Open Geospatial Consortium Inc.

Date: 2007-05-10

Reference number of this document: OGC 07-031r1

Version: 0.0.3

Category: OpenGIS® Implementation Specification

Editor: Arliss Whiteside

OpenGIS® Image geopositioning metadata GML 3.2 application schema

Warning

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

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.

Document type:	OpenGIS® Implementation Specification
Document subtype:	GML Application Schema RFC
Document stage:	Proposed version 0.1.0
Document language:	English

Contents

Page

Figures

Page

Tables

Page

i. Preface

This is a proposed GML 3.2 Application Schema Implementation Specification for image geopositioning metadata. This schema is also an Application Schema of ISO 19139. This geopositioning metadata schema is used by the separately specified Image Geopositioning Service (IGS) interface, but is specified separately because it is expected to also be used by other services.

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

ii. Document terms and definitions

This document uses the specification terms 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 conform to this specification.

iii. Submitting organizations

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

BAE Systems E&IS

iv. Document contributor contact points

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

Name	Organization
Arliss Whiteside	BAE Systems E&IS

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2006-06-05	0.0.0	Arliss Whiteside	All	First draft
2006-07-03	0.0.0	Arliss Whiteside	vii, 7.2, 8.2, 9.2, 10.2, 11.2	Editorial improvements
2006-11-30	0.0.2	Arliss Whiteside	12, TBD	Added IGM file (data package) types definitions
2007-03-21	0.0.3	Arliss Whiteside	vi, 4, 7.3, 8.2, 10, 12.2, 14, B, C	Separated AdjustedGroup UML package and XML Schema document, revised IGM file (data package) types definitions, added Annex C, general improvement

vi. Changes to the OGC Abstract Specification

The OpenGIS® Abstract Specification requires the following changes to accommodate the technical contents of this document:

a) Change the type of the valueUnit attribute of the DQ_QuantitativeResult class, in Figure A.4 of Subclause A.2.4.1 and in item 135 of Subclause B.2.4.4.4 of Topic 11 "Metadata" (which includes ISO 19115). (The corresponding change is also needed in ISO 19139.) The valueUnit type is currently UnitOfMeasure, meaning a full definition of the unit of measure. In general, only a reference to a unit of measure is needed, since the same units are used multiple times. In this specific application, an ordered sequence of references to units of measure is needed, with the same units often used multiple times in one sequence.

b) Change the type of the association from the CC_ParameterValueGroup class to the CC_GeneralParameterValue class, in Figure 12 in Subclause 11.2 of Topic 2 "Spatial referencing by coordinates" (which contains ISO 19111). The corresponding change is also needed in the GML 3.2 Implementation Specification. This association is currently a composition association to a parameter value group, which thus cannot be used more than once. In this specific application, a general association should be allowed, to avoid requiring duplication of a parameter value group when the same values are needed more than once. Multiple uses of the same parameter value group is common for interior orientation parameters, which are the same for many images collected by the same image sensor.

vii. Future work

Extensions of this Application Profile are desirable to:

a) Define specific image geometry (sensor) models. To allow useful implementation, at least one extension of this Application Profile is needed, that specifies at least one specific image geometry model. For each specific image geometry model added to this Application Schema, it is necessary to specify:

1) A concrete subclass of one of the abstract classes GIG_MovingImageParameters or GIG_StationaryImageParameters, which specifies the (exterior) orientation parameter values for that sensor
2) A concrete subclass of the abstract class GSC_SensorParameterValues, which specifies the (interior) orientation parameter values for that sensor
3) GML Application Schema for encoding the two items listed above
4) Specific parameters and sub-groups in a CC_OperationParametersGroup, which correspond to items 1) and 2)
5) XML document(s) encoding the two items listed above
6) Specific “formula” referenced by the CC_OperationMethod that is referenced by GSM_ObjectImageTransformation, which uses the parameters and groups specified in items 3) and 4)

NOTE 1 More generally, several extensions of this Application Profile that each specifies a specific image geometry model seem desirable. As stated and explained in Section 2.10 of OGC Abstract Specification Topic 16 [OGC 00-116]: “A number of different image geometry models are needed, for exploiting different image types under different conditions. Multiple different image geometry models should thus be standardized by the OGC (in the long term). However, some proprietary image geometry models are expected to exist, and not be standardized.”

NOTE 2 Where relevant, these specific image geometry models should build upon and adapt the ISO 19130 (Sensor data model for imagery and gridded data), Sensor Markup Language (SensorML), and Transducer Markup Language (TML).

EXAMPLE An example extension of this Application Profile for one specific image geometry model is provided in the Frame image geopositioning metadata GML Application Profile [OGC 07-032].

b) Define image geometry (sensor) model components. For efficient extension of this Application Profile to more models, at least one extension of this Application Profile is needed, that specifies several components which are and/or can be used in multiple image geometry and sensor models.

EXAMPLES Possible components that could be used in multiple image geometry models would model lens distortion, atmospheric refraction, projection optics, correction functions, etc.

c) Stereoscopic pairs of images. Extension is desirable to more completely handle pairs of images that are expected to be exploited stereoscopically.

d) More covariance matrix types. Define additional subclasses of GCM_Matrix for efficiently recording other covariance matrix types.

e) Triangulation results. Extension is desirable to more completely specify useful results from a triangulation

f) Image strip. Improvement is desirable to better handle strips of images.

g) Examples. Extension is desirable to include more example XML documents.

h) Specify GML 3.2 and ISO 19139 profiles used. The profiles of GML 3.2 and ISO 19129 that are used by this Application Schema should be explicitly determined and documented, to simplify understanding those profiles.

Foreword

This document supersedes OGC Discussion Paper 06-055r1, with a different number in 2007. This document does not replace any other previous OGC document, in whole or in part. A subset profile of this Image geopositioning metadata GML Application Schema is used by the separately specified Image Geopositioning Service (IGS) and other services.

This Application Schema uses small profiles (or subsets) of GML 3.2 (ISO 19136) and ISO 19139, although those profiles are not yet formally specified. For GML, the profile used is a subset of the GML grid CRSs and simple features profiles. For ISO 19139, the profile is a subset of the ISO 19139 profile that is used by GML 3.2 (ISO 19136).

This document includes three annexes; Annexes A and B are normative, and Annex C is informative.

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

Introduction

This document specifies a GML 3.2 Application Schema for image geopositioning metadata, which is also an Application Schema of ISO 19139. This image geopositioning metadata schema is used by the separately specified Image Geopositioning Service (IGS) interface that adjusts the georeferencing coordinate transformations of images. This schema can also be used by other future OGC Web Services.

This XML schema encodes image georeferencing coordinate transformations with associated parameter error statistics. These georeferencing coordinate transformations can use many possible image geometry (or sensor) models that can be encoded using extensions of this Application Schema.

This Application Schema also encodes point positions measured in one of more images and optional object coordinates, with associated position error statistics. These object points can be tie points, control points, or check points. A control or check point has a measured position with position error statistics in one or more images, and a known position with error statistics in some geodetic Coordinate Reference System (CRS). A tie point has a measured position with error statistics in two or more images, but not a known position in any geodetic CRS.

Error statistics are represented as variance-covariance matrices, representing both absolute and relative accuracies. These covariance matrices are used to represent correlations between the accuracies of different parameters, coordinates, and positions.

OpenGIS® Image geopositioning metadata GML 3.2 application schema

1 Scope

This document specifies a GML 3.2 Application Schema for image geopositioning metadata, which is also an Application Schema of ISO 19139. This image geopositioning metadata is designed for use by a separately specified Image Geopositioning Service (IGS) that adjusts the georeferencing coordinate transformations of images. This schema can also be used by other future OGC Web Services.

This XML schema encodes image georeferencing coordinate transformations with associated parameter error statistics. These georeferencing coordinate transformations can use many possible image geometry (or sensor) models that can be encoded using extensions of this Application Schema.

This Application Schema also encodes object point positions measured in one of more images and optional object coordinates, with associated position error statistics. These object points can be tie points, control points, and check points.

Error statistics are represented as variance-covariance matrices, representing both absolute and relative accuracies. These covariance matrices are used to represent correlations between the accuracies of different parameters, coordinates, and positions.

2 Compliance

Compliance with this specification shall be checked using all the relevant tests specified in Annex A (normative).

3 Normative references

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

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

ISO 19109:2005, *Geographic information — Rules for application schema*

ISO 19115:2003, *Geographic information — Metadata*

ISO 19123:2005, *Geographic information — Schema for coverage geometry and functions*

OGC 01-111, *OpenGIS® Metadata* (contains ISO/TC 211 DIS 19115)

OGC 05-095r1, *GML 3.1.1 common CRSs profile*

OGC 05-096r1, *GML 3.1.1 grid CRSs profile*

OGC 05-103, *Geographic information — Spatial referencing by coordinates*

OGC 05-105, *Geographic information — Geography Markup Language (GML)* (version 3.2, ISO/DIS 19136)

OGC 06-023r1, *URNs of definitions in ogc namespace*

OGC 06-121r3, *OpenGIS® Web Services Common Specification*, version 1.0.0

NOTE This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

In addition to this document, this specification includes several normative XML Schema Document files as specified in Annex B.

4 Terms and definitions

For the purposes of this specification, the definitions specified in Clause 4 of the OpenGIS® Abstract Specification Topic 2 [OGC 05-103] and in OWS Common Implementation Specification [OGC 06-121r3] shall apply. In addition, the following terms and definitions apply.

