Open Geospatial Consortium Inc.

Date:   2012-09-05

Reference number of this OGC® project document:    OGC 12-110

OGC name of this OGC® project document:    http://www.opengis.net/doc/gml-nil/1.0

Version: 1.0rc

Category: OGC® Implementation Specification

Editor:   Simon J D Cox

Nillable

**Copyright notice**

The companies listed above have granted the Open Geospatial Consortium, Inc. (OGC) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version.

**Warning**

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

Document type:    OGC® Publicly Available Standard

Document subtype:    if applicable

Document stage:    Draft

Document language:  English

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Contents

i. Abstract iv

ii. Keywords iv

iii. Preface iv

iv. Submitting organizations iv

v. Submission contact points v

vi. Changes to the OGC® Abstract Specification v

Foreword vi

Introduction vii

1 Scope 1

2 Conformance 1

2.1 Overview 1

2.2 Specification identifier 2

2.3 Conformance classes 2

3 Normative references 2

4 Terms and definitions 3

5 Conventions 5

5.1 Abbreviated terms 5

5.2 UML Notation 5

5.3 Schema language 5

5.4 Namespace abbreviations 5

5.5 Document presentation of the specification 5

6 Overview 6

6.1 Motivation 6

6.2 Summary of solution 6

7 Nillable setting for information models formalized using UML 7

7.1 General 7

7.2 Package defaults 8

7.3 Property overrides 9

7.4 Imported package overrides 10

8 Nillable setting for GML Application Schemas 11

8.1 General 11

8.2 Schema defaults 12

8.3 Xlink attributes on property elements 13

8.4 Overriding the default 14

8.5 Overriding the default for imported schema 15

8.6 Example schema (informative) 16

9 Nil values in GML data 17

9.1 General 17

9.2 Special URIs in xlinks 18

9.3 XML document is schema-valid 19

10 Client application consuming GML data 20

10.1 General 20

10.2 Nil overrides element content 20

Annex A (normative) Abstract test suite 22

A.1 General 22

A.2 Nillable setting for information models formalized using UML 22

A.3 Nillable setting for GML Application Schemas 24

A.4 Nil values in GML data 26

A.5 Client application consuming GML data 27

Annex B (normative) XML Schema for simple-content types 29

B.1 General 29

B.2 Schema (informative) 29

Bibliography 33

Revision history 34

1. Abstract

This OGC™ Standard defines a profile of GML in which the default value for the nillable behaviour of feature- and object-properties may be set to “true”. The default nillable value is set at the level of an application, and applies to all properties appearing in the application. The default of nillable=“true” established by this standard complements the usual setting, which inherits the default nillable=“false” from W3C XML Schema. This OGC**®** Standard is applicable to communities where for technical or business reasons some legitimate community members are not able to provide values for all mandatory properties, and the community is willing to build applications that accept incomplete data.

1. Keywords

GML, Application Schema, nil, void, nillable, community, XML, URI, xlink

1. Preface

This OGC Standard defines a GML profile for communities whose members have variable ability to deliver all the information defined in the community application schema. It has emerged from work undertaken over a number of years in the earth and environmental science communities.

1. Submitting organizations

The following organizations submitted this Implementation Specification to the Open Geospatial Consortium Inc. as a Request For Comment (RFC):

1. CSIRO Australia
2. Landcare Research New Zealand Ltd
3. British Geological Survey
4. Submission contact points

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

|  |  |
| --- | --- |
| CONTACT | COMPANY |
| Simon Coxsimon.cox@csiro.au  | CSIRO Australia |
| Alistair RitchieRitchieA@landcareresearch.nz  | Landcare Research New Zealand Ltd |
| Tim Duffytrd@bgs.ac.uk | British Geological Survey |

1. Changes to the OGC® Abstract Specification

The OGC**®** Abstract Specification does not require changes to accommodate this OGC**®** standard. However, the new stereotypes and tagged values proposed in this standard should be added to ISO 19109 Geographic Information – Rules for Application Schema when it is revised (underway 2011-2013).

Foreword

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

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

Introduction

This OGC Standard defines a GML profile for communities whose members have varying ability to deliver all the information defined in their community application schema. The patterns described here have emerged from work undertaken over a number of years in the earth and environmental science communities. Statutory agencies are a key provider of data in these domains. They typically have a long history of data collection, with changing standards but limited ability to remediate older datasets which are, nevertheless, still highly valuable. They are also involved in a mixture of public-good and commercial engagements, which may affect the access rights to some elements of the data. As a result, not all participants can provide all the data items expected in the schema, but it is difficult at the time of design of the applications schema to identify all those items that have the potential to be missing at run time. For these communities a generic method for marking ‘nil’ values is required. This specification provides such a method.

GML-nil

# Scope

This OGC™ Standard defines a profile of GML in which the default value for the nillable behaviour of feature- and object-properties may be set to “true”. The default nillable value is set at the level of an application, and applies to all properties appearing in the application, both those defined directly in the application and also those imported from dependencies.

The default of nillable=“true” established by this standard complements the usual setting, which inherits the default nillable=“false” from W3C XML Schema. This standard provides a uniform mechanism to mark any property in a data instance “nil” when the application schema has been so flagged.

This OGC**®** Standard is applicable to communities where for technical or business reasons some legitimate community members are not able to provide values for all mandatory properties, and the community is willing to build applications that accept incomplete data.

# Conformance

## Overview

This OGC™standard defines a mechanism for enabling lax conformance to the requirements of an application schema, specifically by allowing any feature property to have a nil or void value in a dataset. Requirements for four standardization target types are considered.

1. A community information model formalized in UML
2. An GML Application Schema that implements the model
3. XML data instances that conform to the schema
4. Client applications consuming GML data.

## Specification identifier

The implementation standard described in this document is identified as [http://www.opengis.net/spec/gml-nil/1.0](http://www.opengis.net/spec/gml-lax/1.0) . All requirements classes and conformance test classes owned by this standard are identified by URIs with this base. In this document the prefix “gml-nil” is used as an abbreviation for <http://www.opengis.net/spec/gml-nil/1.0/> .

