="gml:_Feature"/>
</def>

<def name="gml:defaultStyle" doc="defaultStyle.xsd">
  <uses name="gml:DefaultStylePropertyType"/>
  <uses name="gml:_property"/>
</def>
<def name="gml:DefaultStylePropertyType" doc="defaultStyle.xsd">
  <uses name="gml:_Style"/>
  <uses name="gml:AssociationAttributeGroup"/>
</def>
<def name="gml:_Style" doc="defaultStyle.xsd">
  <uses name="gml:AbstractStyleType"/>
  <uses name="gml:_GML"/>
</def>
<def name="gml:AbstractStyleType" doc="defaultStyle.xsd">
  <uses name="gml:AbstractGMLType" derivation="extension"/>
</def>
<def name="gml:Style" doc="defaultStyle.xsd">
  <uses name="gml:StyleType"/>
  <uses name="gml:_Style"/>
</def>
<def name="gml:StyleType" doc="defaultStyle.xsd">
  <uses name="gml:AbstractStyleType" derivation="extension"/>
  <uses name="gml:FeatureStyle"/>
```

```
  <uses name="gml:GraphStyle"/>
</def>
<def name="gml:BaseStyleDescriptorType" doc="defaultStyle.xsd">
  <uses name="gml:ScaleType"/>
  <uses name="gml:StyleVariationType"/>
  <uses name="smil20:animate"/>
  <uses name="smil20:animateMotion"/>
  <uses name="smil20:animateColor"/>
  <uses name="smil20:set"/>
</def>
<def name="gml:FeatureStyle" doc="defaultStyle.xsd">
  <uses name="gml:FeatureStyleType"/>
</def>
<def name="gml:FeatureStyleType" doc="defaultStyle.xsd">
  <uses name="gml:GeometryStyle"/>
  <uses name="gml:TopologyStyle"/>
  <uses name="gml:LabelStyle"/>
</def>
<def name="gml:GeometryStyle" doc="defaultStyle.xsd">
  <uses name="gml:GeometryStyleType"/>
</def>
<def name="gml:GeometryStyleType" doc="defaultStyle.xsd">
  <uses name="gml:BaseStyleDescriptorType" derivation="extension"/>
  <uses name="gml:symbol"/>
  <uses name="gml:LabelStyle"/>
</def>
<def name="gml:TopologyStyle" doc="defaultStyle.xsd">
  <uses name="gml:TopologyStyleType"/>
</def>
<def name="gml:TopologyStyleType" doc="defaultStyle.xsd">
  <uses name="gml:BaseStyleDescriptorType" derivation="extension"/>
  <uses name="gml:symbol"/>
  <uses name="gml:LabelStyle"/>
</def>
<def name="gml:LabelStyle" doc="defaultStyle.xsd">
  <uses name="gml:LabelStyleType"/>
</def>
<def name="gml:LabelStyleType" doc="defaultStyle.xsd">
  <uses name="gml:BaseStyleDescriptorType" derivation="extension"/>
  <uses name="gml:LabelType"/>
</def>
<def name="gml:GraphStyle" doc="defaultStyle.xsd">
  <uses name="gml:GraphStyleType"/>
</def>
<def name="gml:GraphStyleType" doc="defaultStyle.xsd">
  <uses name="gml:BaseStyleDescriptorType" derivation="extension"/>
  <uses name="gml:GraphTypeType"/>
  <uses name="gml:DrawingTypeType"/>
  <uses name="gml:LineTypeType"/>
  <uses name="gml:AesheticCriteriaType"/>
</def>
<def name="gml:symbol" doc="defaultStyle.xsd">
  <uses name="gml:SymbolType"/>
</def>
<def name="gml:SymbolType" doc="defaultStyle.xsd">
  <uses name="gml:AssociationType" derivation="extension"/>
  <uses name="gml:SymbolTypeEnumeration"/>
  <uses name="gml:transform"/>
</def>
<def name="gml:SymbolTypeEnumeration" doc="defaultStyle.xsd"/>
```

```
<def name="gml:LabelType" doc="defaultStyle.xsd">
  <uses name="gml:transform"/>
</def>
<def name="gml:transform" doc="defaultStyle.xsd"/>
<def name="gml:StyleVariationType" doc="defaultStyle.xsd">
  <uses name="string" derivation="extension"/>
</def>
<def name="gml:GraphTypeType" doc="defaultStyle.xsd"/>
<def name="gml:DrawingTypeType" doc="defaultStyle.xsd"/>
<def name="gml:LineTypeType" doc="defaultStyle.xsd"/>
<def name="gml:AesheticCriteriaType" doc="defaultStyle.xsd"/>