4.1
absolute accuracy
absolute external accuracy
statistic which gives the uncertainty of a point with respect to the datum required by a product specification (adapted from OGC 00-115)

NOTE This definition implies that the effects of all error sources, both random and systematic, are considered.

4.2
accuracy
degree to which information on a map or in a digital database matches true or accepted values (adapted from OGC 00-115)

4.3
check point
point with known object (or ground) position used to check the geopositioning of one or more images

NOTE The known position of a check point is not used in the geopositioning. The position of a check point is measured in one or more of the images geopositioned.

4.4
control point
point with known object (or ground) position used to geoposition one or more images

NOTE The known position of a control point is used in the geopositioning. The position of a control point is measured in one or more of the images being geopositioned.

4.5

covariance matrix

detailed form of position accuracy data, sometimes called a variance-covariance matrix (adapted from OGC 00-115)

NOTE 1 For three object (or ground) coordinates, a covariance matrix is a 3 by 3 matrix, with the matrix rows and columns each corresponding to the three coordinates. For two horizontal coordinates, a covariance matrix is a 2 by 2 matrix, with the matrix rows and columns each corresponding to the two horizontal coordinates. Similarly, for two image coordinates, a covariance matrix is a 2 by 2 matrix, with the matrix rows and columns each corresponding to the two image coordinates.

NOTE 2 The covariance matrix cells contain the expected average values of the product of the error in the matrix row coordinate times the simultaneous error in the matrix column coordinate. For absolute accuracy, the diagonal matrix cells contain the error variances of the corresponding coordinates, or the squares of the standard deviations. The off-diagonal cells contain the covariances between the errors in the corresponding coordinates; these covariances will be zero when the errors in different coordinates are not statistically correlated. All covariance matrices are symmetrical, meaning that the same cell values appear on both sides of the diagonal cells.

4.6

georeferencing transformation

coordinate transformation that can be used to convert grid coordinate values to values of coordinates referenced to a coordinate reference system that is related to the earth by a datum (adapted from ISO 19123)

4.7

image geopositioning

adjustment of the parameter values of image georeferencing coordinate transformations to produce correct coordinates in a coordinate reference system that is related to the earth or other imaged object

4.8

object point

ground point

point with position on the imaged object(s), often the earth

4.9

photogrammetry

science of mensuration and geometric adjustment of an aerial photograph or satellite image (adapted from OGC 00-115)

NOTE 1 Photogrammetry requires a mathematical model of the image formation process, computation of the internal geometry of an image, and subsequent correction of imagery based upon the ground relationship for every part of the image. This correction of imagery is based on computational algorithms and measurement of geometrical position in an image.

NOTE 2 Effective photogrammetry makes use of ground control points by which photographs are carefully compared and registered to the locations and characteristics of features identified in ground-level surveys.

4.10
relative accuracy
relative internal accuracy
evaluation of random errors in determining the positional accuracy of one point feature with respect to another feature (adapted from OGC 00-115)

4.11
tie point
point with measured position in one or more images, used to geoposition those images

NOTE The estimated object (or ground) position of a tie point is not known before the geopositioning.

4.12
transformation
approximate transformation of position coordinates from one Spatial Reference System (SRS) to another

NOTE For example, this term is used when the transformation coefficients are determined by least squares adjustment. This term is strictly used only when the transformation is known only approximately. This term is loosely used when the transformation is known either approximately or exactly.

5 Conventions

5.1 Abbreviated terms

CRS	Coordinate Reference System
GML	Geography Markup Language
IGS	Image Geopositioning Service
IGM	Image Geopositioning Metadata
ISO	International Organization for Standardization
OGC	Open Geospatial Consortium
OWS	OGC Web Service, or Open Web Service
TBD	To Be Determined
TBR	To Be Reviewed
UML	Unified Modeling Language
URI	Universal Resource Identifier
URL	Uniform Resource Locator
URN	Universal Resource Name
XML	Extensible Markup Language
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional

5.2 UML notation

The diagrams that appear in this specification are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

5.3 Platform-neutral and platform-specific specifications

As specified in Subclause 5.4 of OWS Common [OGC 06-121r3], this document includes both platform-neutral and platform-specific specifications.

EXAMPLES 1 Platform-neutral specifications are contained in Subclauses 7.2, 8.2, 9.2, 10.2, 11.2, and 12.2.

EXAMPLES 2 Platform-specific specifications for XML encoding are contained in the attached normative XML Schema Documents, referenced in Subclauses 7.3, 8.3, 9.3, 10.3, 11.3, and 12.3.

6 Image geopositioning metadata overview

6.1 Image geopositioning metadata

This GML Application Schema XML schema encodes parameter values for image georeferencing coordinate transformations with associated parameter error statistics. These georeferencing coordinate transformations can use many possible image geometry (or sensor) models that can be encoded using extensions of this Application Schema.

This Application Schema also encodes point positions measured in one of more images and optional object coordinates, with associated position error statistics. These object points can be tie points, control points, and check points. A control or check point has a measured position with position error statistics in one or more images, and a known position with error statistics in some geodetic Coordinate Reference System (CRS). A tie point has a measured position with error statistics in two or more images, but not a known position in any geodetic CRS

Error statistics are represented as variance-covariance matrices, representing both absolute and relative accuracies. These covariance matrices are used to represent correlations between the accuracies of different parameters, coordinates, and positions.

6.2 UML model packages

This Image geopositioning metadata GML Application Schema is specified in six parts, corresponding to six UML packages with six corresponding XML Schema Documents. The six UML packages are listed and briefly described in Table 1. The dependencies among these geopositioning metadata UML packages are shown in the Figure 1, together with the classes in each package.

Table 1 — Image geopositioning metadata UML model packages

Package Name	Description
GSM_SensorModel	Records versions of object-to-image transformations and the sensor mathematical models used by these transformations
GIG_ImageGeometry	Records image orientation data, allowing multiple adjustments of orientation data
GSC_SensorCalibration	Records image sensor calibration data, including multiple adjustments of interior orientation data
GAG_AdjustedGroup	Records groups of images adjusted together, allowing multiple adjustments of same group
GOP_ObjectPoint	Records object point positions in one of more images with optional object coordinates, for control points, tie points, and check points
GCM_CovarianceMatrix	Records covariance matrices, for absolute and relative parameter and position error estimates

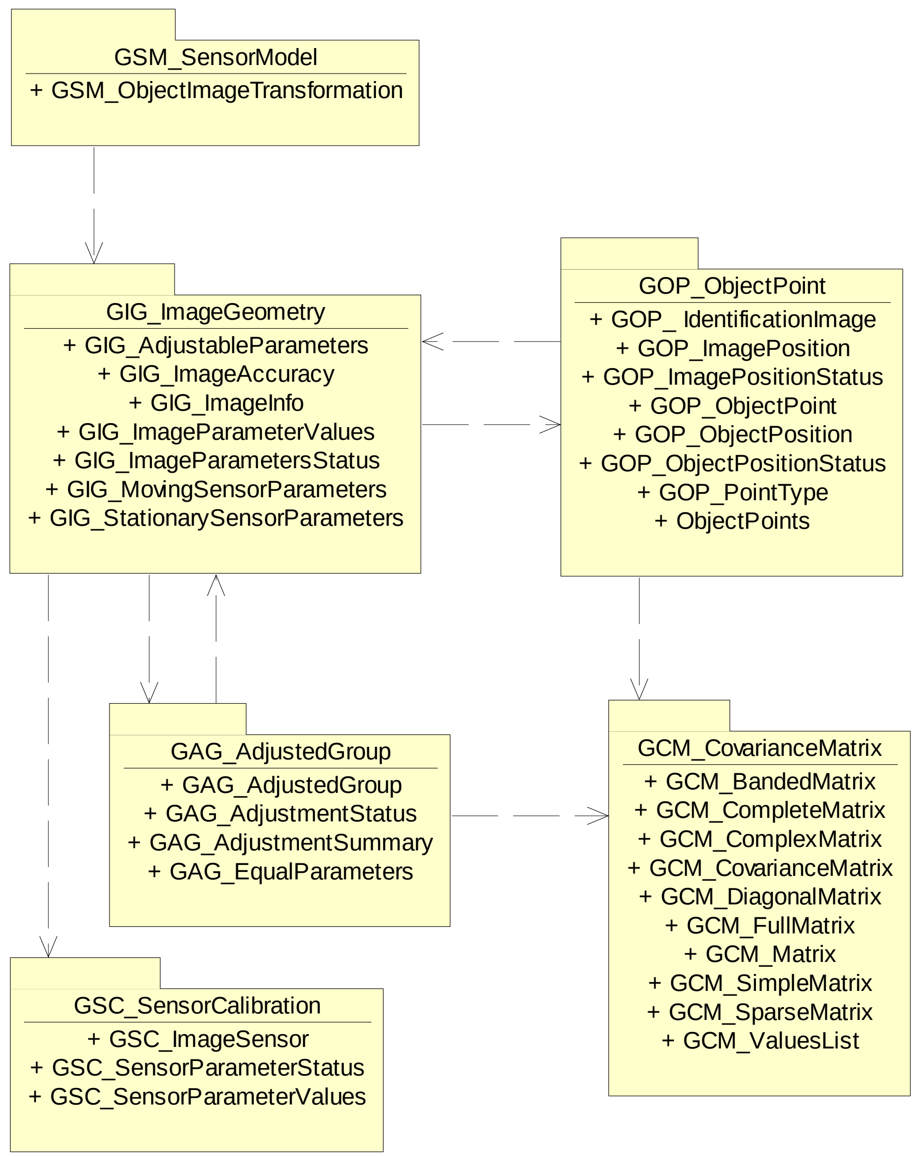

Figure 1 — Image geopositioning metadata UML packages with dependencies

This UML model is built upon the ISO/TC 211 UML model, contained in the ISO 191XX series of standards. The ISO/TC 211 UML model packages used, directly and indirectly, are listed in Table 2. The direct dependencies of the geopositioning metadata UML model packages on ISO/TC 211 UML model packages are shown in the Figure 2 (without the classes in each package).