## Conformance classes

In order to conform to this OGC™standard, an implementation shall pass all tests in any of the four conformance classes shown on Table 1.

Table 1 — Conformance classes related Observations and Measurements instances

|  |  |  |
| --- | --- | --- |
| **Conformance class** | **Description** | **Clause** |
| [gml-nil:conf/UMLNillable](http://www.opengis.net/spec/gml-lax/1.0/conf/UMLNillable) | UML formalization of community information model | A.2 |
| [gml-nil:conf/GMLSchemaNillable](http://www.opengis.net/spec/gml-lax/1.0/conf/GMLSchemaNillable) | GML Application Schema | 3 |
| [gml-nil:conf/GMLDataNillable](http://www.opengis.net/spec/gml-lax/1.0/conf/GMLDataNillable%20)   | GML data | A.4 |
| [gml-nil:conf/GMLConsumer](http://www.opengis.net/spec/gml-lax/1.0/conf/GMLDataNillable%20)   | Client application consuming GML data | A.5 |

#  Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this standard. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this standard 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.

ISO 19109:2006 *Geographic Information-Rules for application schema.*

ISO/TS 19103:2005 *Geographic Information-Conceptual Schema language.*

OGC Geography Markup Language v3.2 OGC Document 07-036 <http://www.opengis.net/doc/gml/3.2.1> (also published as ISO 19136:2007, Geographic information — Geography Markup Language)

OGC-NA Name type specification - definitions: Part 1 - basic name OGC Document 09-048r3 <http://www.opengis.net/doc/def-names-1>

 Policy Directives for Writing and Publishing OGC Standards: TC Decisions. OGC Document 06-135r11 <http://www.opengis.net/doc/policy/2.0>

The Specification Model — A Standard for Modular specifications OGC Document 08-131r3. <http://www.opengis.net/doc/modular-spec>

XML Schema Part 1: Structures Second Edition. W3C Recommendation (28 October 2004) <http://www.w3.org/TR/xmlschema-1/>

#  Terms and definitions

For the purposes of this document, the following terms and definitions apply.

all-components document

XML schema document that includes, either directly or indirectly, all of the components defined and declared in a namespace

[OGC Policy Directives. http://www.opengis.net/doc/POL/Standards/2.0.3]

1.

application schema

conceptual schema for data required by one or more applications

[ISO 19101:2002, definition 4.2]

element <XML>

basic information item of an XML document containing **child elements**, **attributes** and character data

[ISO 19136:2007]

NOTE From the XML Information Set: ―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.

GML application schema

application schema written in XML Schema in accordance with the rules specified in ISO 19136:2007

[ISO 19136:2007]

GML document

XML document with a root element that is one of the elements AbstractFeature, Dictionary or TopoComplex specified in the GML schema or any element of a substitution group of any of these elements

[ISO 19136:2007]

GML schema

schema components in the XML namespace ―http://www.opengis.net/gml/3.2‖ as specified in ISO 19136:2007

[ISO 19136:2007]

nil <XML>

mechanism for signalling that an XML element should be accepted as valid when it has no content despite a content type which does not require or even necessarily allow empty content. An element so labelled must be empty, but can carry attributes if permitted by the corresponding complex type

NOTE also see void

[W3C XML Schema: Part 1]

nillable <XML Schema>

boolean flag to indicate if the mechanism for marking an XML element **nil** is available

[W3C XML Schema: Part 1]

schema <XML Schema>

collection of schema components within the same target namespace

EXAMPLE Schema components of W3C XML Schema are types, elements, attributes, groups, etc.

[ISO 19136:2007]

schema document <XML Schema>

XML document containing schema component definitions and declarations

NOTE The W3C XML Schema provides an XML interchange format for schema information. A single schema document provides descriptions of components associated with a single XML namespace, but several documents may describe components in the same schema, i.e. the same target namespace.

[ISO 19136:2007]

void

datatype representing an object whose presence is syntactically or semantically required, but carries no information in a given instance

NOTE also see nil

[ISO/IEC 11404:2007]

#  Conventions

## Abbreviated terms

GML Geography Markup Language

OGC Open Geospatial Consortium

UML Unified Modeling Language

XML Extensible Markup Language

XSD W3C XML Schema Definition Language

## UML Notation

UML is used in this document according to the profile described in ISO 19103:2005 and extended in Annex D and Annex E of GML 3.2.1.

## Schema language

The XML implementation specified in this Standard is described using the XML Schema language (XSD) [*XML Schema Part 1: Structures* , *XML Schema Part 2: Datatypes*].

## Namespace abbreviations

Within this document the following prefixes are used to substitute for the URIs indicated

gml: <http://www.opengis.net/gml/3.2>

gml-nil: <http://www.opengis.net/spec/gml-nil/1.0/>

myns: dummy XML namespace for an application schema

xlink: http://www.w3.org/1999/xlink

xsd: <http://www.w3.org/2001/XMLSchema>

xsi: http://www.w3.org/2001/XMLSchema-instance

## Document presentation of the specification

This document uses a representation that follows the structures defined in the OGC Policy []. All normative material is organized as requirements, requirements classes, conformance tests and conformance classes. Each is identified with a URI, and the content and dependencies are described in tables whose structure matches the specification model.

#  Overview

## Motivation

When sharing data in heterogeneous environments, interoperability is enhanced by use of a model and schema agreed by the community (a community application schema). However, in practice, values may not be available for some mandatory feature-properties from some suppliers (... at some times and to some audiences ...). This may be for legitimate technical- or business-reasons.

To accommodate this, a property instance in a dataset can carry the XML attribute @xsi:nil=“true” to indicate that no value is supplied for a property whose appearance is not optional (GML 3.2.1 clause 8.2.3.2). This may also be supplemented with a @nilReason attribute giving an explanation (GML 3.2.1 clause 8.2.3.1). This mechanism can be used at run-time for properties that are marked nillable=‘true’ in the XML schema. These typically correspond to properties that were classified ‘voidable’ or ‘nillable’ at design-time. (There is an INSPIRE UML-XML rule covering this, which augments the standard GML 3.2 ruleset.) The default setting is nillable=‘false’, which corresponds to the W3C XML Schema specification.