<def name="smil20:structureModuleAttrs" doc="../smil/smil20.xsd">
  <uses name="xml:lang"/>
</def>
<def name="smil20:skipContentAttrs" doc="../smil/smil20.xsd"/>
<def name="smil20:alternateContentAttrs" doc="../smil/smil20.xsd"/>
<def name="smil20:nonNegativeDecimalType" doc="../smil/smil20.xsd"/>
<def name="smil20:animate" doc="../smil/smil20.xsd">
  <uses name="smil20lang:animateType"/>
  <uses name="smil20lang:animate"/>
</def>
<def name="smil20:animatePrototype" doc="../smil/smil20.xsd">
  <uses name="smil20:animNamedTargetAttrs"/>
  <uses name="smil20:animAddAccumAttrs"/>
  <uses name="smil20:animValuesAttrs"/>
</def>
<def name="smil20:animNamedTargetAttrs" doc="../smil/smil20.xsd"/>
<def name="smil20:animAddAccumAttrs" doc="../smil/smil20.xsd"/>
<def name="smil20:animValuesAttrs" doc="../smil/smil20.xsd">
  <uses name="smil20:animSetValuesAttrs"/>
</def>
<def name="smil20:animSetValuesAttrs" doc="../smil/smil20.xsd"/>
<def name="smil20:animTargetAttrs" doc="../smil/smil20.xsd"/>
<def name="smil20:animModeAttrs" doc="../smil/smil20.xsd"/>
<def name="smil20:animateMotion" doc="../smil/smil20.xsd">
  <uses name="smil20lang:animateMotionType"/>
  <uses name="smil20lang:animateMotion"/>
</def>
<def name="smil20:animateMotionPrototype" doc="../smil/smil20.xsd">
  <uses name="smil20:animAddAccumAttrs"/>
  <uses name="smil20:animValuesAttrs"/>
</def>
<def name="smil20:animateColor" doc="../smil/smil20.xsd">
  <uses name="smil20lang:animateColorType"/>
  <uses name="smil20lang:animateColor"/>
</def>
<def name="smil20:animateColorPrototype" doc="../smil/smil20.xsd">
  <uses name="smil20:animNamedTargetAttrs"/>
  <uses name="smil20:animAddAccumAttrs"/>
  <uses name="smil20:animValuesAttrs"/>
</def>
<def name="smil20:set" doc="../smil/smil20.xsd">
  <uses name="smil20lang:setType"/>
  <uses name="smil20lang:set"/>
</def>
<def name="smil20:setPrototype" doc="../smil/smil20.xsd">
  <uses name="smil20:animNamedTargetAttrs"/>
  <uses name="smil20:animSetValuesAttrs"/>
</def>
```

```
<def name="smil20:syncBehaviorAttrs" doc="../smil/smil20.xsd">
  <uses name="smil20:syncBehaviorType"/>
</def>
<def name="smil20:syncBehaviorType" doc="../smil/smil20.xsd"/>
<def name="smil20:syncBehaviorDefaultAttrs" doc="../smil/smil20.xsd">
  <uses name="smil20:syncBehaviorDefaultType"/>
</def>
<def name="smil20:syncBehaviorDefaultType" doc="../smil/smil20.xsd"/>
<def name="smil20:restartTimingAttrs" doc="../smil/smil20.xsd">
  <uses name="smil20:restartTimingType"/>
</def>
<def name="smil20:restartTimingType" doc="../smil/smil20.xsd"/>
<def name="smil20:restartDefaultAttrs" doc="../smil/smil20.xsd">
  <uses name="smil20:restartDefaultType"/>
</def>
<def name="smil20:restartDefaultType" doc="../smil/smil20.xsd"/>
<def name="smil20:fillTimingAttrs" doc="../smil/smil20.xsd">
  <uses name="smil20:fillTimingAttrsType"/>
</def>
<def name="smil20:fillTimingAttrsType" doc="../smil/smil20.xsd"/>
<def name="smil20:fillDefaultAttrs" doc="../smil/smil20.xsd">
  <uses name="smil20:fillDefaultType"/>
</def>
<def name="smil20:fillDefaultType" doc="../smil/smil20.xsd"/>
<def name="smil20:beginEndTimingAttrs" doc="../smil/smil20.xsd"/>
<def name="smil20:durTimingAttrs" doc="../smil/smil20.xsd"/>
<def name="smil20:repeatTimingAttrs" doc="../smil/smil20.xsd">
  <uses name="smil20:nonNegativeDecimalType"/>
</def>
<def name="smil20:deprecatedRepeatTiming" doc="../smil/smil20.xsd"/>
<def name="smil20:minMaxTimingAttrs" doc="../smil/smil20.xsd"/>