Table 2 — ISO/TC 211 UML model packages used

ISO Standard	UML package name
19107 Geographic information — Spatial schema	GM_Geometry
19109 Geographic information — Rules for application schema	GF_GeneralFeature
19111 Geographic information — Spatial referencing by coordinates	CC_CoordinateOperation CD_Datum CS_CoordinateSystem IO_IdentifiedObject SC_CoordinateReferenceSystem
19115 Geographic information — Metadata	DQ_DataQuality EX_Extent

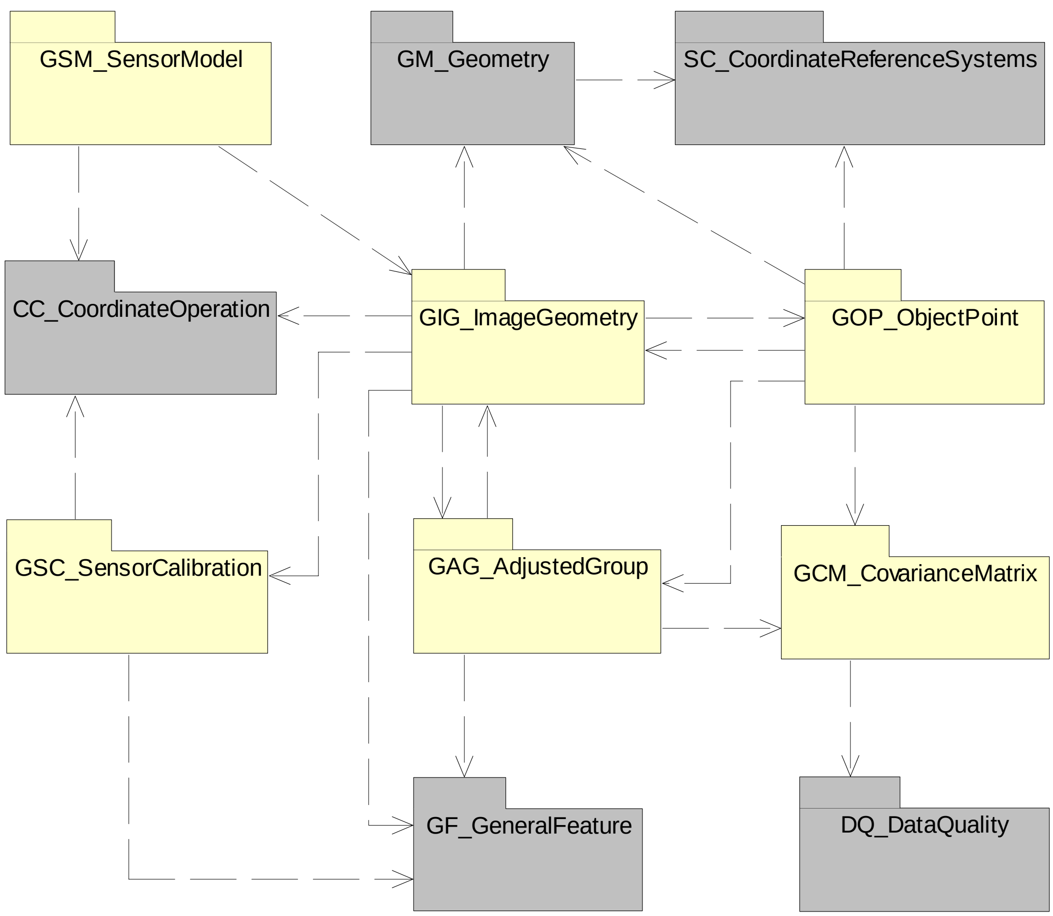

Figure 2 — Image geopositioning metadata and ISO/TC 211 packages

6.3 GML application schema

This GML 3.2 Application Schema for geopositioning metadata is also an Application Schema of ISO 19139. This Application Schema uses small profiles (or subsets) of GML 3.2 (ISO 19136) and ISO 19139, although those profiles are not yet formally specified. For GML, the profile used is essentially a subset of the GML 3.1.1 grid CRSs and simple features profiles. For ISO 19139, the profile is a subset of the ISO 19139 profile that is used by GML 3.2 (ISO 19136).

Most of the ISO/TC 211 classes used in the UML model are encoded by the GML 3.2 XML elements as listed in Table 3.

Table 3 — GML 3.2 elements encoding ISO classes

ISO package name (ISO number) UML class name	GML XML Schema Document XML element name
CC_CoordinateOperation (19111) CC_Transformation	coordinateOperations.xsd Transformation
CC_CoordinateOperation (19111) CC_OperationMethod	coordinateOperations.xsd OperationMethod
CC_CoordinateOperation (19111) CC_ParameterValueGroup	coordinateOperations.xsd ParameterValueGroup
CC_CoordinateOperation (19111) CC_OperationParameterGroup	coordinateOperations.xsd ParameterGroup
GF_GeneralFeature (19109) GF_FeatureType	dictionary.xsd Definition
GM_Geometry (19107) GM_Polygon	geometryBasic2d.xsd Polygon
GM_Geometry (19107) GM_Point GM_Vector GM_Curve	geometryBasic0d1d.xsd Point Vector AbstractCurve

Each of these geopositioning metadata UML packages and corresponding XML Schema Documents is defined in subsequent clauses.

7 Sensor model

7.1 Introduction

The sensor model part of the Image geopositioning metadata GML Application Schema supports recording versions of object-to-image coordinate transformations and the (image geometry) sensor mathematical models used by these transformations. These object-to-image transformations are used for georeferencing coordinate transformations. These coordinate transformations can use many possible image geometry (or sensor) models that can be encoded using extensions of this Application Schema. The parameter values used by these transformations are recorded in the image geometry and sensor calibration parts of this Application Schema.

7.2 UML model

The UML class diagram for the GSM_SensorModel package is shown in Figure 3, together with the CC_CoordinateOperation classes that are directly inherited from and used. The new GSM_ObjectImageTransformation class defined in this GSM_SensorModel package is defined in Table 4. This diagram also shows two other classes added in this Image geopositioning metadata GML Application Schema, namely GIG_ImageParameterValues and GSC_SensorParameterValues.

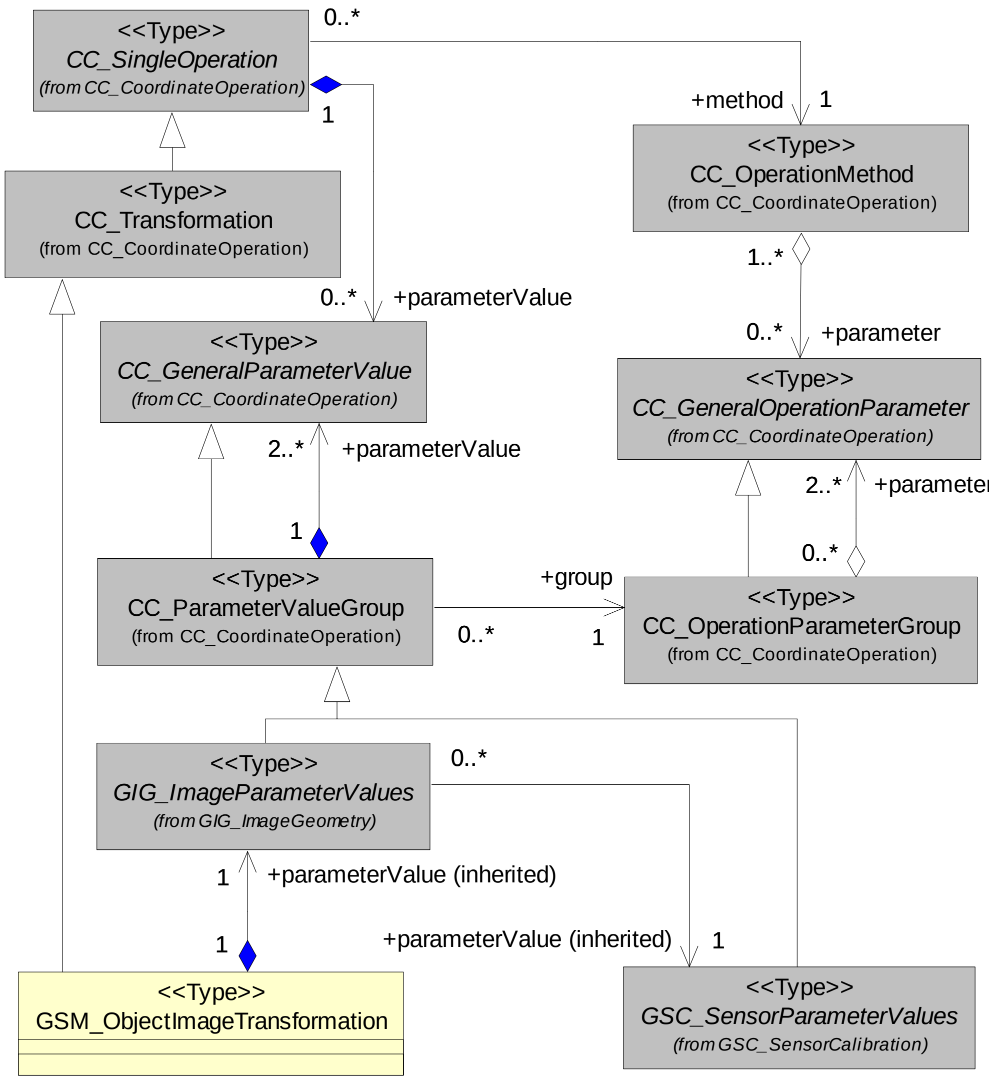

Figure 3 — GSM_SensorModel package UML class diagram

NOTE For simplicity, this diagram does not show the CC_ParameterValue concrete subclass of the CC_GeneralParameterValue abstract class, or the CC_Parameter concrete subclass of the CC_GemeralParameter abstract class. Those concrete subclasses are not used in this document, but are used by extensions of this metadata for specific image sensor geometry models.

