

Open Geospatial Consortium

Date: 2011-05-04

Reference number of this document: OGC 06-135r11

Identifier of this OGC[®] document: <http://www.opengis.net/doc/policy/2.0>

Category: Policy

Editor: Carl Reed

Policy Directives for Writing and Publishing OGC Standards: TC Decisions

Copyright © 2006-2011 Open Geospatial Consortium
To obtain rights of use, visit <http://www.opengeospatial.org/legal/>.

Warning

This document captures OGC Policies and Directives related to the proper writing of technical content in an OGC standards document. It is subject to change without notice. This document is an official position of the OGC membership.

Document type:	OpenGIS [®] Policy Document
Document subtype:	NA
Document stage:	Draft
Document language:	English

Contents	Page
1 Scope.....	1
2 Proper Standard templates	1
3 Copyright statements	1
3.1 OGC document copyright	1
3.2 OGC schema copyright	2
4 Correct cover page Warning descriptive text for DP, ER, and BP documents.....	2
5 Intellectual Property Wording that SHALL be included in all OGC documents that will become public.....	3
6 Use of normative terms such as SHALL, SHOULD etc.	3
6.1 Normative terms	3
6.2 Guidance on the use of normative language	4
7 Guidance on naming and referencing new OGC Standards:	5
8 Appinfo for specifying profile version	6
9 Axis order.....	7
10 Naming of Profiles, Application Schemas, and Application Profiles.....	8
11 Tightly coupled architectures.....	8
12 SOAP/WSDL Guidance for use with OGC Web Services Standards.	9
13 XML Implementation and Versioning.....	10
13.1 Summary of OGC specification version policy	10
13.2 Compatibility in XML implementations	10
13.3 XML namespaces	11
13.3.1 Standard or package name	11
13.3.2 Version designator	11
13.3.3 Pattern for namespace	11
13.3.4 Namespace example (informative)	12
13.4 Bugfix version number	12
13.5 OGC Schema repository.....	13
14 Publication Rules for OGC Schemas.....	14
14.1 Normative schema policy	15
15 SchemaLocation Values.....	15
16 XML Schema implementation for versions.....	16
17 Corrigendum and Schemas	17
18 Catalogue-Web ebRIM guidance.....	18
19 XML MIME Types	18
19.1 Normative clause for MIME-type(s) for XML grammars (initial)	18

19.2 Include MIME type specifications in XML encoding standards (details)18

20 Change Request Wording for Implementation Standards19

21 OGC HTTP URI Policy19

i. Preface

This document provides Best Practices, Policy Directives, and Standards development guidance that the OGC Members have discussed and approved over the years. These Best Practices have not been captured in other formal OGC documents other than meeting notes.

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

ii. Document terms and definitions

This document uses the normative terms (SHALL, SHOULD, MAY, NOT etc) defined in Subclause 5.3 of [OGC 06-121r3], which is based on the ISO/IEC Directives, Part 2: Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to comply with this specification.

Policy Directives for Writing and Publishing OGC Standards

1 Scope

This document captures numerous motions and recommendations from the OGC Technical Committee and Planning Committee regarding policy for writing, content, and publication of OGC standards. As new rules (policy) are identified and agreed to, this document will be updated.

2 Proper Standard templates

There are two similar OGC Document Templates: The standard OGC Standard Document Template and the OGC Standard Template for use in preparing and editing OpenGIS[®] Implementation Standards that normatively reference the OWS Common Implementation Standard. Every OpenGIS implementation and abstract specifications shall use these official OGC Document Templates.

The current version of the Template for normatively referencing OWS Common is available from the members' portal at http://portal.opengeospatial.org/files/?artifact_id=11092.

The current version of the Template for GML Profile Implementation Standards is available from the members' portal at http://portal.opengeospatial.org/files/?artifact_id=12640.

The current version of the document Template for all other OpenGIS Abstract and Implementation Standards is located at:

<http://portal.opengeospatial.org/?m=public&subtab=templates&tab=2>

3 Copyright statements

3.1 OGC document copyright

(Agreed to in January 2005) All OGC documents that are in the categories of Best Practices, candidate or adopted Implementation Standards, Implementation Standard Profiles, Implementation Standard Application Profiles, and Implementation Standard Application Schemas SHALL use the following copyright:

Copyright © <YEAR> Open Geospatial Consortium
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

This copyright SHALL be displayed on all title pages of all OGC documents except as noted below. Further, the following statement must be displayed in the footer of any OGC document: Copyright © <YEAR> OGC.

All OGC Discussion Papers that have been approved for Public Release SHALL also use this official OGC Copyright unless there are circumstances in which a shared copyright is required. In these cases, the document Author/Editor SHALL discuss the copyright requirements for that specific document with the Technical Committee Chair (TCC).

3.2 OGC schema copyright

At the June 2008 Potsdam meetings, the OGC members approved the policy that the following standard copyright wording SHALL be inserted in all OGC published schema documents:

```
<documentation>
  <XXX> is an OGC Standard
  Copyright (c) [YEAR] Open Geospatial Consortium
  To obtain additional rights of use, visit http://www.opengeospatial.org/legal/.
</documentation>
```

Note: If the OGC standard is also an ISO standard, add the clause that the document is also an “ISO standard”.

