Open Geospatial Consortium Inc.

Date: 2007-04-30

Reference number of this OpenGIS® document: **OGC 07-001r3**

Version: 0.5.0

Category: OpenGIS® Discussion Paper

Editor: Simon Cox

Requirements for some specific simple solid, plane and line geometry types

Copyright notice

Warning

This document is not an OGC Standard. This 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.

Document type: OpenGIS® Discussion Paper
Document subtype: Not applicable
Document stage: Draft
Document language: English

Contents

Figures

Code Listings

i. Preface

This Document was produced as part of the OGC's Sensor Web Enablement (SWE) activity.

Suggested additions, changes, and comments on this report are welcome and encouraged. Such suggestions may be submitted by OGC portal message, email message, or by making suggested changes in an edited copy of this document.

The changes made in this document version, relative to the previous version, are tracked by Microsoft Word, and can be viewed if desired. If you choose to submit suggested changes by editing this document, please first accept all the current changes, and then make your suggested changes with change tracking on.

ii. Submitting organizations

The following organizations submitted this document to the Open Geospatial Consortium Inc.

Commonwealth Scientific and Industrial Research Organisation (Australia) (CSIRO).

iii. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Contact	Company	Address
Simon Cox (Editor)	CSIRO Exploration and Mining	ARRC PO Box 1130 Bentley WA 6102 Australia

iv. Revision history

Date	Release	Editor	Primary clauses modified	Description
2006-12-13	0.1.0	Simon Cox	Initial version	Extract relevant sections from 05-087r4 O&M and re-organise; rename SimpleSolid -> ConvexPolyhedron per recommendation from John Herring
2007-04-04	0.2.0	Simon Cox	Clause 1, 7	In Scope clause, added reference to BREP vs CSG; Remove clause 7 (TriSurface) – no significant advantage over gml:TriangulatedSurface
2007-04-13	0.3.0	Simon Cox	Clause 6	Added Schematron constraints in convexPolyhedron.xsd
2007-04-26	0.4.0	Paul Scarponci ni	All	Reconfigured document to be requirements document
2007-04-30	0.5.0	Simon Cox	Annex C	Added instance example of gml:Solid for contrast with proposed schema.

v. Changes to the OpenGIS® Specification

The OpenGIS® Abstract Specification should be updated to reflect the new geometry types being proposed so that changes implemented in GML will be consistent with other possible implementations, such as SQL.

vi. Future work

These requirements should be coordinated with similar requirements being developed by the 3DIM WG. The work there is based on IFC classes. Both sets of requirements should be resolved in the Geometry WG.

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.

This report replaces parts of Annex C in the the OGC Best Practices Paper 05-087r4 *Observations and Measurements*.

This specification was developed under the OWS 4 initiative as part of the Sensor Web Enablement thread. It provides an encoding for some specific geometries that are useful in the description of sampling feature types, as well as for other purposes.

All parts of the document are normative except for ANNEX A – XML implementation examples.

Introduction

This specification describes requirements for specific geometry types, including some simple solids, and planes and lines defined using an implicit parameterization. The geometries are specializations of the GM_Solid, GM_Surface and GM_Curve classes (ISO 19107), but provide a closer match to the parameterization conventionally used in certain application domains than do the representations described in ISO 19107 and currently implemented in GML. These application domains include, in particular, numerical simulations using finite elements and finite differences.

The requirement for these emerged from (a) data transfer between coupled earth simulation systems in the mineral exploration sector; (b) descriptions of sampling artefact feature types required for Sensor Web Enablement. The requirements were originally included as part of the Observations and Measurements specification (OGC 05-087r4) but is considered to be of more general interest, so has been refactored into a separate document.

Because of ongoing work in the 3DIM WG on Extended Geometries, these requirements are submitted to the Geometry WG for resolution and ultimately extensions to the ISO 19107 based Abstract Specification which is the basis for existing GML geometries. It should be coordinated with implementations other than GML (e.g., Simple Features for SQL) as well as other organizations such as TC211 and SQL/MM who have already harmonized with the existing Abstract Specification.

Requirements for some simple solids, planes and lines

1 Scope

ISO 19107 describes the spatial schema that is implemented in GML. It provides a comprehensive general model for spatial objects, including types for points, curves, surfaces and solids. The models are rigorous and robust, but when used directly as the basis for implementation they involve a degree of elaboration that is higher than commonly used in software applications, particularly for some simple 3-D cases.

For example, ISO 19107 uses a Boundary Representation (BREP) model for solids, where the volume of interest is bounded by one outer and (optionally) several inner shells, defined by oriented-surfaces; where a surface is composed of surface patches; where a surface patch is bounded by one outer and (optionally) several inner oriented-rings; where a ring is composed of curve segments; where a curve segment is defined by an ordered set of points; where a point has a position. The representation of a solid included in the explicit implementation of the geometry schema within GML reflects this complexity.