However, there is a desire in some communities to allow data providers to ‘do their best’ to provide data that conforms to the community model, and accept incomplete data in preference to no data. For these communities a mechanism is required to allow a data provider to mark feature properties ‘nil’, without the data modellers having to anticipate each and every ‘voidable’ property at design time. For these applications the default of nillable=’false’ is too strict.

Different communities may legitimately have different views about how strict their conformance requirements are. Reflecting this, it is useful to be able to declare the default value at the *application* level, so that flexible (inclusive) communities can have a default value corresponding to nillable=‘true’, while strict (exclusive) communities can have the default of nillable=‘false’. The latter are already accommodated by the nillable capability provided by W3C XML Schema. But since the W3C XML Schema default is nillable=‘false’ a different mechanism is required to accommodate flexible communities.

## Summary of solution

The standard GML pattern for implementing feature properties has ‘xlink’ attributes available on property elements, to support values being given by-reference as an alternative to the value being shown inline (GML 3.2.1 clause 7.2.3.3). Data containing property elements with XLink attributes and no content is schema-valid. The Gml-nil profile defines a pattern where the xlink attributes may carry certain key values to indicate that the property value is nil or void. The key values are the standard OGC nil URIs, currently:

* <http://www.opengis.net/def/nil/OGC/0/unknown>
	+ The correct value is not known to, or not computable by, the sender of this data. However, the correct value probably exists
* <http://www.opengis.net/def/nil/OGC/0/missing>
	+ The correct value is not readily available to the sender of this data. Furthermore, a correct value may not exist
* <http://www.opengis.net/def/nil/OGC/0/withheld>
	+ The value is not divulged.
* http://www.opengis.net/def/nil/OGC/0/inapplicable
	+ There is no value
* <http://www.opengis.net/def/nil/OGC/0/template>
	+ The value will be available later
* <http://www.opengis.net/def/nil/OGC/0/AboveDetectionRange>
	+ Value was above the detection range of the instrument used to estimate it.
* <http://www.opengis.net/def/nil/OGC/0/BelowDetectionRange>
	+ Value was below the detection range of the instrument used to estimate it.

An interpretation of this at a conceptual or model level is that, when used as the value of an object property in a data instance, the nil URI denotes a resource of the required class or type, but with no value.

This behaviour is permitted where the application schema carries a tag allowing nillableProperties as explained below. This setting applies to the application schema as-a-whole, but may be overridden at a property level where it is desired to disallow a nil value for a property indicated by any mechanism. The xsi:nil and gml:nilReason attributes may be available on some properties declared in imported schemas, the expectation is that, when an application schema is marked in this way, then the xlink method would be preferred in data instances conforming to this schema, in order that a uniform processing approach can be used.

# Nillable setting for information models formalized using UML

## General

A community information model can be formalized using UML, following the ISO 19109 *Rules for Application Schema*, and using the UML profile given in ISO/TS 19103 *Conceptual Schema Language*. This clause describes the requirements for indicating the default setting for ‘nillable’ for an application schema as a whole, and for overriding the default setting on specific class properties.

|  |
| --- |
| **Requirements class** |
| gml-nil:req/UMLNillable |
| **Target type** | UML Model |
| **Dependency** | <http://standards.iso.org/iso/19103> (Conceptual schema language) |
| **Dependency** | <http://standards.iso.org/iso/19109> (Rules for application schema) |
| **Requirement** | gml-nil:req/UMLNillable/package  |
| **Requirement** | gml-nil:req/UMLNillable/property |
| **Requirement** | gml-nil:req/UMLNillable/import |

## Package defaults

In this standard, the default setting for ‘nillable’ is scoped to the UML package containing an application schema. A tagged value *voidableProperties*[[1]](#footnote-1) is used (Figure 1). If set ‘true’ then all class attributes and association roles are voidable.

**Figure 1: (informative) – Example of an application schema package with the voidableProperties tagged value**

|  |
| --- |
| **Requirement** |
| gml-nil:req/UMLNillable/package |
| An application schema in which all class attributes and association roles are nillable SHALL be indicated using a tag *voidableProperties* with the value *‘true’* on the UML package containing the classes and associations in the schema. The default value for *voidableProperties* SHALL be “false”. The value specified for voidableProperties SHALL apply to all class attributes and association roles in the application schema, including those defined in its dependency packages.  |

## Property overrides

The nillable behavior of any specific class attribute or association role is set using a stereotype *«voidable»* or *«notvoidable»*.

With the default package setting *voidableProperties=“false”,* then an individual attribute or association role with the stereotype *«voidable»* is identified as an attribute or association role which may be void in an instance.

If the package setting is *voidableProperties=“true”*, then this may be overridden by stereotyping an individual attribute or association role *«notvoidable»,* thereby identifying those attributes or association roles which may not be void and must have valid values at the instance level (Figure 2).

NOTE: the stereotype *«voidable»* was introduced in the INSPIRE Generic Conceptual Model [1], so the same mechanism is used here for the complementary purpose.

**Figure 2: (informative) – Example of a feature type with some attributes and association roles stereotyped *«notvoidable».***

|  |
| --- |
| **Requirement** |
| gml-nil:req/UMLNillable/property |
| Within an application schema, any class attribute or association role MAY receive the stereotype *«voidable» or* *«notvoidable»*. The stereotype *«notvoidable»* overrides the default for individual attributes and association roles in the case when the package is tagged voidableProperties=“true”. . |

## Imported package overrides

The nillable behavior of any class attribute or association role in any specific imported application schema package is set using a stereotype *«voidable», «notvoidable» or «preservevoidable»*.

With the default package setting *voidableProperties=“false”,* then an individual package import with the stereotype *«voidable»* identifies an application schema in which all class attributes and association roles are nillable.

If the package setting is *voidableProperties=“true”*, then this may be overridden by stereotyping an individual package import *«notvoidable»,* thereby identifying those application schema in which all class attributes and association roles may not be void (Figure 3).

If an individual package import is stereotyped *«preservevoidable»,* then the default nillable behavior, and any subsequent property-specific overrides, of the imported application schema shall be preserved.