The GSM_SensorModel package specializes the CC_CoordinateOperation package in ISO 19111 [OGC 05-103], by defining more-specific subclasses. By specializing the CC_CoordinateOperation package, the GSM_SensorModel package uses the existing

ISO 19111 concepts and terms, for coordinate Transformations, Operation Methods, Operation Parameter Groups, etc.

Specifically, the GSM_ObjectImageTransformation class specializes the CC_Transformation class for image georeferencing object-to-image coordinate transformations. That new class uses the GIG_ImageParameterValues and GSC_SensorParameterValues classes which specialize the CC_ParameterValueGroup class. The CC_OperationMethod and CC_OperationParameterGroup classes have instances that are specialized for this use (specialized classes are not used).

Table 4 — Defining elements of GSM_ObjectImageTransformation class

Description:	Definition of one object-to-image coordinate transformation for image georeferencing, using one group of (adjusted) image parameter values, which contains one group of (calibrated) sensor parameter values.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	CC_Transformation
Association roles:	parameterValue to GIG_ImageParameterValues [1] [a] (Composition association to values of group of image parameters used by this transformation) method to CC_OperationMethod [1] [b] (Inherited association to method used by this transformation) transformation from GIG_ImageInfo [1] (Association from image that uses this object-image transformation) adjustedTransformation from GAG_AdjustedGroup [1..*] (Association from adjusted group that adjusted this object-image transformation)
Public attributes:	11 attributes inherited from CC_Transformation and IO_IdentifiedObjectBase
a Association inherited through CC_Transformation, CC_SingleOperation, CC_GeneralPareameterValues, and CC_ParameterValueGroup.	
b Association inherited through CC_Transformation and CC_SingleOperation.	

NOTE These tables are using the table format used in ISO 19111 [OGC 05-103].

7.3 XML encoding

The GSM_SensorModel UML package is encoded in the igmSensorModel.xsd XML Schema Document, which imports the coordinateOperations.xsd XML Schema Document from GML 3.2.0. The ISO 19111 classes used in the UML model are encoded by the GML 3.2 XML elements as listed in Table 3. The contents of the igmSensorModel.xsd XML Schema Document shall be as specified in the attached file.

EXAMPLE 1 An example XML fragment for an ObjectImageTransformation is:

```
<igm:ObjectImageTransformation gml:id="Transformation999.999">
    <gml:identifier
codeSpace="IGM">Transformation999.999</gml:identifier>
    <gml:scope>Domain of targetCRS</gml:scope>
    <gml:operationVersion>0.0.0</gml:operationVersion>
    <gml:sourceCRS xlink:href="urn:ogc:def:crs:EPSG:6.8:4979"/>
    <gml:targetCRS
xlink:href="urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:TBDimageID"/>
```



```
    <gml:method xlink:href="frameOperationMethodFile.xml"/>
    <!-- ============================================== -->
    <igm:parameterValue>
       <ImageParameterValues>
          <igm:image xlink:href="imageInforFile999.xml#ImageInfo999"/>
<!-- Reference to ImageInfo element for this image -->
          <igm:inGroup
xlink:href="adjustedGroupFile999.xml#IAdjustedGroup999"/> <!--
Reference to AdjustedGroup element for this adjusted set of image
parameter values  -->
          <igm:imageParametersStatus
codeSpace="../imageParametersStatusValues.xml">adjusted</igm:imageParam
etersStatus>
          <igm:sensorParameterValues
xlink:href="sensorParametersFile999.999.xml#SensorParameters999.999"/>
<!-- Reference to SensorParameterValues element for this image -->
          <igm:adjustableParameters
xlink:href="adjustableParametersFile999.xml#AdjustableParameters999"/>
<!-- Reference to AdjustableParameters element for this image -->
          <!-- ============================================== -->
          <igm:imageAccuracySummary>
             <igm:ImageAccuracySummary>
                <igm:CE uom="m">99.9</igm:CE>
             </igm:ImageAccuracySummary>
          </igm:imageAccuracySummary>
          <!-- ============================================== -->
          <igm:sensorPosition>
             <gml:Point gml:id="Point999" srsName="LSR">
                <gml:pos>999 999 999</gml:pos>
             </gml:Point>
             <igm:operationParameter
xlink:href="frameOperationMethodFile.xml#SensorPosition"/> <!--
Reference to correct OperationParameter element -->
          </igm:sensorPosition>
          <!-- ============================================== -->
          <igm:sensorAttitude>
             <gml:Vector gml:id="Angles999" srsName="AnglesInLSR">
                <gml:pos>9.99 9.99 9.99</gml:pos>
             </gml:Vector>
             <igm:operationParameter
xlink:href="frameOperationMethodFile.xml#SensorAttitude"/> <!--
Reference to correct OperationParameter element -->
          </igm:sensorAttitude>
          <!-- ============================================== -->
          <igm:group
xlink:href="frameOperationMethodFile.xml#ImageOrientation"/> <!--
Reference to correct OperationParameterGroup element -->
       </ImageParameterValues>
    </igm:parameterValue>
</igm:ObjectImageTransformation>
```

NOTE An ObjectImageTransformation will often include inline encoding of the parameterValue element.

EXAMPLE 2 A longer example XML fragment for an ObjectImageTransformation is:

```
<igm:ObjectImageTransformation gml:id="Transformation999.999">
    <gml:identifier
codeSpace="IGM">Transformation999.999</gml:identifier>
    <gml:scope>Domain of targetCRS</gml:scope>
```



```
    <gml:operationVersion>0.0.0</gml:operationVersion>
    <gml:sourceCRS xlink:href="urn:ogc:def:crs:EPSG:6.8:4979"/>
    <gml:targetCRS
xlink:href="urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:TBDimageID"/>
    <gml:method xlink:href="templateFrameOperationMethod1.xml"/>
    <!-- ============================================== -->
    <igm:parameterValue>
       <ImageParameterValues>
          <igm:image xlink:href="imageInfoFile999.xml#ImageInfo999"/>
<!-- Reference to ImageInfo element for this image -->
          <igm:inGroup
xlink:href="adjustedGroupFile999.xml#IAdjustedGroup999"/> <!--
Reference to AdjustedGroup element for this adjusted set of image
parameter values  -->
          <igm:imageParametersStatus
codeSpace="../imageParametersStatusValues.xml">adjusted</igm:imageParam
etersStatus>
          <igm:sensorParameterValues
xlink:href="sensorParametersFile999.999.xml#SensorParameters999.999"/>
<!-- Reference to SensorParameterValues element for this image -->
          <igm:footprint>
             <gml:Polygon srsName="urn:ogc:crs:EPSG:6.0:9999"
gml:id="Polygon999">
                <gml:exterior>
                   <gml:LinearRing>
                      <gml:pos>999 999</gml:pos>
                      <gml:pos>999 999</gml:pos>
                      <gml:pos>999 999</gml:pos>
                      <gml:pos>999 999</gml:pos>
                   </gml:LinearRing>
                </gml:exterior>
                <gml:interior> <!-- Repeated for each void in exterior
ring -->
                   <gml:LinearRing>
                      <gml:pos>999 999</gml:pos>
                      <gml:pos>999 999</gml:pos>
                      <gml:pos>999 999</gml:pos>
                      <gml:pos>999 999</gml:pos>
                   </gml:LinearRing>
                </gml:interior>
             </gml:Polygon>
          </igm:footprint>
          <igm:adjustableParameters
xlink:href="adsjustableParametersFile999.xml#AdjustableParameters999"/>
<!-- Reference to AdjustableParameters element for this image -->
          <!-- ============================================== -->
          <igm:imageAccuracySummary>
             <igm:ImageAccuracySummary>
                <igm:CE uom="m">99.9</igm:CE>
                <igm:LE uom="m">99.9</igm:LE> <!-- Omit when not
stereoscopic image -->
                <igm:horizontalShear uom="m">99.9</igm:horizontalShear>
<!-- Omit when not applicable -->
                <igm:verticalShear uom="m">99.9</igm:verticalShear> <!-
- Omit when not stereoscopic image -->
             </igm:ImageAccuracySummary>
          </igm:imageAccuracySummary>
          <!-- ============================================== -->
```



```
          <igm:sensorPosition>
             <gml:Point gml:id="Point999" srsName="LSR">
                <gml:pos>999 999 999</gml:pos>
             </gml:Point>
             <igm:operationParameter
xlink:href="frameOperationMethodFile.xml#SensorPosition"/> <!--
Reference to correct OperationParameter element -->
          </igm:sensorPosition>
          <!-- =============================================== -->
          <igm:sensorAttitude>
             <gml:Vector gml:id="Angles999" srsName="AnglesInLSR">
                <gml:pos>9.99 9.99 9.99</gml:pos>
             </gml:Vector>
             <igm:operationParameter
xlink:href="frameOperationMethodFile.xml#SensorAttitude"/> <!--
Reference to correct OperationParameter element -->
          </igm:sensorAttitude>
          <!-- =============================================== -->
          <igm:group
xlink:href="frameOperationMethodFile.xml#ImageOrientation"/> <!--
Reference to correct OperationParameterGroup element -->
       </ImageParameterValues>
    </igm:parameterValue>
</igm:ObjectImageTransformation>
```

8 Image geometry

8.1 Introduction

The image geometry part of the Image geopositioning metadata GML Application Schema supports recording image (exterior) orientation parameter values, including image sensor position and attitude. This package supports multiple adjustments of this image orientation data. Most of these orientation parameter values are specific to each image collected by one image sensor, and usually must be adjusted for each image collected. These parameters can be for many possible image geometry (or sensor) models that can be encoded using extensions of this Application Schema.