In contrast, many conventional engineering implementations, including numerical simulations, construct discretized models using multiples of simple primitives similar to Constructive Solid Geometry (CSG). In mechanics simulations using finite elements these are typically hexahedron, wedge, pyramid and tetrahedron. The description of each of these is simply by an ordered sequence of points. The semantics is ultimately the same as the BREP, but conventions are applied for each type which allow the information to be compressed into the position and order of the vertex points. Conversely, transformation of the sequence-of-points into the rigorous solid model from ISO 19107 is algorithmically possible, and hence so is their representation using the standard GML encoding, but this is of questionable value in the context of likely data transfer applications.

This specification describes of the requirements for the simplified models for these solid geometry types.

The second case considered in this specification concerns description of planar surfaces and simple lines using a functional notation based on vector normals and directions, as a complementary view compared to the standard model.

These components have application in a variety of application domains. For this reasons it is appropriate that a standard model and encoding be provided to save specific application domains from defining potentially incompatible variations. For example, the AIA specifies constructive solid geometries (CSG) in their Industry Foundation Classes

(IFC). Since the 3DIM WG is working towards adopting IFC for CAD/GIS/BIM integration, then the implications of this decision need to be assed along with the requirements contained herein.

2 Conformance

2.1 Overview

This Specification uses the Unified Modeling Language (UML) to present the conceptual schemas. This schema defines conceptual classes that (i) may be considered to comprise a cross-domain application schema, or (ii) may be used in application schemas, profiles and implementation specifications.

3 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

ISO 19101:2002, *Geographic Information--ReferenceModel*

ISO/TS 19103:2005, *Geographic Information — Conceptual schema language*

ISO 19107:2003, *Geographic Information — Spatial schema*

ISO 19109:2006, *Geographic Information — Rules for application schemas*

ISO 19118:2005, *Geographic Information — Encoding*

ISO DIS 19136:2006, *Geographic Information — Geography Markup Language*

ISO/IEC 19757-3:2006, *Information technology -- Document Schema Definition Language (DSDL) -- Part 3: Rule-based validation -- Schematron*

IETF RFC 2396, *Uniform Resource Identifiers (URI): Generic Syntax. (August 1998)*

OpenGIS® Implementation Specification *Geography Markup Language, version 3.2. OGC Document 05-108r1* http://www.opengeospatial.org/

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

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

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

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

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

4 Terms and definitions

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

4.1
application schema
conceptual schema for data required by one or more applications

[ISO 19101]

4.2
GML application schema
application schema written in XML Schema according to the rules specified in ISO 19136

[ISO 19136]

4.3
namespace <XML>
collection of names, identified by a URI reference, which are used in XML documents as element names and attribute names [W3C XML Namespaces]

4.4
schema
formal description of a model

[ISO 19101]

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

4.5
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.

4.6
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.

5 Conventions

5.1 Symbols (and abbreviated terms)

BREP	Boundary Representation for solid geometry
CSG	Constructive Solid Geometry
GML	Geography Markup Language
ISO	International Organization for Standardization
OGC	Open Geospatial Consortium
UML	Unified Modeling Language
WXS	W3C XML Schema Definition Language
XML	Extensible Markup Language
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional

5.2 UML notation

Most diagrams that appear in this specification are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of the OGC Web Services Common Implementation Specification [OGC 04-016r2].

Many of the models refer to classes from various models in the ISO 19100 series of international standards. In this document these components have been imported from the ISO Harmonized Model as of 2006-06-14.

The UML is conformant with the profile described in ISO 19103 and ISO 19136 (GML) Annex E. Use of this restricted idiom supports direct transformation into a GML Application Schema.

The prose explanation of the model uses the term “property” to refer to both class attributes and association roles. This is consistent with the General Feature Model described in ISO 19109. In the context of properties, the term “value” refers to either a literal (for attributes whose type is simple), or to an instance of the class providing the type of the attribute or target of the association. Within the explanation, the property names are sometimes used as natural language words where this assists in constructing a readable text.

5.3 Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 04-016r2].

6 Simple solids

6.1 Model

Figure 1 shows the schema for some simple solids. Classes in Figure 1 without a GM_ prefix are presented for consideration as possible extensions to the ISO 19107 based Abstract Specification.

ConvexPolyhedron is the root of a hierarchy of simple solid primitives. A ConvexPolyhedron is the solid contained by the convex hull defined by a set of four or more vertices, each of which is a GM_Point. Any point which falls in the interior of the convex hull is not a vertex of the hull bounding the solid. Four specializations are defined: tetrahedron, pyramid, wedge and hexahedron, in which the number of vertices is limited to four, five, six or eight, respectively, with the vertices connected in rings to form either triangular or quadrilateral surface patches, in the arrangement required for the respective solid. Finally, MultiConvexPolyhedron is a geometric aggregate whose members are constrained to be simple primitives.