Note: Any schema published outside the OGC process using OGC schema should also include this copyright notice.

4 Correct cover page Warning descriptive text for DP, ER, and BP documents

The following Warning shall be shown on the cover page for an OGC Implementation Standard:

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

The following “Warning” SHALL be shown on the cover page of a Discussion Paper:

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

The following “Warning” SHALL be shown on the cover page of a public Engineering Report:

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements.

The following “Warning” SHALL be shown on the cover page of a Best Practice:

This document defines an OGC Best Practices on a particular technology or approach related to an OGC standard. This document is not an OGC Standard and may not be referred to as an OGC Standard. It is subject to change without notice. However, this document is an official position of the OGC membership on this particular technology topic.

5 Intellectual Property Wording that SHALL be included in all OGC documents that will become public.

The following paragraphs SHALL be included in all OGC documents that will become public. In the case of OGC Standards, Abstract Specifications, or Best Practices, this wording is included in the Preface.

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

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

6 Use of normative terms such as SHALL, SHOULD etc.

6.1 Normative terms

(Agreed to in 2005) All OGC standards documents SHALL use the following terminology as defined in Subclause 5.3 of [OGC 06-121r3], which is based on the ISO/IEC Directives, Part 2: Rules for the structure and drafting of International Standards

1. SHALL – verb form used to indicate a requirement to be strictly followed to conform to this standard, from which no deviation is permitted
2. SHOULD – verb form used to indicate desirable ability or use, without mentioning or excluding other possibilities

3. MAY – verb form used to indicate an action permissible within the limits of this standard
4. CAN – verb form used for statements of possibility
5. INFORMATIVE – a part of a document that is provided for explanation, but is not required
6. NORMATIVE – a part of a standards document that is required

Note: Other standards organizations also use the term “MUST”. For the work of the OGC, the term MUST is equivalent to the term SHALL. Also, when the term “SHALL” is used in an OGC document, it SHALL be capitalized.

6.2 Guidance on the use of normative language

Conformance language (use of MUST / SHALL / SHOULD etc.) and conformance assessment assume that the primary language of a standard will be some natural language--English, French, and so on. This is sometimes referred to as "narrative" (or just "text") as opposed to tables, figures, schemas, etc., and the portion of it that specifies requirements (by the use of SHALL/ SHOULD / MAY) is called "normative text". In OGC standards, for example, the mark of a mandatory requirement is the presence of the word SHALL in a sentence. If a document consists entirely of descriptive text, tables, figures, and equations, without any SHALL statements, that document cannot be a standard because it doesn't specify any requirements, and therefore there is nothing that an implementation can claim conformance to. Such documents are typically called "technical reports", "best practices", or other such, depending on the organization that publishes them. They "describe" something rather than "specify" something.

The above implies that a standard cannot consist only of a schema, with no normative text surrounding it. Likewise, a standard cannot consist exclusively of a data dictionary with no normative text surrounding it. (The presence of a few SHALLs inside the data dictionary is not sufficient to that effect.)

However, it is clearly useful to be able to specify a subset of the requirements by using a formal language (e.g., UML, ASN.1, XML Schema, SDL) or some ad-hoc formalism or semi-formalism (e.g., a table or a data dictionary). There is no doubt that, in many cases, the use of formal languages or tables and other editorial artifacts can make a standard much easier to read and can help prevent mistakes.

The key is that each instance of use of a formalism has to be wrapped in normative text in some way. For example:

- a fragment of XML Schema has to be preceded by a sentence in English containing a SHALL (e.g., "the element <abcd> SHALL have the type <xyz> specified as follows:", followed by a type definition in XSD);

- a normative table has to be preceded by one or more paragraphs of normative text saying that something SHALL be (or SHALL be done) according to that table, and specifying precisely what the table means and how it is to be used;
- a portion of a data dictionary (comprising one or more entries) has to be preceded by one or more paragraphs of normative text saying that something SHALL be (or SHALL be done) according to those data dictionary (entries), and specifying precisely what the dictionary entries mean and how they are to be used.

The normative text that "wraps" the formalism does two things: it specifies the meaning of the formalism (which is not needed if the formalism is well-known), and delegates its normative status to the embedded formal artifact.

This will create a complete path from the Conformance section of the standard down into the formal or semiformal standard, granting the appropriate normative status to everything that is intended to be a requirement, in whatever language or notation it is expressed (English, UML, XML Schema, ASN.1, table, data dictionary entry, etc.).

So it is very important that the formalism be specified extremely well. It is very important to avoid both ambiguity and redundancy, which can easily occur when using a semi-formal standard. In a semiformal standard, it must be very clear which portions are normative and which are informative, and among the normative portions, which ones are to be interpreted as SHALLs, which ones are to be interpreted as SHOULDs, and which ones are to be interpreted as MAYs.

If all these things are not done properly, it will be impossible to tell what it means for an implementation to conform to the standard, or it will be impossible to test conformance.