The package import setting will propagate to all packages imported by the target package. If there are specific requirements for an indirectly imported package, or if this causes conflict – for example, a package is imported by several packages, each applying different settings – then the settings should be applied using a direct package import.

**Figure 3: (informative) – Example of an application schema package with an imported application schema package stereotyped «preservevoidable». Package1 explicitly imports Package4 with the setting «notvoidable» to resolve the conflicting settings arising from the indirect import via Package2 and Package3.**

|  |
| --- |
| **Requirement** |
| gml-nil:req/UMLNillable/import |
| Within an application schema, any application schema package import MAY receive the stereotype «voidable», «notvoidable» or «preservevoidable».The stereotype «notvoidable» SHALL override the default for the application schema in the case when its package is tagged voidableProperties=“true”.The stereotype «voidable» SHALL override the default for the application schema in the case when its package is tagged voidableProperties=“false”.The stereotype «preservevoidable» SHALL preserve the default nillable behavior, and any subsequent property overrides, of the imported application schema package.The value specified SHALL apply to all properties in the imported schema, including those imported from packages NOT explicitly imported by the source application schema. |

# Nillable setting for GML Application Schemas

## General

A community information model can be formalized as a GML Application Schema, which describes and provides validation constraints for XML data transfer. The GML Application Schema may be designed directly using the W3C XML Schema language, importing the GML components and following the patterns in the GML standard (GML 3.2.1), or may be converted from a model formalized in UML using the rules from GML 3.2.1 Annex E. This clause describes the requirements for indicating the default setting for ‘nillable’ for an application schema as a whole, and for overriding the default setting on specific class properties.

|  |
| --- |
| **Requirements class** |
| gml-nil:req/GMLSchemaNillable |
| **Target type** | GML Application Schema |
| **Dependency** | <http://www.opengis.net/spec/gml> (Geography Markup Language) |
| **Dependency** | <http://www.opengis.net/doc/policy/2.0> (Policies Related to OGC Standards) |
| **Requirement** | gml-nil:req/GMLSchemaNillable/package  |
| **Requirement** | gml-nil:req/GMLSchemaNillable/propertyLinks |
| **Requirement** | gml-nil:req/GMLSchemaNillable/propertyNillable |
| **Requirement** | gml-nil:req/GMLSchemaNillable/import |

## Schema defaults

In this standard, the default setting for ‘nillable’ is scoped to the GML application schema (GML 3.2.1) and attached to the ‘all-components’ schema document (Policies). If an <appinfo> element containing the value *nillableProperties* is present then all object properties (in the sense described in GML 3.2.1) are nillable.

|  |
| --- |
| **Requirement** |
| gml-nil:req/GMLSchemaNillable/package |
| A GML application schema in which all properties are nillable by default SHALL be indicated by including the element <xsd:appinfo  source="http://www.opengis.net/spec/Gml-nil/" >nillableProperties</xsd:appinfo>as a sub-element of the <annotation> element in the ‘all-components’ schema document. The value specified for nillableProperties SHALL apply to all properties in the GML application schema, including those imported from other namespaces.  |

The following listing shows an example of an abbreviated XML Schema document with the nillableProperties header included:

<xsd:schema targetNamespace="http://www.example.org/mySchema"
 xmlns:myns="http://www.example.org/mySchema" xmlns:gml="http://www.opengis.net/gml/3.2"
 xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 elementFormDefault="qualified">
 <xsd:annotation>
 <xsd:appinfo source="http://www.opengis.net/spec/gml-nil/">nillableProperties</xsd:appinfo>
 </xsd:annotation>
 <xsd:import namespace="http://www.opengis.net/gml/3.2"
 schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
…
</xsd:schema>

## Xlink attributes on property elements

This standard specifies that in an XML data instance, a property may be marked ‘nil’ or ‘void’ through the use of one of a set of special URIs as the value of a xlink:href attribute (clause 9). Therefore, in order to enable use of this mechanism, every nillable property must have the xlink attributes included in the schema definition of the property element type.

Properties that are defined using the default GML pattern for associations (GML 3.2.1 clause 7.2.3.3) automatically have xlinks available, so many properties in most GML applications schemas are already enabled. But any properties defined using the pattern for inline properties (GML 3.2.1 clause 7.2.3.8) and all properties with simple content (GML 3.2.1 clause 7.2.3.1 paragraph 2) will not have xlinks available. Therefore there are additional requirement to enable the mechanism on all properties.

|  |
| --- |
| **Requirement** |
| gml-nil:req/GMLSchemaNillable/propertyLinks |
| Within a GML application schema in which all properties are nillable by default, the type of every XML element representing a nillable object property SHALL carry the gml:AssociationAttributeGroup. The application schema MAY use the types defined in the GML-NIL XML Schema documented in Annex B.  |

Properties with complex (element) content can use the GML AssociationRoleType pattern, described in GML 3.2.1 clause 7.2.3.3:

 <xsd:complexType name="Class2PropertyType">
 <xsd:sequence minOccurs="0">
 <xsd:element ref="myns:Class2"/>
 </xsd:sequence>
 <xsd:attributeGroup ref="gml:OwnershipAttributeGroup"/>
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:complexType>

where *myns:Class2* is an element name. This pattern should be used regardless of whether the content is required to be inline or by-reference, since even an inline property will be empty when nilled, so the minOccurs=”0” value is required to retain schema validity.

Properties with simple content can use the types from the schema shown in Annex B, or by following the pattern:

 <xsd:complexType name="AType">
 <xsd:simpleContent>
 <xsd:extension base="ns:AType">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

where *ns:AType* is an example of a simple type.

NOTE: gml-nil:req/GMLSchemaNillable/package specifies that the nillable default applies to imported properties as well as those defined locally. However, the degree to which this can be applied will also depend on the XML schema implementation of the dependencies. For example, for simple-content properties it is unlikely that xlinks will be available on imported XML elements in pre-existing XML namespaces.