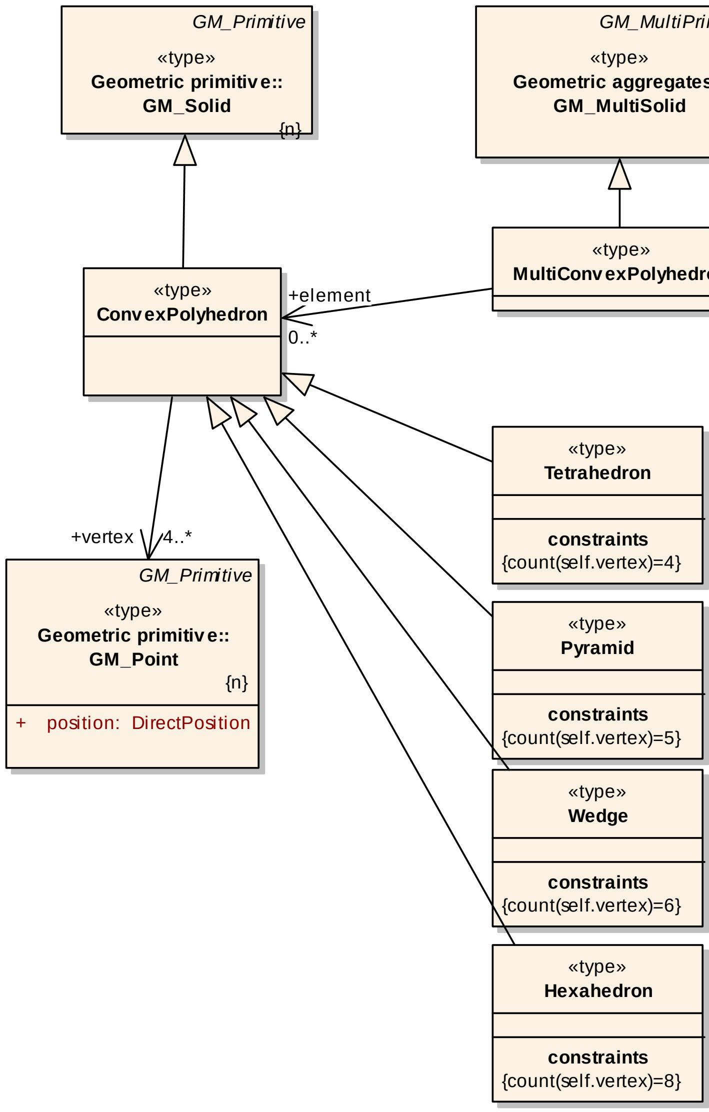

Figure 1. Simple solids schema, showing relationship with ISO 19107 types

Most algorithms that build the convex hull from a set of points produce solids with a triangulated surface. In contrast, the applications that gave rise to this requirements document use a discretization of space into cells with both three- and four-cornered faces, with the latter possibly not planar, so topology is maintained using an explicit method.

In many applications, this is through listing vertices in a fixed order which defines the connectivity and orientation of the edges and faces, and hence the interior of the solid. The most common ordering is shown for the four specializations in Figure 2.

NOTE: This is consistent with a BREP model. Enforcing a specific topology is in contrast with simply computing the polyhedron topology using the convex hull, and is typically used to ensure alignment of mesh elements in a 3-D discretization with known boundaries and interfaces in the world being modelled. Having a convention for topology (i) provides a unique, repeatable method for building mesh elements (ii) allows faces to be identified for class membership schemes, and (iii) supports applications ability to convey information (fluxes, forces) within a discretized space, represented internally as transmission across specific cell faces and vertices.

<table>
<tr><th colspan="2">Hexahedron</th><th colspan="2">Wedge</th><th colspan="2">Pyramid</th><th colspan="2">Tetrahedron</th></tr>
<tr><td colspan="2">7____6
/ | / |
4____5 |
| 3 _ _| 2
| / | /
0____1</td><td colspan="2">5
/ | \
/ 4 \
3 / -\- 2
|/   \ |
0____1</td><td colspan="2">4
//\ \
3 / -\- 2
|/   \ |
0____1</td><td colspan="2">3
/ | \
2- -| - -1
\ | /
0</td></tr>
<tr><td>face</td><td>vertices</td><td>face</td><td>vertices</td><td>face</td><td>vertices</td><td>Face</td><td>vertices</td></tr>
<tr><td>0
1
2
3
4
5</td><td>0 1 5 4
1 2 6 5
2 3 7 6
3 0 4 7
0 3 2 1
4 5 6 7</td><td>0
1
2
3
4</td><td>0 1 4
1 2 5 4
2 3 5
3 0 4 5
0 3 2 1</td><td>0
1
2
3
4</td><td>0 1 4
1 2 4
2 3 4
3 0 4
0 3 2 1</td><td>0
1
2
3</td><td>0 1 3
1 2 3
2 0 3
0 2 1</td></tr>
</table>

Figure 2. (Informative) Conventional vertex order in specialized polyhedrons,

NOTE: Vertex order is with respect to a right-handed coordinate system. Face order is indicated in the lower tabulation.

Users should verify that similar conventions are used when exchanging data between applications that use vertex order to specify connectivity. The orderings shown here are based on the local (polyhedral element) coordinate system. Transformation to a global system is then application specific.

NOTE: For historical reasons, most geographic (un-projected) coordinate reference systems specify that the horizontal coordinate elements appear in latitude-longitude order. When combined with a elevation positive vertical axis, this results in a left-handed coordinate system, which contrasts with most engineering coordinate reference systems.

7 Planes and lines

7.1 Model

Figure 3 shows the proposed schema for the description of a plane or line in terms of its vector-normal or direction, respectively. This parameterization is frequently used when (parts of) the boundary of a line or plane are unknown. Classes in Figure 3 without a GM_ prefix are presented for consideration as possible extensions to the ISO 19107 based Abstract Specification.