7 Guidance on naming and referencing new OGC Standards:

Whenever possible and as appropriate, when naming an OGC Standard try to work the name so that a three (3) or (4) letter acronym can be used to be a short-hand designation for that new standard. Examples are WMS for Web Map Service Implementation Standard and GML for Geography Markup Language. By way of guidance, many OGC web service interface standards begin with the letter "W" for Web and end in "S" for Service. If three or 4 letters does not work, then shorter or longer abbreviations are OK. Examples of a shorter abbreviation is OM for Observations and Measurements (although O&M is also used) and netCDF is an example of a longer abbreviation. In all cases, please be consistent in all documentation and references to the standard.

During the Boulder 2007 meetings, the OGC Planning Committee approved new guidance on the proper referencing, naming, and abbreviations for OGC standards. As per the current Technical Committee Policies and Procedures, OGC documents approved as an OGC standard SHALL be termed a standard in the title. All titles for an OGC standard SHALL reference the standard type; i.e.: interface, encoding, applications schema, or extension. For example, the correct titles for the Web Map Service and Geography Markup Language standards are:

OpenGIS® Web Map Service Interface Standard 1.3 OpenGIS® Geography Markup Language Encoding Standard

Typing and reading long titles is not always optimal. Therefore, the PC also approved a standard approach for abbreviating the names of OGC standards. For WMS and GML, the correct abbreviations are:

OGC-WMS or OGC-WMS 1.3
OGC-GML or OGC-GML 3.2.1

Please observe these naming rules in all OGC documents or Member organization marketing and product materials.

8 Appinfo for specifying profile version

At the March 2006 meetings, the TC first agreed that a GML application schema document conforming to one or more GML Profiles SHALL provide an appInfo annotation element `<gml:gmlProfileSchema>` for every profile in the root schema document `<schema>` element where the value is a schema location of the profile schema. Note that an application schema may conform to multiple profiles. Example:

```
<schema ...>
  <annotation>
    <appInfo>
      <gml:gmlProfileSchema>http://schemas
.opengis.net/gml/3.1.1/profiles/gmlSimpleFeatureProfile/1.0.0/gmlsf.xsd</gml:gmlProfileSc
hema>
      <gml:gmlProfileSchema>http://schemas
.opengis.net/gml/3.1.1/profiles/gmlPointProfile/1.0.0/gmlPointProfile.xsd</gml:gmlProfile
Schema>
    </appInfo>
  </annotation>
  ...
</schema>
```

- Define the `<gml:gmlProfileSchema>` element in the GML Schema as

```
<element name="gmlProfileSchema"
  type="anyURI"/>
```

The discussion then broadened to OGC profiles in general. The agreement is that any OGC application schema document conforming to one or more Profiles SHALL provide an appInfo annotation element for every profile in the root schema document `<schema>` element where the value is a schema location of the profile schema.

As a result, the TC also decided that OGC Application Schema documents SHALL directly (or primarily) reference the location of the complete GML Schema, not the profile XML Schema. Furthermore, an OGC Application Schema document shall also be valid when it is modified to directly reference the profile XML Schema(s) that it uses.

Important NOTE:

GML simple features profile [OGC 06-049r1] specifies using `<gmlsf:GMLProfileSchema>`, not `<gml:gmlProfileSchema>` as specified above. Since GML 3.2 has not been adopted and since GMLSF references GML 3.1.1 which does not contain such an element, GMLSF (in its GML 3.1.1 based version) has to use its own element for now.

GML 3.2 has accepted the change proposal referenced in the Huntsville motion (Bonn was the general motion, Huntsville was the GML profile motion), so `gml:gmlProfileSchema` it will be.

9 Axis order

During the closing TC Plenary for June 2005 meeting in Bonn , the members in attendance agreed that:

1. Going forward, for new standards, coordinate values SHALL be listed in the axis order as specified by the referenced coordinate reference system (CRS).
2. Going forward, when a Standards Working Group (SWG) is working on edits to an existing adopted standard, coordinate values SHALL be listed in the axis order specified by the referenced coordinate reference system (CRS).
3. Going forward, for updated and new OGC standards, CRSs SHOULD be referenced using URNs in the "ogc" URN namespace, as now specified in Clause 7 of OGC 06-121r3/OWS Common, unless a URL is used to reference a CRS.
4. Existing adopted standards will remain as is unless change requests are submitted and accepted, that seek consistency in CRS usage - such as for WMS.

However, there was still confusion as to overall guidance on how the OGC expresses axis order in our standards. Therefore, the axis order issue was again visited and discussed by the OGC Architecture Board and at the 2008 St. Louis and Potsdam meetings. The members agreed that in all cases, honesty in how axis order is expressed is the key. The following is the key additional guidance.

Any OGC documentation, encoding, payload, or service interface MUST state how the coordinate axis order is actually encoded in the coordinate strings. (CRS = Coordinate Reference System and CS = Coordinate System)

- **Case 1: CRS explicitly stated in the interface or payload:** Coordinates are expressed using the axis order as defined in the CRS (Example is a GML payload where the axis order of the CRSName and the encoding of the geometry are consistent). The CRS MAY be defined by value, by reference, or by use of a URN as defined in the "ogc" URN namespace, as now specified in Clause 7 of OGC 05-023r1/OWS Common, unless a URL is used to reference a CRS in a registry¹.