## Overriding the default

The default value for the schema can be overridden locally on any property element. The standard xsd:nillable attribute is used to indicate a specific behavior on a single property element.

|  |
| --- |
| **Requirement** |
| gml-nil:req/GMLSchemaNillable/propertyNillable |
| Within a GML application schema, any element that implements an object property MAY carry the attribute xsd:nillable. The value of this attribute SHALL override the value for the application schema as a whole.  |

The following listing shows an example of a feature type definition with some properties marked nillable=“false”:

 <xsd:element name="Class1" type="myns:Class1Type"/>
 <xsd:complexType name="Class1Type">
 <xsd:complexContent>
 <xsd:extension base="gml:AbstractFeatureType">
 <xsd:sequence>
 <xsd:element name="propM" type="gmln:MeasureType"/>
 <xsd:element name="propS" type="xsd:string" nillable="false"/>
 <xsd:element name="propB" type="myns:Class2PropertyType"/>
 <xsd:element name="propC" type="myns:Class3PropertyType" nillable="false"/>
 </xsd:sequence>
 </xsd:extension>
 </xsd:complexContent>
 </xsd:complexType>

## Overriding the default for imported schema

The default value for the application schema can be overridden locally on any schema import element. The <appinfo> element is used to indicate the specific setting required for the imported schema.

If the <appinfo> element containing the value *nillableProperties* is absent then an individual schema import with the <appinfo> element containing the value *nillable* identifies a schema in which all imported object properties are nillable.

If the <appinfo> element containing the value *nillableProperties* is present then an individual schema import with the <appinfo> element containing the value *notNillable* identifies a schema in which all imported object properties are not nillable.

An individual schema import with the <appinfo> element containing the value *preserveNillable* identifies a schema in which all imported nillable settings must be preserved.

The schema import setting will propagate to all schema imported by the target schema. If there are specific requirements for indirectly imported schema, then these should be specified locally using an explicit schema import with the appropriate <appinfo> value.

|  |
| --- |
| **Requirement** |
| gml-nil:req/GMLSchemaNillable/import |
| Within a GML application schema, the nillable property settings of an imported schema MAY be indicated by including the element<xsd:appinfo source="http://www.opengis.net/spec/Gml-nil/" >*nillable*</xsd:appinfo>or<xsd:appinfo source="http://www.opengis.net/spec/Gml-nil/" >*notNillable*</xsd:appinfo>or<xsd:appinfo source="http://www.opengis.net/spec/Gml-nil/" >*preserveNillable*</xsd:appinfo>as a sub-element of the <annotation> element in the schema import element. The value specified SHALL apply to all properties in the imported schema, including those imported from other namespaces NOT explicitly imported by the GML application schema. |

The following listing shows an example of an abbreviated XML Schema document with the nillableProperties header and a preserveNillable schema import overriding this setting:

<xsd:schema targetNamespace="http://www.example.org/mySchema"
 xmlns:myns="http://www.example.org/mySchema" xmlns:tns="http://www.example.org/theirSchema"
 xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
 <xsd:annotation>
 <xsd:appinfo source="http://www.opengis.net/spec/gml-nil/">nillableProperties</xsd:appinfo>
 </xsd:annotation>
 <xsd:import namespace="http://www.opengis.net/gml/3.2"
 schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
 <xsd:import namespace="http://www.example.org/theirSchema"
 schemaLocation="http://schemas.example.org/theirs/schema.xsd">
 <xsd:annotation>
 <xsd:appinfo source="http://www.opengis.net/spec/gml-nil/">preserveNillable</xsd:appinfo>
 </xsd:annotation>
 </xsd:import>
…
</xsd:schema>

## Example schema (informative)

The following listing is a complete example of an application schema using all the mechanisms defined in this requirements class:

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema targetNamespace="http://www.example.org/mySchema"
 xmlns:myns="http://www.example.org/mySchema" xmlns:tns="http://www.example.org/theirSchema"
 xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
 <xsd:annotation>
 **<xsd:appinfo source="http://www.opengis.net/spec/gml-nil/">nillableProperties</xsd:appinfo>** </xsd:annotation>
 <xsd:import namespace="http://www.opengis.net/gml/3.2"
 schemaLocation="http://schemas.opengis.net/gml/3.2.1/gml.xsd"/>
 <xsd:import namespace="http://www.example.org/theirSchema"
 schemaLocation="http://schemas.example.org/theirs/schema.xsd">
 <xsd:annotation>
 **<xsd:appinfo source="http://www.opengis.net/spec/gml-nil/">preserveNillable</xsd:appinfo>** </xsd:annotation>
 </xsd:import>
 <!-- ============================================================ -->
 <xsd:element name="Class1" type="myns:Class1Type"/>
 <xsd:complexType name="Class1Type">
 <xsd:complexContent>
 <xsd:extension base="gml:AbstractFeatureType">
 <xsd:sequence>
 <xsd:element name="propM" type="gmln:MeasureType"/>
 **<xsd:element name="propS" type="xsd:string" nillable="false"/>** <xsd:element name="propB" type="myns:Class2PropertyType"/>
 **<xsd:element name="propC" type="myns:Class3PropertyType" nillable="false"/>** </xsd:sequence>
 </xsd:extension>
 </xsd:complexContent>
 </xsd:complexType>
 <!-- ============================================================ -->
 **<xsd:complexType name="NillableMeasureType">
 <xsd:simpleContent>
 <xsd:extension base="gml:MeasureType">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>** <!-- ============================================================ -->
 **<xsd:element name="Class2"/>
 <xsd:complexType name="Class2PropertyType">
 <xsd:sequence minOccurs="0">
 <xsd:element ref="myns:Class2"/>
 </xsd:sequence>
 <xsd:attributeGroup ref="gml:OwnershipAttributeGroup"/>
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:complexType>** <!-- ============================================================ -->
 <xsd:element name="Class3"/>
 <xsd:complexType name="Class3PropertyType">
 <xsd:sequence minOccurs="0">
 <xsd:element ref="myns:Class3"/>
 </xsd:sequence>
 <xsd:attributeGroup ref="gml:OwnershipAttributeGroup"/>
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:complexType>
</xsd:schema>

# Nil values in GML data

## General

A dataset prepared for transfer can be formalized as a GML document conforming to the GML Application Schema. This clause describes the requirements for indicating that the value for a property is not available when the application schema as a whole has a default “nillableProperties” setting.