8.2 UML model

The UML class diagram for the GIG_ImageGeometry package is shown in Figure 4 and Figure 5, together with the ISO 191XX UML classes that are directly inherited from. The new classes defined in this GIG_ImageGeometry package are described in Table 5 through Table 12.

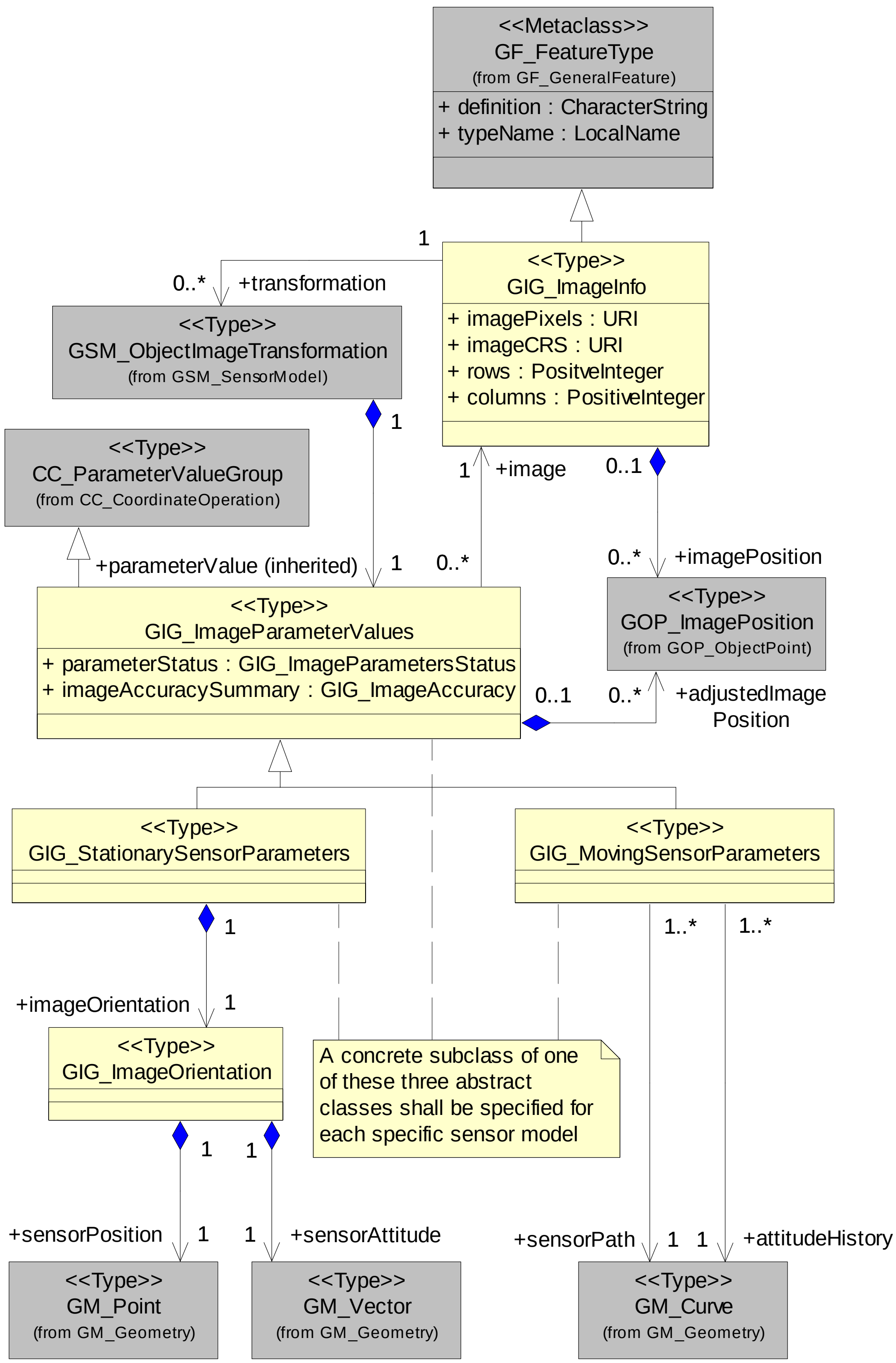

Figure 4 — GIG_ImageGeometry package UML class diagram, part 1

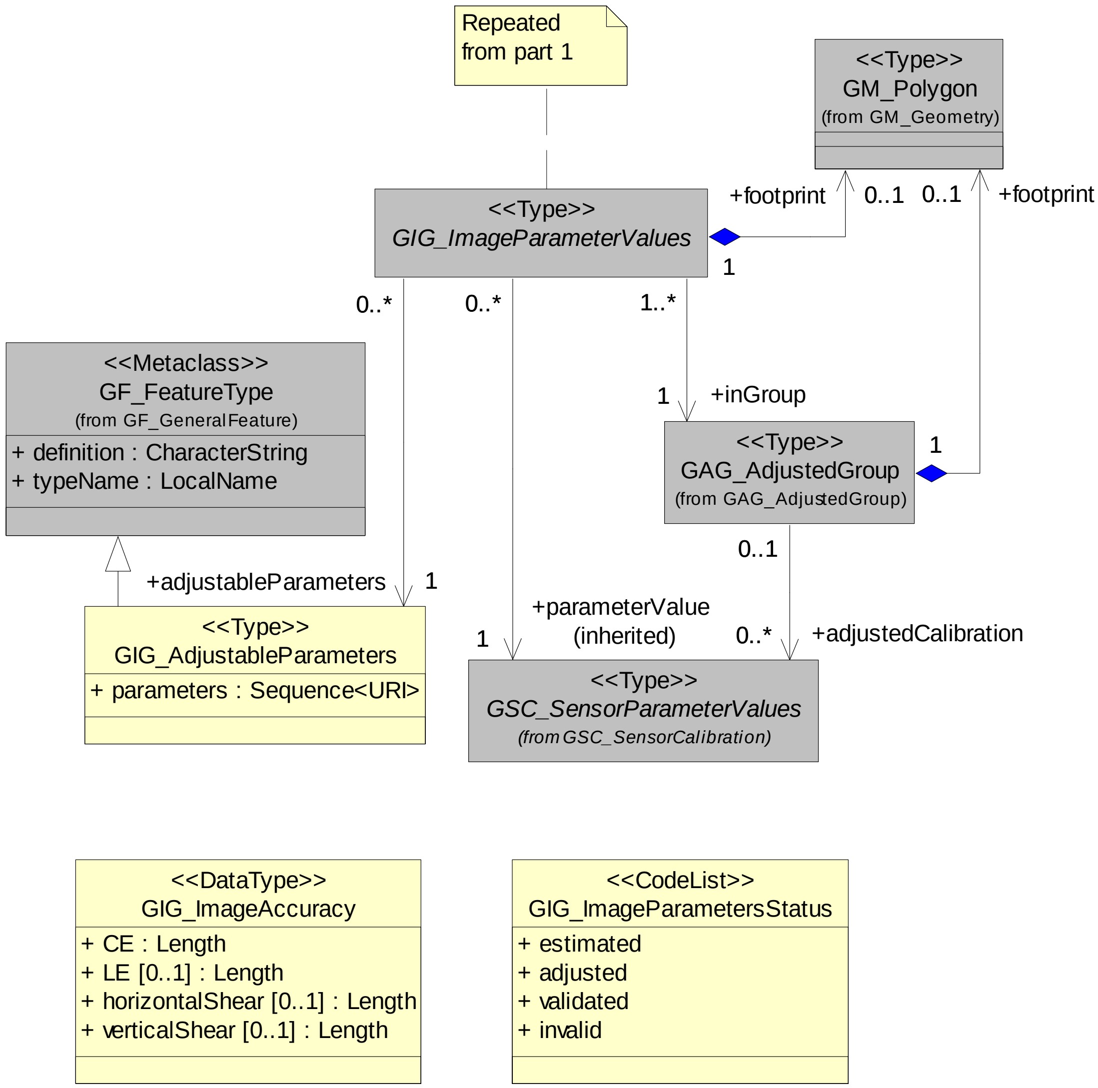

Figure 5 — GIG_ImageGeometry package UML class diagram, part 2

Part 1 of this UML class diagram shows how multiple versions of orientation parameter values can be recorded for each image. That is, multiple ImageParameterValues objects can be associated with the ImageInfo object for one image. As needed for different image types and purposes, different sensor models can be used by the multiple ImageParameterValues objects which are associated with the ImageInfo for one image. Similarly, different image parameters can be adjusted in different ImageParameterValues objects.