<def name="smil20lang:animate" doc="../smil/smil20-language.xsd">
  <uses name="smil20lang:animateType"/>
</def>
<def name="smil20lang:animateType" doc="../smil/smil20-language.xsd">
  <uses name="smil20:animatePrototype" derivation="extension"/>
  <uses name="smil20lang:CoreAttrs"/>
  <uses name="smil20lang:TimingAttrs"/>
  <uses name="smil20:animTargetAttrs"/>
  <uses name="smil20:animModeAttrs"/>
  <uses name="smil20:skipContentAttrs"/>
</def>
<def name="smil20lang:CoreAttrs" doc="../smil/smil20-language.xsd">
  <uses name="smil20:structureModuleAttrs"/>
  <uses name="smil20:alternateContentAttrs"/>
</def>
<def name="smil20lang:TimingAttrs" doc="../smil/smil20-language.xsd">
  <uses name="smil20lang:BasicTimingAttrs"/>
  <uses name="smil20:syncBehaviorAttrs"/>
  <uses name="smil20:syncBehaviorDefaultAttrs"/>
  <uses name="smil20:restartTimingAttrs"/>
  <uses name="smil20:restartDefaultAttrs"/>
  <uses name="smil20:fillTimingAttrs"/>
  <uses name="smil20:fillDefaultAttrs"/>
</def>
<def name="smil20lang:BasicTimingAttrs" doc="../smil/smil20-language.xsd">
  <uses name="smil20:beginEndTimingAttrs"/>
  <uses name="smil20:durTimingAttrs"/>
  <uses name="smil20:repeatTimingAttrs"/>
```

```
  <uses name="smil20:deprecatedRepeatTiming"/>
  <uses name="smil20:minMaxTimingAttrs"/>
</def>
<def name="smil20lang:animateMotion" doc="../smil/smil20-language.xsd">
  <uses name="smil20lang:animateMotionType"/>
</def>
<def name="smil20lang:animateMotionType" doc="../smil/smil20-language.xsd">
  <uses name="smil20:animateMotionPrototype" derivation="extension"/>
  <uses name="smil20lang:CoreAttrs"/>
  <uses name="smil20lang:TimingAttrs"/>
  <uses name="smil20:animTargetAttrs"/>
  <uses name="smil20:animModeAttrs"/>
  <uses name="smil20:skipContentAttrs"/>
</def>
<def name="smil20lang:animateColor" doc="../smil/smil20-language.xsd">
  <uses name="smil20lang:animateColorType"/>
</def>
<def name="smil20lang:animateColorType" doc="../smil/smil20-language.xsd">
  <uses name="smil20:animateColorPrototype" derivation="extension"/>
  <uses name="smil20lang:CoreAttrs"/>
  <uses name="smil20lang:TimingAttrs"/>
  <uses name="smil20:animTargetAttrs"/>
  <uses name="smil20:animModeAttrs"/>
  <uses name="smil20:skipContentAttrs"/>
</def>
<def name="smil20lang:set" doc="../smil/smil20-language.xsd">
  <uses name="smil20lang:setType"/>
</def>
<def name="smil20lang:setType" doc="../smil/smil20-language.xsd">
  <uses name="smil20:setPrototype" derivation="extension"/>
  <uses name="smil20lang:CoreAttrs"/>
  <uses name="smil20lang:TimingAttrs"/>
  <uses name="smil20:animTargetAttrs"/>
  <uses name="smil20:skipContentAttrs"/>
</def>

<def name="xml:lang" doc="../smil/xml-mod.xsd"/>
<def name="xml:space" doc="../smil/xml-mod.xsd"/>
<def name="xml:specialAttrs" doc="../smil/xml-mod.xsd">
  <uses name="xml:lang"/>
  <uses name="xml:space"/>
</def>
</depends>
```


Bibliography

ISO 31 (all parts), *Quantities and units*.

ISO 1000, *SI units and recommendations for the use of their multiples and of certain other units*.

Patterns in GML. OGC document 02-004

Cover Pages Technology Report at http://xml.coverpages.org/geographyML.html

Langran, G. *Time in Geographic Information Systems.* London: Taylor & Francis Ltd. 1992.

Kaufman, M. & Wagner D. *Drawing Graphs*, Springer LNCS 2025, 1998.

Battista, G., Eades P., Tamassia R. & Tollis I. *Graph Drawing*, Prentice Hall 1999.