¹ There are two well defined and publicly accessible CRS registries: <http://www.epsg-registry.org> and <http://spatialreference.org/>

- **Case 1: CRS is not explicitly stated in the interface or payload:** The documentation shall explicitly state the CRS that is always used and your application/service/payload shall use the axis order as stated in the standard practice documentation. (Examples are CAP and KML), This is a corollary to Case 1.
- **Case 3: A Derived CRS or a local CS transformation is stated in the payload or interface** This case follows the OGC Abstract Specification Topic 2 guidance for defining a derived CRS and applying local transformations. Such a CS transformation could be as simple as changing the axis order.

The complete Axis Order guidance can be obtained from OGC document 08-038r4: Axis Order Policy and Guidance.

10 Naming of Profiles, Application Schemas, and Application Profiles

It is possible for approved OGC Standards to also have associated Profiles, Application Schemas, and Application Profiles. These associated Profiles, Application Schemas, and Application Profiles can be submitted to the OGC for consideration as candidates for approval as formal OGC implementation standards. For example, there have been a number of Profiles and Application Schemas developed for GML. Some of these, such as the GML Simple Features Profile, have gone through the formal approval and adoption process. Below are the definitions for Profiles, Application Schemas, and Application Profiles (from the TC Policies and Procedures, section 7.6.1).

The following are definitions for Profile and Application Schema. They are derived from ISO 19109.

- A profile is a strict subset of a Standard applicable to multiple Application Schemas. An example of a profile is the GML Simple Feature Profile.
- An application schema utilizes an Implementation Standard and adds application specific entities, e.g., feature types. An example of an application schema is LandGML or CityGML.
- An Application Profile is a profile of an OpenGIS interface standard, such as for Catalogue.

<Future Work: We still need well defined rules and examples for naming profiles and extension application schemas>

11 Tightly coupled architectures

At the January 2005 meetings, the Planning Committee unanimously agreed that in addition to web services interfaces, work on API/interfaces for tightly coupled architectures is germane and valuable in terms of the work of the consortium and fully endorses and supports such work.

Component Coupling² refers to the independence of software components. In a narrow sense, coupling refers to the way data is exchanged between components. Loose coupling is generally better than tight coupling. The loosest, and therefore preferred, type of coupling is data coupling, where data is transferred as parameters via well-defined interfaces. The tightest coupling involves components directly referencing shared variables. Tight coupling often indicates that components are not insulated from each other, and are not designed to be separate and independent. Tightly coupled components are usually complex, difficult to maintain, and monolithic. As a result there is very little flexibility regarding physical distribution of components. Two applications that communicate with each other via database management system (DBMS) updates, but which are otherwise independent of each other, would be considered loosely coupled.

12 SOAP/WSDL Guidance for use with OGC Web Services Standards.

Approved at the June 2006 Meetings. Going forward, all future revisions of existing and all new OWS (including OLS) interface standards:

1. SHOULD include an optional SOAP (messaging) binding and
2. SHOULD express that binding in WSDL.

Exceptions may only be granted through appeal to the OAB. The process will be to write a short argument as to why a SOAP binding is not required for a given implementation standard. The argument might document specific market requirements or technology constraints. For example, a given implementation environment might not support SOAP or there are bandwidth restrictions.

In order to insure that all OGC WS standards use a consistent approach, the recommendation also included the requirement for OWS Common to describe a consistent pattern for SOAP/WSDL bindings on OGC interface standards

For the foreseeable future, OGC will maintain existing GET and POST bindings. Further, there is considerable work being done in the OGC OWS Common community on defining a consistent pattern for SOAP binding. Further, please note that SOAP can be thought of as another POST binding. The membership should consider joining into that activity so that we can define a common and consistent pattern for implementing SOAP bindings for relevant OGC W*S interface standards.

Valid in 2006: The PC recommends not using the 2.0 version of WSDL as there are numerous unresolved issues with 2.0.

² Software Architecture in Practice, By Len Bass, Paul Clements, Rick Kazman., Published by Addison Wesley Professional. ISBN-10: ; ISBN-13: ; Published: Dec 30, 1997; Copyright 1998; Dimensions 6-1/4x9-1/4 ; Pages: 480; Edition: 1st.

13 XML Implementation and Versioning

13.1 Summary of OGC specification version policy

The guidelines for versions of OGC documents are provided in the OGC TC Policies and Procedures (OGC 05-020r10) clause 8.5.2. A three-level version pattern is used, where major, minor, and corrigendum (bug-fix³) versions are specified using integers, separated by periods.

NOTE major.minor is not a decimal number. Version 0.9 is not "almost 1.0" and may be succeeded by version 0.10, 0.11, 0.12 etc on the road to version 1.0.

A compatible revision does not change the meaning of any name or term used in the previous version of an OGC standard, including in all text, XML, and other documents which are part of that standard.

A major revision need not be backward compatible with previous major versions.

A minor revision is backward compatible with previous versions with the same major version designation. The initial version of a major revision has minor version=0 and bug fix version=0.

A corrigendum is used to correct errors in previous versions with the same major.minor version designation. The version attribute shall use the complete Major.Minor.Corrigendum version number initially set at 0 and shall be incremented after each bug fix (ie. version="3.2.1" for the first bug fix of the 3.2 version.)