As indicated, the ImageInfo class contains pointers to the image pixels and the image Coordinate Reference System (CRS). It also contains the size of this image, in pixel rows and columns. This class is associated to all the ImagePositions measured in this image.

The ImageParameterValues abstract class records the parameter values for any sensor model, including the status of that group of parameter values. Each ImageParameterValues object also has an optional association to a Polygon giving the (approximate) footprint for the image using those parameter values.

The ImageParameterValues abstract class has StationaryImageParameters and Moving-ImageParameters abstract subclasses, which record parameter values common to all stationary or moving sensors. For a stationary sensor, these parameters are the sensor position coordinates, recorded using a GM_Point object, and the sensor attitude (rotation) parameters, recorded using a GM_Vector object. For a moving sensor, these parameters are the sensor path positions and the sensor attitude history, each recorded using a GM_Curve object (TBR). The sensor path and attitude history can be shared by multiple images collected in a strip, with each image specifying the collection time or range within this history.

The details specific to each specific sensor model are recorded as concrete subclasses of the StationarySensorParameters and MovingSensorParameters abstract classes, which shall be specified in extensions of this Application Schema. These classes may share more classes for common aspects (also not shown here). Wherever relevant, these specific image geometry models should build upon and adapt ISO 19130 (Sensor data model for imagery and gridded data).

Part 2 of this UML class diagram shows that each ImageParameterValues object also has an association to one AdjustableParameters object that identifies which of the image and sensor parameters could have been adjusted by the associated AdjustedGroup. Each AdjustableParameters object can be associated from multiple ImageParameterValues objects which have the same set of adjustable parameters (as is common). Similarly, multiple ImageParameterValues objects can be associated with an AdjustedGroup object for one adjustment (or triangulation). Each AdjustedGroup object is associated to the ImageParameterValues objects that were adjusted or computed by this adjustment.

The ImageGeometry package specialises the GF_FeatureType package in ISO 19109, by defining more-specific subclasses. By defining the ImageInfo class as a feature collection, identification metadata is inherited and additional properties for recording needed metadata can be easily added. By defining the AdjustableParameters as a feature, identification metadata is inherited and additional properties for recording needed metadata can be easily added.

This package also specialises the CC_CoordinateOperation package in ISO 19111, by defining more-specific subclasses of CC_ParameterValueGroup. By defining the ImageParameterValues class as a parameter value group, individual parameters and subsidiary parameter groups can be defined for different image geometry (or sensor) models. This package also uses parts of the GM_Geometry package from ISO 19107, avoiding new definitions of points, curves, and polygons.

Table 5 — Defining elements of GIG_ImageInfo class

Description:	Metadata for one image, including image identification, referencing image pixels, and associated with groups of parameter values determined for this image.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GF_FeatureType
Association roles:	transformation to GSM_ObjectImageTransformation [0..*] (Association to object-image transformation determined for this image) imagePosition to GOP_ImagePosition [0..*] (Composition association to position measured in this image) image from GIG_ImageParameterValues [1] (Association from group of parameter values determined for this image)
Public attributes:	Two attributes inherited from GF_FeatureType, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum occurrence	Attribute description
Image Pixels	imagePixels	URI	M	1	Reference to pixels of this image.
Image CRS	imageCRS	URI	M	1	Image CRS for all image positions in this image.
Rows	rows	PositiveInteger	M	1	Number of pixel rows in this image.
Columns	columns	PositiveInteger	M	1	Number of pixel columns in this image.

Table 6 — Defining elements of GIG_ImageParameterValues class

Description:	Group of parameter values for one adjustment of one image, including most exterior orientation parameters.
Stereotype:	Type
Class attribute:	Abstract
Inheritance from:	CC_ParameterValueGroup
Association roles:	image to GIG_ImageInfo [1] (Association to the image using this group of image parameter values)
	parameterValue to GSC_SensorParameterValues [1] [a] (Inherited association to the calibrated sensor parameter values used by this image parameter adjustment)
	footprint to GM_Polygon [0..1] (Composition association to approximate footprint of this image using this group of image parameter values)
	adjustableParameters to GIG_AdjustableParameters [1} (Association to list of adjustable parameters in this group)
	group to CC_OperationParameterGroup [1] [a] (Inherited association to operation parameter group for this group of image parameter values)
	adjustedImagePosition to GOP_ImagePosition [0..*] (Composition association to image positions determined by this parameter group)
	inGroup to GAG_AdjustedGroup [0..1] (Association to the adjustment group that produced this group of image parameter values)
	parameterValue from GSM_ObjectImageTransformation [1] (Association from object to image transformation that uses this set of parameter values)
Public attributes:	No attributes inherited from CC_ParameterValueGroup, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum occurrence	Attribute description
Image parameter status	imageParameterStatus	GIG_ImageParametersStatus	M	1	Status of this group of image parameter values.
Image accuracy summary	imageAccuracySummary	GIG_ImageAccuracySummary	M	1	Summary of positional accuracy of this image using this group of parameter values.

a Association inherited from CC_ParameterValueGroup.

NOTE 1 Each GM_Polygon object references the geodetic coordinate reference system of that polygon.

Table 7 — Defining elements of GIG_MovingSensorParameters class

Description:	Generic group of image parameter values for one image collected by a sensor which moved during image collection. A concrete subclass of this abstract class shall be specified for each specific moving image geometry (sensor) mathematical model, in an extension of this Image geopositioning metadata GML Application Schema.
Stereotype:	Type
Class attribute:	Abstract
Inheritance from:	GIG_ImageParameterValues
Association roles:	sensorPath to GM_Curve [1] (Association to sensor path position coordinates during the collection period of one or more images)
	sensorAttitudeHistory to GM_Curve (TBR) [1] [a] (Association to sensor attitude "history" during the collection period of one or more images)
	Plus associations inherited from GIG_ImageParameterValues class
Public attributes:	One attribute inherited from GIG_ImageParameterValues
a This geopositioning metadata structure assumes that GM_Curve is interpreted to allow recording of the four quantities used to specify the attitude of an imaging sensor, in place of the three coordinates of an imaging sensor position.	

Table 8 — Defining elements of GIG_StationarySensorParameters class

Description:	Generic group of image parameter values for one image collected by a sensor which was stationary during image collection. A concrete subclass of this abstract class shall be specified for each specific stationary image geometry (sensor) mathematical model, in an extension of this Image geopositioning metadata GML Application Schema.
Stereotype:	Type
Class attribute:	Abstract
Inheritance from:	GIG_ImageParameterValues
Association roles:	imageOrientation to GIG_ImageOrientation [1] (Composition association to image orientation at the collection time)
	Plus associations inherited from GIG_ImageParameterValues class
Public attributes:	One attribute inherited from GIG_ImageParameterValues

NOTE 2 All GM_Curve and GM_Point objects reference the geodetic coordinate reference system for these positions and attitudes

Table 9 — Defining elements of GIG_ImageOrientation class

Description:	Values of the parameter group for the sensor position and attitude at the image collection time.
Stereotype:	Type
Class attribute:	concrete
Inheritance from:	(none)
Association roles:	sensorPosition to GM_Point [1] (Composition association to sensor position coordinates at the collection time) sensorAttitude to GM_Vector [1] (Composition association to sensor attitude at the collection time) group to CC_OperationParameterGroup [1] [a] (Inherited association to operation parameter group for this group of parameter values) imageOrientation from GIG_StationarySensorParameters [1] (Composition association from stationary sensor parameters group)
Public attributes:	(none)

Table 10 — Defining elements of GIG_ImageAccuracySummary class

Description:	Summary of the estimated positional accuracies of positions extracted using this image with this group of parameter values.
Stereotype:	DataType
Class attribute:	Concrete
Inheritance from:	(none)
Association roles:	(none)
Used by:	GIG_ImageParameterValues
Public attributes:	

Attribute name	UML identifier	Data type	Obligation	Maximum occurrence	Attribute description
Circular Error (90%)	CE	Length	M	1	Estimated absolute (external) horizontal error in object positions extracted using this image.
Linear Error (90%)	LE	Length	C [a]	1	Estimated absolute (external) elevation error in object positions extracted using this image with its stereo mate image in this adjusted group.
Horizontal Shear	horizontal Shear	Length	C [b]	1	Estimated (relative internal) horizontal difference between object positions extracted using this image and overlapping image(s) in this adjusted group.
Vertical Shear	verticalSh ear	Length	C [c]	1	Estimated (relative internal) elevation difference between object positions extracted using this image and overlapping image(s) in this adjusted group.

a Condition: At least one stereo mate image is included in this adjusted group.

b Condition: One or more overlapping images, excluding any stereo mate image(s), are included in this adjusted group.

c Condition: One or more overlapping stereo pairs of images are included in this adjusted group.

Table 11 — Defining elements of GIG_AdjustableParameters class

Description:	List of the adjustable parameters in a group of image parameter values, for the adjusted group that includes these parameter values.				
Stereotype:	Type				
Class attribute:	Concrete				
Inheritance from:	GF_FeatureType				
Association roles:	adjustableParameters from GIG_ImageParameterValues [0..*] (Association from group of parameter values using these adjustable parameters)				
Public attributes:	Two attributes inherited from GF_FeatureType, plus:				
Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
Name	name	CharacterString	M	N	Identifier of adjustable parameter.