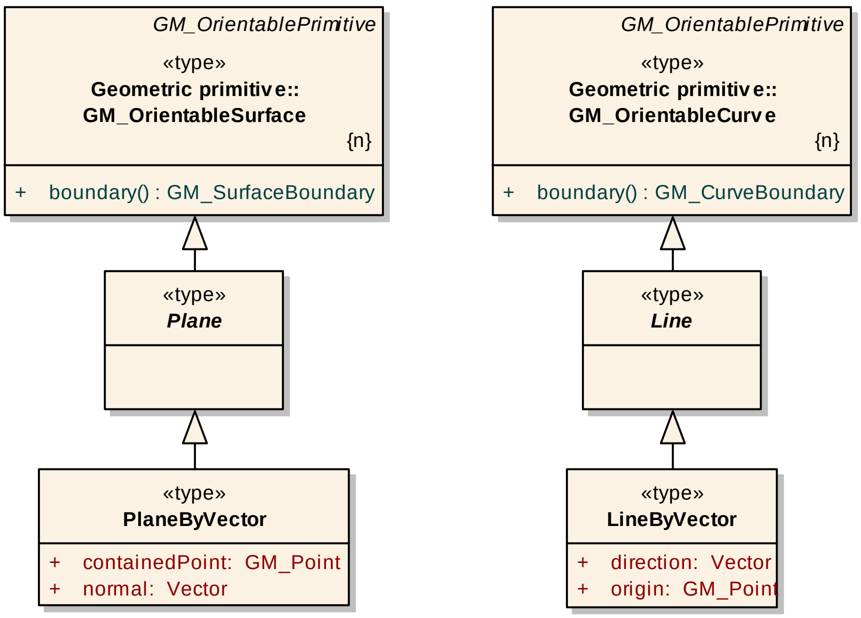

Figure 3. Proposed schema for planes and lines, showing relationship with ISO 19107 types

ANNEX A Possible GML Implementation for Simple Solids (informative)

The XML Schema representation follows the UML-GML encoding rules described in Annex E of GML 3.2 (ISO DIS 19136). Schematron rules are added to constrain the vertex count in elements representing the specialized classes.

In this XML implementation, vertex order is defined by XML element order. The W3C XML specification requires that document-order is significant, so this strategy is robust in conformant processors. Note also that the GML geometry implementation makes use of document order to indicate geometric topology in a number of places.

NOTE: In XML processing environments that do not preserve document element order this strategy will fail.

Listing 1. convexPolyhedron.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
xmlns:geox="http://www.opengis.net/geomExt/0.0" targetNamespace="http://www.opengis.net/geomExt/0.0"
elementFormDefault="qualified" attributeFormDefault="unqualified"
    xmlns:sch="http://purl.oclc.org/dsdl/schematron" version="0.0.0">
  <annotation>
    <appinfo>
      <sch:title>Schematron validation</sch:title>
      <sch:ns prefix="gml" uri="http://www.opengis.net/gml"/>
      <sch:ns prefix="geox" uri="http://www.opengis.net/geomExt/0.0"/>
      <sch:ns prefix="xlink" uri="http://www.w3.org/1999/xlink"/>
      <sch:ns prefix="xs" uri="http://www.w3.org/2001/XMLSchema"/>
      <sch:ns prefix="xsi" uri="http://www.w3.org/2001/XMLSchema-instance"/>
    </appinfo>
    <documentation> convexPolyhedron.xsd

A GML conformant schema for specialised geometries
simple solids

Copyright (c) 2007 Open Geospatial Consortium - see http://www.opengeospatial.org/about/?page=ipr</documentation>
  </annotation>
  <!-- ====================================================================== -->
  <!-- bring in other schemas -->
  <import namespace="http://www.opengis.net/gml"
schemaLocation="http://testschemas.opengis.net/gml/3.2.0/gml/gml.xsd"/>
  <!-- ====================================================================== -->
  <!-- ====================================================================== -->
  <!-- =========================================================== -->
  <!-- === Solids === -->
  <!-- =========================================================== -->
  <complexType name="ConvexPolyhedronType">
    <complexContent>
      <extension base="gml:AbstractSolidType">
        <sequence>
          <element name="vertex" type="gml:PointPropertyType" minOccurs="4" maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ............. -->
```