|  |
| --- |
| **Requirements class** |
| gml-nil:req/GMLDataNillable |
| **Target type** | GML Data |
| **Dependency** | gml-nil:req/GMLSchemaNillable (from this standard) |
| **Requirement** | gml-nil:req/GMLDataNillable/nilURIinXlink |
| **Recommendation** | gml-nil:req/GMLDataNillable/xlink4nil |
| **Requirement** | gml-nil:req/GMLDataNillable/documentValid |

## Special URIs in xlinks

If a GML Application Schema has the appinfo setting *nillableProperties* then any property can be marked “nil” by setting the value of xlink:href to one of the OGC URIs denoting a nil. The currently registered nil URIs are:

* http://www.opengis.net/def/nil/OGC/0/unknown
* http://www.opengis.net/def/nil/OGC/0/missing
* http://www.opengis.net/def/nil/OGC/0/withheld
* http://www.opengis.net/def/nil/OGC/0/invalid
* http://www.opengis.net/def/nil/OGC/0/template
* http://www.opengis.net/def/nil/OGC/0/AboveDetectionRange
* http://www.opengis.net/def/nil/OGC/0/BelowDetectionRange

with the definitions provided in sub-clause .

|  |
| --- |
| **Requirement** |
| gml-nil:req/GMLDataNillable/nilURIinXlink |
| Within a GML data document conforming to a GML Application Schema which contains an appinfo element with the value “nillableProperties”, a property element for which this setting is not overridden in the schema MAY have the xlink:href attribute value set to an OGC URI denoting a nil value if the value of the property is not available.  |

In addition, the values of the certain other xlink attributes MAY be set to provide more information. For example, xlink:title may have a text value describing the nil type. This plays the same function as the ‘nilReason’ pattern used in conjunction with xsi:nil=“true” as described in the GML 3.2.1 standard.

The xlink method described here provides an functional alternative to use of the xsi:nil and gml:nilReason attributes to indicate a nil property value, as described in GML 3.2. While both methods could be used within the same data document, carrying the same semantics, it would be advantageous for data providers to adopt a uniform approach within a single document. This may not be possible because of limitations on schema validity for imported components, so is included here as a ‘recommendation’ rather than ‘requirement’.

|  |
| --- |
| **Recommendation** |
| gml-nil:req/GMLDataNillable/xlink4nil |
| Within a GML data document conforming to a GML Application Schema which contains an appinfo element with the value “nillableProperties”, the xlink:href mechanism for marking nil values SHOULD be used uniformly where not disallowed by schema validation issues.  |

## XML document is schema-valid

Notwithstanding the fact that this standard provides a new mechanism to indicate nil values in XML data, each instance document containing data is still required to be schema-valid. For example, xlinks containing nil values can only appear on elements that have these as part of the type in the schema, and simple-content properties need to always contain content of the appropriate type, even if they are marked nil using the xlink mechanism. For the XML Schema type “string” an empty string is schema valid, but for all other types a dummy value must be included.

Note: for simple-content types based on xsd:float and xsd:double, the special value “NaN” (not a number) is available.

|  |
| --- |
| **Requirement** |
| gml-nil:req/GMLDataNillable/documentValid |
| A document containing data that claims to be valid according to a GML Application Schema conforming to this standard SHALL also be valid according to standard XML Schema validation rules.  |

In the following listing, the value NaN appears in the element myns:propM alongside xlink:href="http://www.opengis.net/def/nil/OGC/0/AboveDetectionRange". The nil value of the xlink:href attribute overrides any value in the element.

<myns:Class1 gml:id="\_78s56rt">
 <myns:propM xlink:href="http://www.opengis.net/def/nil/OGC/0/AboveDetectionRange"
 xlink:title="The value was above the range of detection of the sensor" uom="1"
 >NaN</myns:propM>
 <myns:propS>a text string</myns:propS>
 <myns:propB xlink:href="http://www.opengis.net/def/nil/OGC/0/inapplicable"/>
 <myns:propC xlink:href="http://www.example.org/wfs/feature/\_7912nm6"/>
</myns:Class1>

# Client application consuming GML data

## General

A conformant application consuming data conformant with this standard is required to ignore the value of any element that carries an OGC nil URI.

|  |
| --- |
| **Requirements class** |
| gml-nil:req/GMLConsumer |
| **Target type** | GML Data consumer |
| **Dependency** | gml-nil:req/GMLDataNillable (from this standard) |
| **Requirement** | gml-nil:req/GMLConsumer/ignoreNilledElementContent |

## Nil overrides element content

If an object property carries an OGC URIs denoting a nil, then any element content is ignored.

|  |
| --- |
| **Requirement** |
| gml-nil:req/GMLConsumer/ignoreNilledElementContent |
| If a data consuming application encounters an XML element representing an object property carrying an xlink:href attribute whose value is one of the OGC URIs denoting a nil, the value of the property SHALL be interpreted as void.  |

Annex A
(normative)

Abstract test suite

## A.1 General

This standard provides three conformance classes, one corresponding to each requirements class. These apply to the three standardization target types identified in 2.1. Following the requirements of the OGC Standards Policy [], any conformant implementation must pass all tests in the relevant conformance class.

## A.2 Nillable setting for information models formalized using UML

|  |
| --- |
| **Conformance class** |
| gml-nil:conf/UMLNillable |
| **Requirements** | gml-nil:req/UMLNillable |
| **Dependency** | <http://standards.iso.org/iso/19103> (Conceptual schema language) |
| **Dependency** | <http://standards.iso.org/iso/19109> (Rules for application schema) |
| **Test** | gml-nil:conf/UMLNillable/package |
| **Test** | gml-nil:conf/UMLNillable/property |
| **Test** | gml-nil:conf/UMLNillable/import |

|  |
| --- |
| **Conformance test** |
| gml-nil:conf/UMLNillable/package |
| **Requirement** | gml-nil:req/UMLNillable/package |
| **Test purpose** | If intention of the community is to allow all class attributes and association roles to be nillable in datasets corresponding to a particular application schema, unless explicitly overridden, verify that the tagged value voidableProperties is present on the UML Package corresponding to the application schema.  |
| **Test method** | Inspect the UML representation of the application schema to confirm that the package with the stereotype *«ApplicationSchema»* has a tagged value *voidableProperties* with the value *“true”.*  |
| **Test type** | Basic |