Table 12 — Defining elements of GIG_ImageParametersStatus class

Description:	Status of this group of image parameter values.				
Stereotype:	CodeList				
Class attribute:	Concrete				
Inheritance from:	(none)				
Association roles:	(none)				
Used by:	GIG_ImageParameterValues				
Public attributes:					
Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
estimated	estimated	CharacterString	C	1	Contains initial estimated values.
adjusted	adjusted	CharacterString	C	1	Contains computed adjusted values, not yet checked
validated	validated	CharacterString	C	1	Contains validated (or checked) adjusted values.
invalid	invalid	CharacterString	C	1	Contains values considered not valid for some reason (but object not yet deleted).
Condition: One and only one of the listed attributes shall be supplied.					

8.3 XML encoding

The GIG_ImageGeometry UML package is encoded in the igmImageGeometry.xsd XML Schema Document, which imports the coordinateOperations.xsd and geometryBasic2d.xsd XML Schema Documents from GML 3.2.0. The ISO 19107, 19109, and 19111 classes used in the UML model are encoded by the GML 3.2 XML elements as listed in Table 3. The contents of the igmImageGeometry.xsd XML Schema Document shall be as specified in the attached file.

Since the ImageParametersStatus is a UML <CodeList> stereotyped class, it is converted into XML Schema following the GML pattern for XML encoding of <CodeList> classes. That is, in igmImageGeometry.xsd, the normal values of this class are NOT encoded in the

XML Schema, but in a default-referenced GML dictionary. That dictionary document is named imageParametersStatusValues.xml, and is also attached to this document.

NOTE The following examples use "999" as a place holder for specific numerical values.

EXAMPLE 1 A simple example XML fragment for an ImageInfo element is:

```
<ImageInfo gml:id="ImageInfo999">
   <gml:identifier codeSpace="IGM">ImageInfo999</gml:identifier> <!--
Include image ID -->
   <imagePixels>TBD</imagePixels> <!-- Reference to image pixels -->
   <imageCRS>urn:ogc:def:crs:OGC:0.0:
ImageCRSpixelCenter:Image999</imageCRS>
   <rows>999</rows>
   <columns>999</columns>
   <transformation
xlink:href="transformationFile999.999.xml#Transformation999.999"/> <!--
Repeat for each adjustment of this image. Reference to concrete element
in AbstractImageParameterValues substitutionGroup -->
</ImageInfo>
```

EXAMPLE 2 A larger example XML fragment for an ImageInfo element, which includes one ImagePosition, is:

```
<ImageInfo gml:id="ImageInfo999">
   <gml:identifier codeSpace="IGM">ImageInfo999</gml:identifier>
   <imagePixels>TBD</imagePixels> <!-- Reference to image pixels -->
   <imageCRS>urn:ogc:def:crs:OGC:0.0:
ImageCRSpixelCenter:Image999</imageCRS>
   <rows>999</rows>
   <columns>999</columns>
   <transformation
xlink:href="transformationFile999.999.xml#Transformation999.999"/> <!--
Repeat for each adjustment of this image, Reference to concrete element
in AbstractImageParameterValues substitutionGroup -->
   <!-- ============================================== -->
   <imagePosition> <!-- Repeat for each point measured in this image --
>
      <ImagePosition gml:id="ImagePosition999">
         <gml:identifier
codeSpace="IGM">ImagePosition999</gml:identifier>
         <point xlink:href="objectPointFile999.xml#ObjectPoint999"/>
<!-- Reference to this ObjectPoint element -->
         <imagePositionStatus
codeSpace="imagePositionStatusValues.xml">measured</imagePositionStatus
>
         <numberMeasurements>999</numberMeasurements> <!-- Optional --
>
         <position>
            <gml:Point gml:id="Point999">
               <gml:pos>999 999</gml:pos>
            </gml:Point>
         </position>
         <positionErrorEstimates>
            <CovarianceMatrix>
               <matrixSize>2</matrixSize>
               <adjustedParameters>row column</adjustedParameters>
               <result>
```



```
                        <DiagonalMatrix>
                            <valueUnit>rowSpacing columnSpacing</valueUnit>
                            <valuesList>0.999 0.999</valuesList>
                        </DiagonalMatrix>
                    </result>
                </CovarianceMatrix>
            </positionErrorEstimates>
        </ImagePosition>
    </imagePosition>
    <!-- =============================================== -->
</ImageInfo>
```

EXAMPLE 3 A simple example XML fragment for a specific ImageParameterValues element, in the AbstractGeneralParameterValue substitutionGroup, is:

```
<ImageParameterValues>
    <igm:image xlink:href="imageInfofile999.xml#ImageInfor999"/> <!--
Reference to ImageInfo element for this image -->
    <igm:inGroup
xlink:href="adjustedGroupFile999.xml#AdjustedGroup999"/> <!-- Reference
to AdjustedGroup element for this group -->
    <igm:imageParametersStatus
codeSpace="../imageParametersStatusValues.xml">adjusted</igm:imageParam
etersStatus>
    <igm:sensorParameterValues
xlink:href="sensorParametersFile999.999.xml#SensorParameters999.999"/>
<!-- Reference to SensorParameterValues element for this image -->
    <igm:adjustableParameters
xlink:href="adjustableParametersFile999.xml#AdjustableParameters999"/>
<!-- Reference to AdjustableParameters element for this image -->
    <!-- =============================================== -->
    <igm:imageAccuracySummary>
        <igm:ImageAccuracySummary>
            <igm:CE uom="m">99.9</igm:CE>
        </igm:ImageAccuracySummary>
    </igm:imageAccuracySummary>
    <!-- =============================================== -->
    <igm:group
xlink:href="frameOperationMethodFile.xml#FrameImageGroup"/> <!--
Reference to correct OperationParameterGroup element -->
    <!-- =============================================== -->
    <igm:imageOrientation>
        <igm:ImageOrientation>
            <igm:sensorPosition>
                <igm:SensorPosition
srsName="urn:ogc:def:crs:IGM:0.0:imageLSRcrs:AA99"
gml:id="SensorPosition999.999">
                    <gml:pos>999 999 999</gml:pos>
                </igm:SensorPosition>
                <igm:operationParameter
xlink:href="frameOperationMethodFile.xml#SensorPosition"/>
            </igm:sensorPosition>
            <igm:sensorAttitude>
                <igm:SensorAttitude
srsName="urn:ogc:def:crs:IGM:0.0:imageLSRcrs:AA99"
gml:id="SensorAttitude999.999">
                    <gml:vector>9.99 9.99 9.99</gml:vector>
                </igm:SensorAttitude>
```



```
            <igm:operationParameter
xlink:href="frameOperationMethodFile.xml#SensorAttitude"/>
          </igm:sensorAttitude>
          <igm:group
xlink:href="frameOperationMethodFile.xml#ImageOrientation"/>
        </igm:ImageOrientation>
    </igm:imageOrientation>
    <!-- =============================================== -->
    <igm:group
xlink:href="frameOperationMethodFile.xml#SensorOrientation"/> <!--
Reference to correct OperationParameterGroup element -->
</ImageParameterValues>
```

EXAMPLE 4 A larger example XML fragment for a specific ImageParameterValues element in the AbstractGeneralParameterValue substitutionGroup, encoded inline in an ObjectImageTransformation element, is:

```
<igm:ObjectImageTransformation gml:id="Transformation999.999">
    <gml:identifier
codeSpace="IGM">Transformation999.999</gml:identifier>
    <gml:scope>Domain of targetCRS</gml:scope>
    <gml:operationVersion>0.0.0</gml:operationVersion>
    <gml:sourceCRS xlink:href="urn:ogc:def:crs:EPSG:6.8:4979"/>
    <gml:targetCRS
xlink:href="urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:TBDimageID"/>
    <gml:method xlink:href="templateFrameOperationMethod1.xml"/>
    <!-- =============================================== -->
    <igm:parameterValue>
        <ImageParameterValues>
            <igm:image xlink:href="imageInfoFile999.xml#ImageInfo999"/>
<!-- Reference to ImageInfo element for this image -->
            <igm:inGroup
xlink:href="adjustedGroupFile999.xml#IAdjustedGroup999"/> <!--
Reference to AdjustedGroup element for this adjusted set of image
parameter values  -->
            <igm:imageParametersStatus
codeSpace="../imageParametersStatusValues.xml">adjusted</igm:imageParam
etersStatus>
            <igm:sensorParameterValues
xlink:href="sensorParametersFile999.999.xml#SensorParameters999.999"/>
<!-- Reference to SensorParameterValues element for this image -->
            <igm:footprint>
                <gml:Polygon srsName="urn:ogc:crs:EPSG:6.0:9999"
gml:id="Polygon999">
                    <gml:exterior>
                        <gml:LinearRing>
                            <gml:pos>999 999</gml:pos>
                            <gml:pos>999 999</gml:pos>
                            <gml:pos>999 999</gml:pos>
                            <gml:pos>999 999</gml:pos>
                        </gml:LinearRing>
                    </gml:exterior>
                    <gml:interior> <!-- Repeated for each void in exterior
ring -->
                        <gml:LinearRing>
                            <gml:pos>999 999</gml:pos>
                            <gml:pos>999 999</gml:pos>
                            <gml:pos>999 999</gml:pos>
```



```
                        <gml:pos>999 999</gml:pos>
                     </gml:LinearRing>
                  </gml:interior>
               </gml:Polygon>
            </igm:footprint>
            <igm:adjustableParameters
xlink:href="adsjustableParametersFile999.xml#AdjustableParameters999"/>
<!-- Reference to AdjustableParameters element for this image -->
            <!-- ============================================== -->
            <igm:imageAccuracySummary>
               <igm:ImageAccuracySummary>
                  <igm:CE uom="m">99.9</igm:CE>
                  <igm:LE uom="m">99.9</igm:LE> <!-- Omit when not
stereoscopic image -->
                  <igm:horizontalShear uom="m">99.9</igm:horizontalShear>
<!-- Omit when not applicable -->
                  <igm:verticalShear uom="m">99.9</igm:verticalShear> <!-
- Omit when not stereoscopic image -->
               </igm:ImageAccuracySummary>
            </igm:imageAccuracySummary>
            <!-- ============================================== -->
            <igm:sensorPosition>
               <gml:Point gml:id="Point999" srsName="LSR">
                  <gml:pos>999 999 999</gml:pos>
               </gml:Point>
               <igm:operationParameter
xlink:href="frameOperationMethodFile.xml#SensorPosition"/> <!--
Reference to correct OperationParameter element -->
            </igm:sensorPosition>
            <!-- ============================================== -->
            <igm:sensorAttitude>
               <gml:Vector gml:id="Angles999" srsName="AnglesInLSR">
                  <gml:pos>9.99 9.99 9.99</gml:pos>
               </gml:Vector>
               <igm:operationParameter
xlink:href="frameOperationMethodFile.xml#SensorAttitude"/> <!--
Reference to correct OperationParameter element -->
            </igm:sensorAttitude>
            <!-- ============================================== -->
            <igm:group
xlink:href="frameOperationMethodFile.xml#ImageOrientation"/> <!--
Reference to correct OperationParameterGroup element -->
         </ImageParameterValues>
      </igm:parameterValue>
</igm:ObjectImageTransformation>
```