NOTE There is no precise definition of "bug". It refers to an inadvertent error in the implementation, where it deviates from the intention expressed in the prose of the specification document, or is technically invalid for some reason, but may encompass other kinds of error. A judgement from the document editor and OGC technical committee chair is required to determine if a proposed revision may be deemed a "bug-fix" or corrigendum. Bug-fix revisions may not be backward compatible, and in general will not be since there was judged to be an error in the previous version. However, a bug-fix revision should not be used as a back door to add new features.

13.2 Compatibility in XML implementations

Where XML encoding is used, "compatibility" is evaluated at the XML instance document level. A new version of a schema SHALL be deemed to be "backward compatible" if all instance documents that were valid according to a previous version are valid according to the new version.

Applying this criterion to the versioning policy leads to the following practice for designing XML implementations:

A major revision of an XML implementation MAY use XML components from previous versions.

NOTE 1 any incompatible change to the definition of an existing component requires a major revision.

³ A corrigendum is defined as a "bug-fix" Unfortunately a definition of "bug" has proved impossible. Essentially, a bug is an unintended error in execution. Assessing this almost always involves a judgement call on the part of the editor, in consultation with the TC chair.

A minor revision of an XML implementation SHALL incorporate all XML components from previous minor revisions of the same major version, using their original namespace. A minor revision MAY add new XML components.

NOTE 2 The mechanism for constructing an XML Schema implementation for minor revisions is described in clause 16.

NOTE 3 GML 3.2 uses a different namespace than GML 3.1 for all components. Using the test described in this sub-clause GML 3.2 is not backward compatible with GML 3.1. Thus, notwithstanding the version number, GML 3.2 is a major revision compared to GML 3.1. The error in designation pre-dates the clarification of OGC Standards Best Practice described in this document.

A bug fix revision of an XML implementation SHOULD NOT add new XML attributes or elements.

13.3 XML namespaces

13.3.1 Standard or package name

An OGC standard may describe components or services in one or more packages, implemented using one or more XML namespaces. Each package intended to be implemented separately SHALL be labeled with a short label, usually of 3- or 4 letters (clause 7). The XML namespace SHALL incorporate the label of the package in which the implemented components are defined.

13.3.2 Version designator

The XML namespace can assist traceability from an XML implementation to its specification document. New components can be defined in major and minor revisions of a specification document. Hence, the XML namespace used in an implementation SHALL incorporate the major and minor version designation of the documents that contains the specification of the components implemented.

A bug fix revision corrects but does not change the meaning of, or add any new functionality to an OGC standard or implementation of that standard. The bug fix version designator SHALL NOT appear in the XML namespace.

13.3.3 Pattern for namespace

The pattern for an XML namespace used for XML implementation of an OGC standard is `http://www.opengis.net/{standard}/{major}.{minor}[/{reqclass}]` where:

- * {standard} denotes the standard or profile that defines this XML implementation
- * {major}.{minor} are the major and minor elements from the version designator of the standard or profile
- * {reqclass} denotes the requirements class implemented by this XML namespace.

{reqclass} may be omitted if the standard or profile has a single requirements class and thus is implemented in a single XML namespace.

Each of these elements should be consistent with the corresponding usage and tokens in the specification that defines this XML implementation, and with the OGC URIs that identify the relevant document and specification elements.

This XML namespace pattern is consistent with the policy for OGC URIs for related resources.

The use of only the major and minor version elements is consistent with the principle that the third number in a version designator designates bug-fix releases only, with no change of content, so users should only use the latest bug-fix version.

OGC will arrange that the XML Namespace URI will resolve to an appropriate resource representation, typically the all-components schema document.

13.3.4 Namespace example (informative)

Observations and Measurements (O&M) v1.0 describes components in two packages: Observations has the label OM, and Observation Extensions is labelled OMX (per OGC documents 07-022r1 and 08-022r1). These packages are implemented in XML as GML Application Schemas using GML 3.1.1 in namespaces that follow the pattern described in 13.3.3, viz. **<http://www.opengis.net/om/1.0>** and **<http://www.opengis.net/omx/1.0>**.

A minor revision of O&M v1.1 might add components to the Observation Extensions package. The GML 3.1.1 XML implementation would now use three namespaces corresponding to the document versions that contain the definitions of the components implemented: **<http://www.opengis.net/om/1.0>**, **<http://www.opengis.net/omx/1.0>** and **<http://www.opengis.net/omx/1.1>**.

A further minor revision O&M v1.2 might add components to the Observations package only, so the complete XML implementation would now use four namespaces: **<http://www.opengis.net/om/1.0>**, **<http://www.opengis.net/omx/1.0>**, **<http://www.opengis.net/omx/1.1>** and **<http://www.opengis.net/om/1.2>**. Each namespace contains only components newly defined in the corresponding minor revision of the specification. Thus, if a revision does not define new components in a package, there will be an apparent gap in the versioning sequence of the XML namespace.

13.4 Bugfix version number

Bug fixes SHALL NOT be reflected in the XML namespace. In XML implementations described using W3C XML Schema, the bug fix number SHALL be recorded in the `xsd:schema/@version` attribute. The version SHALL use the complete Major.Minor.Bugfix version number as the version attribute.