```
  <element name="ConvexPolyhedron" type="geox:ConvexPolyhedronType" substitutionGroup="gml:AbstractSolid">
    <annotation>
      <documentation>Head of substitution group for primitive Solids with simple descriptions.</documentation>
    </annotation>
  </element>
  <!-- .................. -->
  <complexType name="ConvexPolyhedronPropertyType">
    <sequence minOccurs="0">
      <element ref="geox:ConvexPolyhedron"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- =============================================================== -->
  <!-- =============================================================== -->
  <complexType name="MultiConvexPolyhedronType">
    <annotation>
      <documentation>AbstractGeometricAggregateType is used as root class instead of MultiSolid to avoid derivation-
by-restriction to override the inherited "solidMember" property</documentation>
    </annotation>
    <complexContent>
      <extension base="gml:AbstractGeometricAggregateType">
        <sequence>
          <element name="element" type="geox:ConvexPolyhedronPropertyType" minOccurs="0"
maxOccurs="unbounded"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
  <!-- ............. -->
  <element name="MultiConvexPolyhedron" type="geox:MultiConvexPolyhedronType"
substitutionGroup="gml:AbstractGeometricAggregate">
    <annotation>
      <documentation>Head of substitution group for multiSolids with simple descriptions.</documentation>
    </annotation>
  </element>
  <!-- .................. -->
  <complexType name="MultiConvexPolyhedronPropertyType">
    <sequence minOccurs="0">
      <element ref="geox:MultiConvexPolyhedron"/>
    </sequence>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
  </complexType>
  <!-- =============================================================== -->
  <complexType name="HexahedronType">
    <annotation>
      <documentation>The "Hexahedron" element is a simple solid with eight vertices and six sides.
Relative to a right-handed coordinate system the conventional vertex sequence is
   7____6
  / |     / |
4____5 |
| 3 __  |  2
| /       | /
0____1

For cases where the face order is used implicitly, then the faces are usually defined with the following vertices (in the local
coordinate system)

0 - 0 1 5 4
1 - 1 2 6 5
2 - 2 3 7 6
3 - 3 0 4 7
4 - 0 3 2 1
5 - 4 5 6 7</documentation>
    </annotation>
    <complexContent>
      <extension base="geox:ConvexPolyhedronType"/>
    </complexContent>
  </complexType>
  <!-- .................. -->
  <element name="Hexahedron" type="geox:HexahedronType" substitutionGroup="geox:ConvexPolyhedron">
    <annotation>
```



```
        <appinfo>
          <sch:pattern id="hex" name="Hexahedron has 8 vertices">
            <sch:rule context="//geox:Hexahedron">
              <sch:assert test="count(geox:vertex) = 8">Hexahedron must have 8 vertices</sch:assert>
            </sch:rule>
          </sch:pattern>
        </appinfo>
        <documentation>The "Hexahedron" element is a simple solid with eight vertices and six sides.</documentation>
      </annotation>
    </element>
    <!-- ============================================================ -->
    <complexType name="WedgeType">
      <annotation>
        <documentation>The "Wedge" element is a simple solid with six vertices and five sides.
Relative to a right-handed coordinate system the conventional vertex sequence is
      5
    / | \
   / 4  \
 3 / -\- 2
  |/     \ |
 0____1

For cases where the face order is used implicitly, then the faces are usually defined with the following vertices (in the local coordinate system)

0 - 0 1 4
1 - 1 2 5 4
2 - 2 3 5
3 - 3 0 4 5
4 - 0 3 2 1</documentation>
      </annotation>
      <complexContent>
        <extension base="geox:ConvexPolyhedronType"/>
      </complexContent>
    </complexType>
    <!-- ................. -->
    <element name="Wedge" type="geox:WedgeType" substitutionGroup="geox:ConvexPolyhedron">
      <annotation>
        <appinfo>
          <sch:pattern id="wedge" name="Wedge has 6 vertices">
            <sch:rule context="//geox:Wedge">
              <sch:assert test="count(geox:vertex) = 6">Wedge must have 6 vertices</sch:assert>
            </sch:rule>
          </sch:pattern>
        </appinfo>
        <documentation>The "Wedge" element is a simple solid with six vertices and five sides.</documentation>
      </annotation>
    </element>
    <!-- ============================================================ -->
    <complexType name="PyramidType">
      <annotation>
        <documentation>The "Pyramid" element is a simple solid with five vertices and five sides.
Relative to a right-handed coordinate system the conventional vertex sequence is
     4
    //\ \
 3 / -\- 2
  |/      \ |
 0____1

For cases where the face order is used implicitly, then the faces are usually defined with the following vertices (in the local coordinate system)

0 - 0 1 4
1 - 1 2 4
2 - 2 3 4
3 - 3 0 4
4 - 0 3 2 1</documentation>
      </annotation>
      <complexContent>
        <extension base="geox:ConvexPolyhedronType"/>
      </complexContent>
```

```xml
    </complexType>
    <!-- ................. -->
    <element name="Pyramid" type="geox:PyramidType" substitutionGroup="geox:ConvexPolyhedron">
        <annotation>
            <appinfo>
                <sch:pattern id="pyramid" name="Pyramid has 5 vertices">
                    <sch:rule context="//geox:Pyramid">
                        <sch:assert test="count(geox:vertex) = 5">Pyramid must have 5 vertices</sch:assert>
                    </sch:rule>
                </sch:pattern>
            </appinfo>
            <documentation>The "Pyramid" element is a simple solid with five vertices and five sides.</documentation>
        </annotation>
    </element>
    <!-- ============================================================ -->
    <complexType name="TetrahedronType">
        <annotation>
            <documentation>The "Tetrahedron" element is a simple solid with four vertices and four sides.
Relative to a right-handed coordinate system the conventional vertex sequence is
      3
   /   | \
 2- -| - -1
   \   | /
      0