|  |
| --- |
| **Conformance test** |
| gml-nil:conf/UMLNillable/property |
| **Requirement** | gml-nil:req/UMLNillable/property |
| **Test purpose** | Verify that the appropriate stereotype is present on all class attributes and association roles to override the model default, where required.  |
| **Test method** | In an application schema with a tagged value of voidableProperties=“true”, for each class attribute or association role where it is required that a value be always provided, inspect the model to verify that it has a stereotype *«notVoidable».* In an application schema where the tagged value of voidableProperties is not present, or voidableProperties=“false”, for each class attribute or association role where it is allowed that a value need not be provided, inspect the model to verify that it has a stereotype *«voidable».* |
| **Test type** | Capability |

|  |
| --- |
| **Conformance test** |
| gml-nil:conf/UMLNillable/import |
| **Requirement** | gml-nil:req/UMLNillable/import |
| **Test purpose** | Verify that the appropriate stereotype is present on all package imports to override the model default, where required.  |
| **Test method** | In an application schema with a tagged value of voidableProperties=“true”, for each imported package containing class attributes or association roles where it is required that a value be always provided, inspect the model to verify that it has a stereotype *«notVoidable».* In an application schema where the tagged value of voidableProperties is not present, or voidableProperties=“false”, for each imported package containing class attributes and association roles where it is allowed that a value need not be provided, inspect the model to verify that it has a stereotype *«voidable».*In an application schema where the tagged value of voidableProperties is not present, voidableProperties=“false” or voidableProperties=“true”, for each imported package where the default package setting and class attributes and association overrides must be observed, inspect the model to verify that it has a stereotype *«preservevoidable».* |
| **Test type** | Capability |

# A.3 Nillable setting for GML Application Schemas

|  |
| --- |
| **Conformance class** |
| gml-nil:conf/GMLSchemaNillable |
| **Requirements** | gml-nil:req/GMLSchemaNillable |
| **Dependency** | <http://www.opengis.net/spec/gml> (Geography Markup Language) |
| **Test** | gml-nil:conf/GMLSchemaNillable/package |
| **Test** | gml-nil:conf/GMLSchemaNillable/propertyLinks |
| **Test** | gml-nil:conf/GMLSchemaNillable/propertyNillable |
| **Test** | gml-nil:conf/GMLSchemaNillable/import |

|  |
| --- |
| **Conformance test** |
| gml-nil:conf/GMLSchemaNillable/package |
| **Requirement** | gml-nil:req/GMLSchemaNillable/package |
| **Test purpose** | If intention of the community is to allow all properties to be nillable in datasets conforming to a particular GML application schema, unless explicitly overridden, verify that the schema is annotated with the value “nillableProperties”.  |
| **Test method** | Inspect the all-components schema document to determine if the element <xsd:appinfo source="http://www.opengis.net/spec/gml-nil/">nillableProperties</xsd:appinfo> is present.  |
| **Test type** | Basic |

|  |
| --- |
| **Conformance test** |
| gml-nil:conf/GMLSchemaNillable/propertyLinks |
| **Requirement** | gml-nil:req/GMLSchemaNillable/propertyLinks |
| **Test purpose** | For schemas marked nillableProperties, verify that the xlink attributes are available on all properties except if nillable=“false” |
| **Test method** | Inspect the schema document(s) to determine if each property element has the gml:AssociationAttributeGroup included in its type definition.  |
| **Test type** | Basic |

|  |
| --- |
| **Conformance test** |
| gml-nil:conf/GMLSchemaNillable/propertyNillable |
| **Requirement** | gml-nil:req/GMLSchemaNillable/propertyNillable |
| **Test purpose** | Verify that the schema default value for nillable is overridden for each property element that requires this.  |
| **Test method** | Inspect the schema document(s) to determine if each element declaration for which the community requirement for content does not match the schema default has an attribute xsd:nillable set to override the default.  |
| **Test type** | Basic |

|  |
| --- |
| **Conformance test** |
| gml-nil:conf/GMLSchemaNillable/import |
| **Requirement** | gml-nil:req/GMLSchemaNillable/import |
| **Test purpose** | Verify that the schema default value for nillable is overridden for each imported schema that requires this.  |
| **Test method** | Inspect the schema document(s) to determine if each schema import for which the community requirement for content does not match the schema default has an element<xsd:appinfo source="http://www.opengis.net/spec/gml-nil/">nillable</xsd:appinfo> or<xsd:appinfo source="http://www.opengis.net/spec/gml-nil/">notNillable</xsd:appinfo> or<xsd:appinfo source="http://www.opengis.net/spec/gml-nil/">preserveNillable</xsd:appinfo> set to override the default.  |
| **Test type** | Basic |

# A.4 Nil values in GML data

|  |
| --- |
| **Conformance class** |
| gml-nil:conf/GMLDataNillable |
| **Requirements** | gml-nil:req/GMLDataNillable |
| **Dependency** | gml-nil:conf/GMLSchemaNillable |
| **Test** | gml-nil:conf/GMLDataNillable/nilURIinXlink |
| **Test** | gml-nil:conf/GMLDataNillable/documentValid |

|  |
| --- |
| **Conformance test** |
| gml-nil:conf/GMLDataNillable/nilURIinXlink |
| **Requirement** | gml-nil:req/GMLDataNillable/nilURIinXlink |
| **Test purpose** | Verify that nil URIs in xlink:href attributes are only present where permitted.  |
| **Test method** | If an XML element representing an object property has an xlink:href attribute whose value is an OGC nil URI, verify that the all-components schema document for the data document includes an appinfo element containing “nillableProperties” and that the element declaration for the property element is not marked @nillable=“false”.  |
| **Test type** | Basic |

|  |
| --- |
| **Conformance test** |
| gml-nil:conf/GMLDataNillable/documentValid |
| **Requirement** | gml-nil:req/GMLDataNillable/documentValid |
| **Test purpose** | Verify that the data document is valid.  |
| **Test method** | Test that the data document is schema-valid against the GML Application Schema.  |
| **Test type** | Basic |