NOTE The XML namespace combined with the value of `schema/@version` uniquely identify the XML schema

EXAMPLE If there were a 3rd bugfix revision of the XML implementation of the schema with the namespace <http://www.opengis.net/om/1.2> the <schema> element should contain the following:

```
<xsd:schema targetNamespace="http://www.opengis.net/om/1.2" ... version="1.2.3"> ...
```

13.5 OGC Schema repository

The OGC schema repository provides a canonical location for each XML implementation of OGC standards. The location may be used in a schemaLocation attribute in XML instance documents containing elements and attributes declared in OGC XML schemas, and in XML schema documents that import an OGC XML schema. This supports online validation and allows applications to verify which implementation they are using. XML implementations for validation purposes SHALL be provided in perpetuity.

The XML Namespace provides the primary key for a specific implementation. The W3C recommendation that defines XML Schema describes several alternative strategies for resolving namespaces when validating documents. An analysis of these strategies has shown that for all processors to get the same outcome, the schema location must also be treated as canonical, as discussed in [OGC05-063r5]. Hence, in order to robustly support long-term use of OGC XML standards, the repository must provide a persistent location for every implementation with a different namespace, i.e. every major and minor version.

In order to support transparency the repository must also provide access to every bug fix version. However, since bug fix revisions use the same namespace within a minor version, in order to discourage the use of superseded versions, bug fix versions prior to the most recent one ought not to be available for online validation. These requirements are met by having all bug fix versions available as compressed archives, with the most recent also available for online validation at the canonical location for its namespace.

The structure of the OGC schema repository ought to reflect the same considerations used in designing XML namespaces: i.e. package label, major.minor version. There ought to be a directory for each package, within which there ought to be a directory for each major.minor version. Compressed archives of bug fix releases SHALL reside in the schema repository as siblings to the directories containing the minor versions.

NOTE 1 There is no distinction made for standards and packages that are profiles or extensions of existing standards. A separate directory should be used for all cases, as a sibling of all the other independently versioned standards or packages. The rationale for this is that every standard is likely to have multiple dependencies, so while some may seem to be more closely related to existing standards, complete lineage cannot be indicated in the repository structure, and furthermore may change in future revisions.

The structure of the repository can be described by example. Following these rules, the GML schemas issued by OGC prior to 2008-09-11 would be re-organized into the following structure:

```
http://schemas.opengis.net/gml
- 1.0 // unchanging 1.0.0 versions
- 2.0 // unchanging 2.0.0 versions
- 2.1 // contains 2.1.2 bugfix of the 2.1 schema
- 2.1.1.zip
```

- 2.1.2.zip
- 3.0 // contains 3.0.1 bugfix of the 3.0 schema
- 3.0.0.zip
- 3.0.1.zip
- 3.1 // contains 3.1.1 bugfix of the 3.1 schema
- 3.1.0.zip
- 3.1.1.zip
- 3.2 // contains 3.2.1 bugfix of the 3.2 schema
- 3.2.1.zip

NOTE 2 In practice, since the repository existing prior to 2008-09-11 contains earlier bugfix versions unzipped, and it is known that some implementations make reference to these locations, it is necessary that the current directories remain available.

In the case of GML the "all-components" schema document, as defined in [clause 14](#) of this document, is called gml.xsd. Thus, the all-components schema document for the current bugfix implementation of the namespace **<http://www.opengis.net/gml/3.2>** would be located at **<http://schemas.opengis.net/gml/3.2/gml.xsd>**.

The scenario for the XML implementation of Observations and Measurements described in the namespace example (13.3.4) leads to the following structure:

```
http://schemas.opengis.net/om
- 1.0
- 1.0.0.zip
- 1.2
- 1.2.0.zip
...
http://schemas.opengis.net/omx
- 1.0
- 1.0.0.zip
- 1.1
- 1.1.0.zip
```

In the case of O&M the "all-components" schema document (clause 14) is called om.xsd. Thus, the all-components schema document for the current bugfix implementation of the namespace **<http://www.opengis.net/om/1.2>** would be located at **<http://schemas.opengis.net/om/1.2/om.xsd>**

14 Publication Rules for OGC Schemas

For all OGC XML schemas, when writing a candidate standard and when publishing that standard, one XML schema document SHALL be provided that includes, either directly or indirectly, all of the components defined and declared in the namespace for that XML schema (the "all-components" document). This "all-components" XML schema document for each XML schema SHALL be clearly identified in the document file name.

EXAMPLES: The "all-components" XML schema document for GML is named gml.xsd, and for OWS Common is named owsAll.xsd.

Whenever an XML schema published by the OGC has dependencies on another OGC XML schema, a single XML schema document that provides access to the full namespace

SHALL be referenced by the dependent XML schema. The "all-components" XML schema document SHALL be referenced by an <import> statement in each XML schema document that uses any component of that referenced XML schema.

Where a schema for a single namespace is broken into several schema documents, each document other than the all-components document SHALL <include> the all-components schema document. NOTE – this ensures that any reference to any schema document in the OGC repository will automatically get all components for the namespace, according to OGC’s schema publication policy⁴. Also, be aware that when a new component/extension is added, you will need a new all components schema in a new namespace

14.1 Normative schema policy

OGC Standards Documents do not contain normative schemas but instead reference them (such as in the OGC schema repository. Therefore, schema references are normative and reference is by namespace and location. All examples of schema use in an OGC standards document are informative and should be labeled as examples. In the case of an inconsistency between the OGC standards document and the normative schema in the OGC repository, the schema in the repository takes precedence.