 For cases where the face order is used implicitly, then the faces are usually defined with the following vertices (in the
local coordinate system)

1 - 1 2 3
2 - 2 0 3
0 - 0 1 3
3 - 0 2 1</documentation>
        </annotation>
        <complexContent>
            <extension base="geox:ConvexPolyhedronType"/>
        </complexContent>
    </complexType>
    <!-- ................. -->
    <element name="Tetrahedron" type="geox:TetrahedronType" substitutionGroup="geox:ConvexPolyhedron">
        <annotation>
            <appinfo>
                <sch:pattern id="tetrahedron" name="Tetrahedron has 4 vertices">
                    <sch:rule context="//geox:Tetrahedron">
                        <sch:assert test="count(geox:vertex) = 4">Tetrahedron must have 4 vertices</sch:assert>
                    </sch:rule>
                </sch:pattern>
            </appinfo>
            <documentation>The "Tetrahedron" element is a simple solid with four vertices and four sides.</documentation>
        </annotation>
    </element>
    <!-- ====================================================================== -->
</schema>
```

ANNEX B Possible GML Implementation for Planes and Lines

(informative)

The XML Schema representation follows the UML-GML encoding rules described in Annex E of GML 3.2 (ISO DIS 19136).

Listing 2. linePlane.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
xmlns:geox="http://www.opengis.net/geomExt/0.0"
targetNamespace="http://www.opengis.net/geomExt/0.0" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="0.0.0">
   <annotation>
      <documentation>linePlane.xsd

A GML conformant schema for specialised geometries
lines and planes

Copyright (c) 2006 Open Geospatial Consortium - see http://www.opengeospatial.org/about/?page=ipr</documentation>
   </annotation>
   <!-- ========================================================================= -->
   <!-- bring in other schemas -->
   <import namespace="http://www.opengis.net/gml"
schemaLocation="http://testschemas.opengis.net/gml/3.2.0/gml/gml.xsd"/>
   <!-- ========================================================================= -->
   <!-- ========================================================================= -->
   <!-- === Geometric primitives === -->
   <!-- ========================================================================= -->
   <!-- === Parameterised Lines === -->
   <!-- ========================================================================= -->
   <!-- ========================================================================= -->
   <complexType name="AbstractLineType" abstract="true">
      <annotation>
         <documentation>An abstraction of a line to support the different levels of complexity.
   The line can always be viewed as a geometric primitive, i.e. is logically continuous.</documentation>
      </annotation>
      <complexContent>
         <extension base="gml:AbstractCurveType"/>
      </complexContent>
   </complexType>
   <!-- ...... -->
   <element name="AbstractLine" type="geox:AbstractLineType" abstract="true" substitutionGroup="gml:AbstractCurve">
      <annotation>
         <documentation>The "AbstractLine" element is the abstract head of the substituition group for all elements
representing (infinite) Lines.</documentation>
      </annotation>
   </element>
   <!-- ================================================================ -->
   <complexType name="LineByVectorType">
      <annotation>
         <documentation>Line carries an id, so can be used as the target of a reference. This is useful so that the survey
defines a 1-D CRS used for indexing information along the borehole.</documentation>
      </annotation>
      <complexContent>
         <extension base="geox:AbstractLineType">
            <sequence>
               <element name="origin" type="gml:PointPropertyType">
                  <annotation>
                     <documentation>Line is expressed as an origin and direction.</documentation>
                  </annotation>
               </element>
```

```xml
                <choice>
                    <element name="direction" type="gml:VectorType">
                        <annotation>
                            <documentation>A simple direction relative to the origin.</documentation>
                        </annotation>
                    </element>
                    <element name="directionDescription" type="gml:CodeType"/>
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<!-- ...... -->
<element name="LineByVector" type="geox:LineByVectorType" substitutionGroup="geox:AbstractLine"/>
<element name="Line" type="geox:LineByVectorType" substitutionGroup="geox:AbstractLine"/>
<!-- ============================================================== -->
<!-- === Parameterised Planes === -->
<!-- ============================================================== -->
<complexType name="AbstractPlaneType" abstract="true">
    <annotation>
        <documentation>An abstraction of a plane to support the different levels of complexity.
  The plane can always be viewed as a geometric primitive, i.e. is logically continuous.</documentation>
    </annotation>
    <complexContent>
        <extension base="gml:AbstractSurfaceType"/>
    </complexContent>
</complexType>
<!-- ...... -->
<element name="AbstractPlane" type="geox:AbstractPlaneType" abstract="true"
substitutionGroup="gml:AbstractSurface">
    <annotation>
        <documentation>The "AbstractPlane" element is the abstract head of the substituition group for all elements
representing (infinite) planes.</documentation>
    </annotation>
</element>
<!-- ======================================================= -->
<complexType name="PlaneByVectorType">
    <annotation>
        <documentation>Plane defined by its normal and a point contined within the plane.</documentation>
    </annotation>
    <complexContent>
        <extension base="geox:AbstractPlaneType">
            <sequence>
                <element name="containedPoint" type="gml:PointPropertyType"/>
                <element name="normal" type="gml:VectorType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
<!-- ...... -->
<element name="PlaneByVector" type="geox:PlaneByVectorType" substitutionGroup="geox:AbstractPlane"/>
<!-- ============================================================== -->
</schema>
```

ANNEX C XML implementation - examples

(informative)

1 Introduction

The details of a GML implementation may be explored using instance examples. The implementation is an explicit mapping from the UML model, using the same names as XML element and attribute names, so inspection of sample data is an effective way to assess the effectiveness of the model in capturing the required information. In this clause we present a series of examples to illustrate the model and encoding.