# A.5 Client application consuming GML data

|  |
| --- |
| **Conformance class** |
| gml-nil:conf/GMLConsumer |
| **Requirements** | gml-nil:req/GMLConsumer |
| **Dependency** | gml-nil:conf/GMLDataNillable |
| **Test** | gml-nil:conf/GMLConsumer/ignoreNilledElementContent |

|  |
| --- |
| **Conformance test** |
| gml-nil:conf/GMLConsumer/ignoreNilledElementContent |
| **Requirement** | gml-nil:req/GMLConsumer/ignoreNilledElementContent |
| **Test purpose** | Verify that property elements that are marked nil are interpreted as void.  |
| **Test method** | If an XML element representing an object property has an xlink:href attribute whose value is an OGC nil URI, verify that the consuming application does not use the element content.  |
| **Test type** | Capability |

Annex B
(normative)

XML Schema for simple-content types

## B.1 General

XML Schema definitions of simple-content types which carry the xlink attributes required to use the mechanism described in this standard are provided in the XML namespace

<http://www.opengis.net/gmln/1.0>

## B.2 Schema (informative)

The schema below provides type definitions for nillable simple-content types. The text below is an informative representation. The normative version of the schema is available from

<http://schemas.opengis.net/gmln/1.0>

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml/3.2"
 xmlns:gmln="http://www.opengis.net/gmln/1.0" targetNamespace="http://www.opengis.net/gmln/1.0"
 elementFormDefault="qualified">
<!-- ======================================================== -->
 <xsd:annotation>
 <xsd:appinfo source="http://www.opengis.net/spec/gml-nil/1.0">gmln.xsd</xsd:appinfo>
 <xsd:documentation>
 A set of simple-content types which carry the gml:AssociationAttribute group.
 These may be used in applications which require to use the uniform mechanism for
 marking property values 'nil' as described in http://www.opengis.net/doc/gml-nil
 These types may also be useful for other applications that require links on property values.

 Simon J D Cox
 GML-NIL is a proposed OGC Standard.
 Copyright (c) 2012 CSIRO

 </xsd:documentation>
 </xsd:annotation>
 <!-- ======================================================== -->
 <xsd:import namespace="http://www.opengis.net/gml/3.2" schemaLocation="http://schemas.opengis.net/gml/3.2.1/basicTypes.xsd"></xsd:import>

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

 <!-- ======================================================== -->
 <xsd:complexType name="boolean">
 <xsd:simpleContent>
 <xsd:extension base="xsd:boolean">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
 <xsd:complexType name="Name">
 <xsd:simpleContent>
 <xsd:extension base="xsd:Name">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
 <xsd:complexType name="NCName">
 <xsd:simpleContent>
 <xsd:extension base="xsd:NCName">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
 <xsd:complexType name="QName">
 <xsd:simpleContent>
 <xsd:extension base="xsd:QName">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
 <xsd:complexType name="float">
 <xsd:simpleContent>
 <xsd:extension base="xsd:float">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
 <xsd:complexType name="double">
 <xsd:simpleContent>
 <xsd:extension base="xsd:double">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
 <xsd:complexType name="decimal">
 <xsd:simpleContent>
 <xsd:extension base="xsd:decimal">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
 <xsd:complexType name="integer">
 <xsd:simpleContent>
 <xsd:extension base="xsd:integer">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
 <xsd:complexType name="dateTime">
 <xsd:simpleContent>
 <xsd:extension base="xsd:dateTime">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
 <xsd:complexType name="date">
 <xsd:simpleContent>
 <xsd:extension base="xsd:date">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
 <xsd:complexType name="MeasureType">
 <xsd:simpleContent>
 <xsd:extension base="gml:MeasureType">
 <xsd:attributeGroup ref="gml:AssociationAttributeGroup"/>
 </xsd:extension>
 </xsd:simpleContent>
 </xsd:complexType>

 <!-- ======================================================== -->
</xsd:schema>

Bibliography

[1] INSPIRE Generic Conceptual Model, D2.5. European Commission, 2010. <http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/D2.5_v3_3.pdf>

Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Release | Author | Paragraph modified | Description |
| 2012-07-25 | 0.1 | Simon Cox | All | Initial draft |
| 2012-07-30 | 0.2 | Simon Cox | Annex A | Completed conformance classes and tests |
| 2012-07-31 | 0.3 | Simon Cox | Clause 10, Annex A.5 | Added requirements and tests for consuming applications |
| 2012-07-31 | 0.4 | Simon Cox | Clause 9.3 | Use NaN for dummy value |
| 2012-08-04 | 0.5 | Alistair Ritchie | Clause 7.1, Clause 7.4 | Added requirements for imported packages. |
| 2012-08-08 | 0.6 | Simon Cox | Annex B | Schema for simple-content types |
| 2012-08-08 | 0.7 | Alistair Ritchie | Clause 8.1, Clause 8.5, Clause 8.6, Annex A.2, Annex A.3, | Added requirements for imported schema. Added conformance tests for clauses 7.4 and 8.5. |
| 2012-08-09 | 0.8 | Alistair Ritchie | Clause 7.4, Clause 8.5, Annex A.2, Annex A.3, | Added third import override state – preservevoid/preserveNillable – to clauses 7.4 and 8.4 and updated the conformance tests accordingly. |
| 2012-08-13 | 0.9 | Simon Cox | Clause 6.2, Annex B.2 | Provided definitions of nil URIsTypo corrected in URI |
| 2012-08-15 | 0.10 | Simon Cox | Clause 6.2, Clause 9.2 | Uniform use of xlink rather than nil  |
| 2012-09-05 | 0.11 | Alistair Ritchie | Clause 7.4, Clause 8.5 | Clarified the requirements for indirectly imported application schema. |
| 2012-09-05 | 0.12 | Simon Cox | Various Clause vi.  | Accept changes. Add material in vi. |

1. The term ‘voidable’ is used in the UML context for consistency with the precedent set in INSPIRE [1]. This is semantically equivalent to ‘nillable’ used in the XML context and generally in the text in this standard. [↑](#footnote-ref-1)