15 SchemaLocation Values

Schema locations appear within XML implementations in three places:

(i) the @xs:schemaLocation attribute on <xs:include> elements within XML schema documents

- this indicates the location of a document containing definitions of schema components whose xs:schema/@targetNamespace value is the same as the including document. I.e. this controls inclusion of components in the same namespace, implicitly under the same governance arrangements.

(ii) the @xs:schemaLocation attribute on <xs:import> elements within XML schema documents

- this indicates the location of a document containing definitions of schema components whose xs:schema/@targetNamespace value differs from the including document. I.e. this controls import of components from another namespace, generally under separate governance arrangements.

(iii) the @xsi:schemaLocation attribute on elements within XML instance documents

- this indicates the location of a document containing definitions of components for the associated namespace.

Maximum portability of XML documents (both schemas and instances) is enabled by applying the following pattern to schemaLocation values:

⁴ This clarification paragraph was added based on a motion at the March 2011 Bonn TC meetings.

Case (i) – inclusion of components in the same namespace – relative path to local address (allows a schema factored into multiple documents to be packaged easily)

EXAMPLE – Within the Sampling features schema, samplingBase.xsd includes surveyProcedure.xsd using a relative path to a location in the same directory

```
<include schemaLocation="./surveyProcedure.xsd"/>
```

Case (ii) – import of components from an independently governed namespace – absolute path to canonical location of all-components schema document (see clause 14)

EXAMPLE – Within the Sampling features schema, samplingBase.xsd imports the Observations schema by reference to the canonical location of the all-components document om.xsd

```
<import
  namespace="http://www.opengis.net/om/1.0"
  schemaLocation="http://schemas.opengis.net/om/1.0/om.xsd"/>
```

Case (iii) – pointer to source of definitions - absolute path to canonical location of all-components schema document

EXAMPLE – Within an instance document describing a specimen, the namespace is bound to the all-components document for Sampling Features sampling.xsd

```
<sa:Specimen gml:id="pr1_s2"
  xmlns:sa="http://www.opengis.net/sampling/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:gml="http://www.opengis.net/gml"
  xsi:schemaLocation="http://www.opengis.net/sampling/1.0
  http://schemas.opengis.net/sampling/1.0/sampling.xsd">
...
</sa:Specimen>
```

16 XML Schema implementation for versions

An XML Schema that implements a minor revision of a standard SHALL incorporate all prior revisions of the same major version by importing the all-components schema document for each previous revision. An XML Schema that implements a minor revision of a standard SHALL NOT re-implement components from any prior revisions of the same major version in a new namespace.

NOTE This is required in order to satisfy the compatibility requirements described in Clause 13.2, the namespace rules described in Clause 13.3, and the schemaLocation rules described in Clause 15.

NOTE The various versions of GML through GML 3.2 do not follow this rule, which was established after GML 3.1 was published.

EXAMPLE Consider an OGC standard with the abbreviation WXX. For the initial major version, the all-components schema document wxx.xsd would start as follows:

```
<xsd:schema targetNamespace="http://www.opengis.net/wxx/1.0" ... version="1.0.0">
```

```
...
```

The first bugfix for this version would start:

```
<xsd:schema targetNamespace="http://www.opengis.net/wxx/1.0" ... version="1.0.1">
...

```

The first minor revision of this version would start:

```
<xsd:schema targetNamespace="http://www.opengis.net/wxx/1.1" ... version="1.1.0">
    <xsd:import namespace="http://www.opengis.net/wxx/1.0"
schemaLocation="http://schemas.opengis.net/wxx/1.0/wxx.xsd">

<!-- new type definitions and global element declarations -->
...

```

The second minor revision of this version would start:

```
<xsd:schema targetNamespace="http://www.opengis.net/wxx/1.2" ... version="1.2.0">
    <xsd:import namespace="http://www.opengis.net/wxx/1.0"
schemaLocation="http://schemas.opengis.net/wxx/1.0/wxx.xsd">

    <xsd:import namespace="http://www.opengis.net/wxx/1.1"
schemaLocation="http://schemas.opengis.net/wxx/1.1/wxx.xsd">

<!-- new type definitions and global element declarations -->
...

```

The first bugfix for the second minor revision of this version would start:

```
<xsd:schema targetNamespace="http://www.opengis.net/wxx/1.2" ... version="1.2.1">
    <xsd:import namespace="http://www.opengis.net/wxx/1.0"
schemaLocation="http://schemas.opengis.net/wxx/1.0/wxx.xsd">

    <xsd:import namespace="http://www.opengis.net/wxx/1.1"
schemaLocation="http://schemas.opengis.net/wxx/1.1/wxx.xsd">

<!-- new type definitions and global element declarations -->
...

```

17 Corrigendum and Schemas

(From December 2007 TC/PC meetings in Stresa Italy)

As per section 9.10 of the OGC Technical Committee Policies and Procedures, a corrigendum is restricted to a **bug fix** to schema(s) and supporting standards document. There is a requirement for an immediate fix. When the corrigendum is published, there SHALL be No namespace change. The corrigendum *shall* be accompanied with a ReadMe file that describes the errors that were corrected. The information for the ReadMe file can be extracted from the Corrigendum document. Also, the version attribute on the schema element SHALL be used to signify that there has been a change. There will be no change to the version number of the actual documents.