2 Simple Solids

2.1 Hexahedron with explicit vertex positions

The example shown in Listing 3 is a description of a cube of edge-length 2 with vertex 0 at the origin of the coordinate reference system. The vertices are described as GML Point elements.

Listing 3. hex1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<geox:Hexahedron gml:id="hex1"
xmlns:geox="http://www.opengis.net/geomExt/0.0" 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/geomExt/0.0 ../convexPolyhedron.xsd">
  <geox:vertex>
    <gml:Point gml:id="p23">
      <gml:pos>0 0 0</gml:pos>
    </gml:Point>
  </geox:vertex>
  <geox:vertex>
    <gml:Point gml:id="p87">
      <gml:pos>2 0 0</gml:pos>
    </gml:Point>
  </geox:vertex>
  <geox:vertex>
    <gml:Point gml:id="p99">
      <gml:pos>2 2 0</gml:pos>
    </gml:Point>
  </geox:vertex>
  <geox:vertex>
    <gml:Point gml:id="p10">
      <gml:pos>0 2 0</gml:pos>
    </gml:Point>
  </geox:vertex>
  <geox:vertex>
    <gml:Point gml:id="p34">
      <gml:pos>0 0 2</gml:pos>
    </gml:Point>
  </geox:vertex>
  <geox:vertex>
    <gml:Point gml:id="p7">
```



```
        <gml:pos>2 0 2</gml:pos>
      </gml:Point>
    </geox:vertex>
    <geox:vertex>
      <gml:Point gml:id="p157">
        <gml:pos>2 2 2</gml:pos>
      </gml:Point>
    </geox:vertex>
    <geox:vertex>
      <gml:Point gml:id="p88">
        <gml:pos>0 2 2</gml:pos>
      </gml:Point>
    </geox:vertex>
</geox:Hexahedron>
```

2.2 Hexahedron with vertex positions provided indirectly

The example shown in Listing 4 is a description of a hexahedron, in which the vertices are described using the GML “by reference” pattern.

This matches the very common pattern in 3-D application software, where solid elements are described using indexes into a list of points or positions that are listed separately.

Listing 4. hex2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:GeometricComplex
      gml:id="gc_hex3"
      xmlns:geox="http://www.opengis.net/geomExt/0.0"
      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/geomExt/0.0 ../convexPolyhedron.xsd">
    <gml:element>
      <gml:Point gml:id="p23">
        <gml:pos>0 0 0</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p87">
        <gml:pos>2 0 0</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p99">
        <gml:pos>2 2 0</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p10">
        <gml:pos>0 2 0</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p34">
        <gml:pos>0 0 2</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p7">
        <gml:pos>2 0 2</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
```

```
      <gml:Point gml:id="p157">
        <gml:pos>2 2 2</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p88">
        <gml:pos>0 2 2</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <geox:Hexahedron gml:id="hex3">
        <geox:vertex xlink:href="#p23"/>
        <geox:vertex xlink:href="#p87"/>
        <geox:vertex xlink:href="#p99"/>
        <geox:vertex xlink:href="#p10"/>
        <geox:vertex xlink:href="#p34"/>
        <geox:vertex xlink:href="#p7"/>
        <geox:vertex xlink:href="#p157"/>
        <geox:vertex xlink:href="#p88"/>
      </geox:Hexahedron>
    </gml:element>
</gml:GeometricComplex>
```

Listing 4 may be compared with Listing 5 in which the same information is presented as a GML Solid. Note that (a) the six faces of the hexahedron appear as separate PolygonPatch elements; (b) the LinearRing bounding each PolygonPatch is defined by five points, the first one being repeated in order to close the ring; (c) the topology of the solid is provided by listing 30 references to Point elements, in contrast with only eight according to the proposed schema in which vertex order is fixed.

Listing 5. solid1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<gml:GeometricComplex gml:id="gc_hex3"
      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/geomExt/0.0 ../convexPolyhedron.xsd">
    <gml:element>
      <gml:Point gml:id="p23">        <!-- vertex 0 -->
        <gml:pos>0 0 0</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p87">        <!-- vertex 1 -->
        <gml:pos>2 0 0</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p99">        <!-- vertex 2 -->
        <gml:pos>2 2 0</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p10">        <!-- vertex 3 -->
        <gml:pos>0 2 0</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p34">        <!-- vertex 4 -->
        <gml:pos>0 0 2</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
```

```
      <gml:Point gml:id="p7">           <!-- vertex 5 -->
        <gml:pos>2 0 2</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p157">           <!-- vertex 6 -->
        <gml:pos>2 2 2</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Point gml:id="p88">           <!-- vertex 7 -->
        <gml:pos>0 2 2</gml:pos>
      </gml:Point>
    </gml:element>
    <gml:element>
      <gml:Solid gml:id="solid1">
        <gml:exterior>
          <gml:Shell>
            <gml:surfaceMember>
              <gml:Surface gml:id="surface57">
                <gml:patches>
                  <gml:PolygonPatch>                    <!-- face 0 -->
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pointProperty xlink:href="#p23"/>
                        <gml:pointProperty xlink:href="#p34"/>
                        <gml:pointProperty xlink:href="#p7"/>
                        <gml:pointProperty xlink:href="#p87"/>
                        <gml:pointProperty xlink:href="#p23"/>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:PolygonPatch>
                  <gml:PolygonPatch>                    <!-- face 1 -->
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pointProperty xlink:href="#p87"/>
                        <gml:pointProperty xlink:href="#p7"/>
                        <gml:pointProperty xlink:href="#p157"/>
                        <gml:pointProperty xlink:href="#p99"/>
                        <gml:pointProperty xlink:href="#p87"/>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:PolygonPatch>
                  <gml:PolygonPatch>                    <!-- face 2 -->
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pointProperty xlink:href="#p99"/>
                        <gml:pointProperty xlink:href="#p157"/>
                        <gml:pointProperty xlink:href="#p88"/>
                        <gml:pointProperty xlink:href="#p10"/>
                        <gml:pointProperty xlink:href="#p99"/>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:PolygonPatch>
                  <gml:PolygonPatch>                    <!-- face 3 -->
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pointProperty xlink:href="#p10"/>
                        <gml:pointProperty xlink:href="#p88"/>
                        <gml:pointProperty xlink:href="#p34"/>
                        <gml:pointProperty xlink:href="#p23"/>
                        <gml:pointProperty xlink:href="#p10"/>
                      </gml:LinearRing>
                    </gml:exterior>
                  </gml:PolygonPatch>
                  <gml:PolygonPatch>                    <!-- face 4 -->
                    <gml:exterior>
                      <gml:LinearRing>
                        <gml:pointProperty xlink:href="#p23"/>
                        <gml:pointProperty xlink:href="#p87"/>
                        <gml:pointProperty xlink:href="#p99"/>
```

```
                      <gml:pointProperty xlink:href="#p10"/>
                      <gml:pointProperty xlink:href="#p23"/>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:PolygonPatch>
                <gml:PolygonPatch>                                  <!-- face 5 -->
                  <gml:exterior>
                    <gml:LinearRing>
                      <gml:pointProperty xlink:href="#p34"/>
                      <gml:pointProperty xlink:href="#p88"/>
                      <gml:pointProperty xlink:href="#p157"/>
                      <gml:pointProperty xlink:href="#p7"/>
                      <gml:pointProperty xlink:href="#p34"/>
                    </gml:LinearRing>
                  </gml:exterior>
                </gml:PolygonPatch>
              </gml:patches>
            </gml:Surface>
          </gml:surfaceMember>
        </gml:Shell>
      </gml:exterior>
    </gml:Solid>
  </gml:element>
</gml:GeometricComplex>
```

3 Planes and Lines

3.1 Plane by vector

Listing 6 shows the description of a plane passing through the point at (1,1,1) whose vector normal is in the direction parallel to the vector whose direction is (2,3,1). In Listing 7 a parallel plane is described, passing through a point indicated "by reference".

Listing 6. Plane1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<geox:PlaneByVector gml:id="plane1"
xmlns:geox="http://www.opengis.net/geomExt/0.0"
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/geomExt/0.0 ../linePlane.xsd">
  <geox:containedPoint>
    <gml:Point gml:id="p79">
      <gml:pos>1 1 1</gml:pos>
    </gml:Point>
  </geox:containedPoint>
  <geox:normal>2 3 1</geox:normal>
</geox:PlaneByVector>
```

Listing 7. Plane2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<geox:PlaneByVector gml:id="plane2"
xmlns:geox="http://www.opengis.net/geomExt/0.0"
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/geomExt/0.0 ../linePlane.xsd">
  <geox:containedPoint xlink:href="http://some.bogus.org/service?p79"/>
  <geox:normal>2 3 1</geox:normal>
</geox:PlaneByVector>
```

3.2 Line by vector

Listing 8 shows the description of a line passing through the point at (1,1,1) in the direction parallel to the vector whose direction is (2,3,1). In Listing 9 a parallel line is described, passing through a point indicated “by reference”.

Listing 8. Line1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<geox:LineByVector gml:id="line1"
xmlns:geox="http://www.opengis.net/geomExt/0.0"
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/geomExt/0.0 ../linePlane.xsd">
  <geox:origin>
    <gml:Point gml:id="p79">
      <gml:pos>1 1 1</gml:pos>
    </gml:Point>
  </geox:origin>
  <geox:direction>2 3 1</geox:direction>
</geox:LineByVector>
```

Listing 9. Line2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<geox:LineByVector gml:id="line2"
xmlns:geox="http://www.opengis.net/geomExt/0.0"
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/geomExt/0.0 ../linePlane.xsd">
  <geox:origin xlink:href="http://some.bogus.org/service?p79"/>
  <geox:direction>2 3 1</geox:direction>
</geox:LineByVector>
```