- Use x.x.x at the third level in the version attribute to signify a change using a corrigenda
 - o There shall be no change in the location of the schemas.
 - o A corrigendum shall replace the old schemas in their current location.

For example, in the code below one could change version="1.0.0" to version="1.0.1":

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  version="1.0.0">
```

Please remember that the XML validator ignores the version attribute. Therefore, it is not an enforceable constraint.

From an OGC Architecture motion August 13, 2009, the published schema takes authority precedence over all other related documents.

18 Catalogue-Web ebRIM guidance

(September 2009 PC) Initially in 2006, the OGC Planning Committee approved a motion stating that OASIS ebRIM is the preferred application profile metadata for all future CSW application profiles. Due to some ambiguity in the guidance, in September 2009, the PC modified and approved the following:

- That OASIS ebRIM be adopted as the preferred (cataloguing) meta-model for future OGC CS-W Catalogue Application Profiles
- With regard to the ebRIM meta-model, an Application Profiles as understood in the CSW context is defined to mean an extension package of ebRIM
- Interoperability issues between ISO Metadata AP and ebRIM AP need to be harmonized.
- The existing ISO Metadata Application Profile is not deprecated.

19 XML MIME Types

19.1 Normative clause for MIME-type(s) for XML grammars (initial)

2010. OGC Frascati Technical Committee Meetings. Any OGC standard that defines an XML grammar shall include a normative clause indicating the MIME-type(s) to be used, and sub-types if applicable.

19.2 Include MIME type specifications in XML encoding standards (details)

Compatibility with content-negotiation mechanisms provided by the http protocol requires a MIME-type to be registered for each distinct encoding. IANA registration accepts MIME-type description included in standards published by other organizations provided they generally follow the IANA template.

Therefore, each OGC encoding standard shall

- Add IETF RFC 4288 'Media Type Specifications and Registration Procedures' as a normative reference.
- Include a MIME-type specification shall be included as a separate clause in every OGC encoding standard⁵. The clause containing the MIME-type specification shall contain the headings taken from the IANA Application for Media Type application <http://www.iana.org/cgi-bin/mediatypes.pl>

For XML encodings, an OGC MIME type should follow the pattern
application/xxxxxxx+xml

20 Change Request Wording for Implementation Standards

The following wording shall be incorporated in the preface of any Member approved OGC Implementation Standard:

Suggested additions, changes and comments on this standard are welcome and encouraged. Such suggestions may be submitted using the online change request form on OGC web site: http://portal.opengeospatial.org/public_ogc/change_request.php

21 OGC HTTP URI Policy

At the June 2010 meetings, the OGC Members approved a change related to the use of URIs in OGC standards documents. There were two parts to the motion:

1. OGC Technical and Planning Committees directs the OGC-Naming Authority that all new OGC identifiers issued for persistent public OGC resources shall be http URIs, instead of URNs;
2. New standards and new major versions of existing standards shall use http URIs for persistent public OGC resources to replace OGC URN identifiers defined in previous standards and versions, unless the OGC-NA approves an exception.

There is an associated OGC White Paper⁶ that provides the research, justification for, and examples of this policy. An example of the structure of an OGC http URI for a CRS is:

<http://www.opengis.net/def/crs/EPSG/0/4326>

Notice that the structure is very similar to that of an OGC URN string. The corresponding OGC URN would be:

urn:ogc:def:crs:EPSG::4326

⁵ Note: This separate clause is included in the new OGC document template.

⁶ Add http reference when white paper is published. OGC Identifiers – the case for http URIs [OGC 10-124]

Annex A: Revision History

Date	Internal version	Editor	Sections modified	Description
15 July 2006	0.1.0 Draft	Carl Reed	N/A	Initialised Draft Document.
19 Sept 2006	0.1.1 Draft	Arliss Whiteside	N/A	Many edits and comments
August 15, 2007	1.2	Carl Reed	Various. Added new clauses	Updated with guidance on standard wording for DP and BP, IP wording for Forward, changed “specification” to “standard”
11 December 2008	2.0	Carl Reed	11, 14, 15	Added/Updated versioning, namespace and repository structure from Change Request 08-148r2
8 January 2009	2.0.1	Simon Cox	3, 11, 16, 17	Clarified issues identified by Arliss Whiteside during approval vote. Inserted new clause 16 explaining XML Schema pattern for minor revisions in response to comment by Andreas Doemming.
20 January 2009	2.0.2	Simon Cox	4, 5, 13, 14, 15	Moved clauses 14, 15 to 4, 5 so that clauses are thematically grouped. Added note discussing definition of bug and bug-fix. Added notes to clarify treatment of profiles and 'extensions' in XML implementations. Capitalized all normative terms. Removed normative language from non-normative clauses.
30 March 2009	2.0.3	Carl Reed	Cover, 13, others	Incorporate edits for all approved changes and create new version of document for posting
June 2009	2.0.3	Carl Reed	Various	Minor changes prior to publication
April 2010	2.0.3	Carl Reed	New sections	Capture several new directives
April 2011		Carl Reed	Various	Updates to sections 7, 13 and 14. New section for MIME types