EXAMPLE 5 A example XML fragment for an AdjustableParameters element is:

```
<AdjustableParameters gml:id="AdjustableParameters999">
   <gml:description>TBD</gml:description>
   <gml:identifier
codeSpace="IGM">AdjustableParameters999</gml:identifier>
   <parameters>TBD999 TBD999 TBD999 TBD999 TBD999 TBD999</parameters>
</AdjustableParameters>
```

9 Sensor calibration

9.1 Introduction

The sensor calibration part of the Image geopositioning metadata GML Application Schema supports recording imaging sensor calibration parameter values, including multiple adjustments of interior orientation data. The sensor calibration parameter values are constant for many images collected by one image sensor, and are not frequently adjusted, so are recorded separate from the image parameter values specified above. This sensor calibration can be for many possible image geometry (or sensor) models that can be encoded using extensions of this Application Schema.

9.2 UML model

The UML class diagram for this package is shown in Figure 6, together with the classes that are inherited from. The three new classes defined in this GSC_SensorCalibration package are described in Table 13 through Table 15.

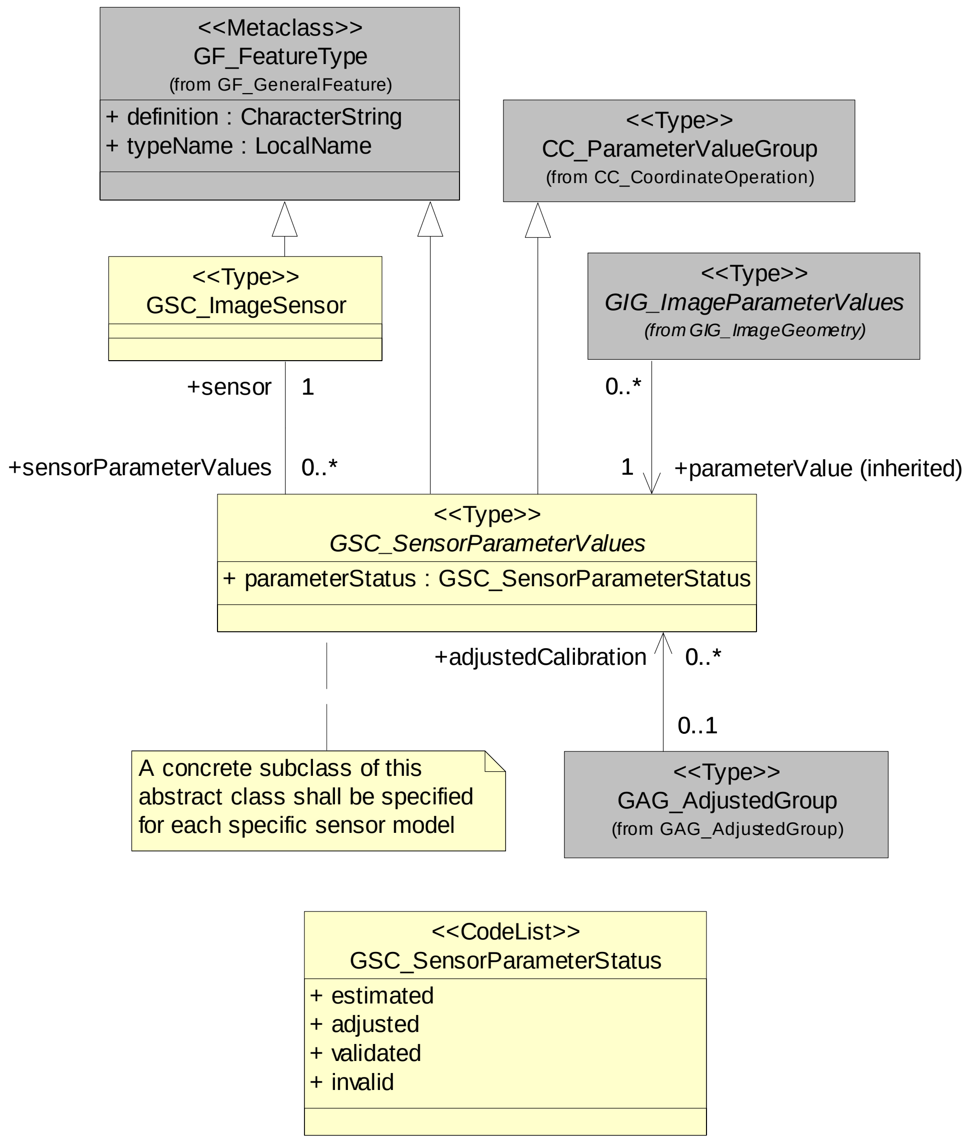

Figure 6 — GSC_SensorCalibration package UML class diagram

This UML class diagram shows how multiple versions of sensor calibration parameter values can be recorded for each image sensor, with each version associated with the one AdjustedGroup that produced it. That is, multiple SensorParameterValues objects are associated with the ImageSensor object for one imaging sensor. The SensorParameterValues abstract class records the parameter values for any sensor model.

The details specific to each specific sensor model are recorded as concrete subclasses of the SensorParameterValues abstract class, which shall be specified in extensions of this Application Schema. These classes should share more classes for common aspects (also not shown here). Wherever relevant, these specific image geometry models should build upon and adapt ISO 19130 (Sensor data model for imagery and gridded data).

NOTE The SensorParameterValues abstract class is a subclass of both CC_ParameterValuesGroup, since it contains a group of parameter values, and of GF_FeatureType, to allow it to be associated by reference from the AdjustedGroup and the ImageSensor classes.

The GSC_SensorCalibration package specializes the GF_FeatureType package in ISO 19109, by defining more-specific subclasses. By defining the GSC_ImageSensor class as a feature, identification metadata is inherited and additional feature properties for recording needed metadata can be easily added.

This package also specializes the CC_CoordinateOperation package in ISO 19111, by defining more-specific subclasses of CC_ParameterValueGroup. By defining the GSC_SensorParameterValues class as a parameter value group, individual parameters and subsidiary parameter groups can be defined for different image sensor models.

Table 13 — Defining elements of GSC_ImageSensor class

Description:	Calibration (interior orientation) data for one imaging sensor.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GF_FeatureType
Association roles:	sensorParameterValues to GSC_SensorParameterValues [0..*] (Association to group of parameter values determined for this image sensor)
	sensor from GSC_SensorParameterValues [1] (Association from group of parameter values determined for this image sensor)
Public attributes:	Two attributes inherited from GF_FeatureType

Table 14 — Defining elements of GSC_SensorParameterValues class

Description:	Generic group of calibrated parameter values for one imaging sensor and sensor configuration, including most interior orientation parameters.
Stereotype:	Type
Class attribute:	Abstract
Inheritance from:	CC_ParameterValueGroup and GF_FeatureType
Association roles:	sensor to GSC_ImageSensor [1] (Association to image sensor for this group of parameter values)
	group to CC_OperationParameterGroup [1] (Inherited association to operation parameter group for this group of sensor parameter values)
	parameterValue from GIG_ImageParameterValues [1] (Association from image parameter adjustment to calibrated imaging sensor parameter used)
	adjustedCalibration from GAG_AdjustedGroup [0..1] (Association from adjusted group that adjusted this group of sensor parameters)
	sensorParameterValues from GSC_ImageSensor [0..*] (Association from image sensor that uses this group of sensor parameter values)
Public attributes:	No attributes inherited from CC_ParameterValueGroup, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum occurrence	Attribute description
Sensor Parameter Status	parameterStatus	GSC_SensorParameterStatus	M	1	Status of this group of sensor parameter values.

Table 15 — Defining elements of GSC_SensorParameterStatus class

Description:	Status of this group of sensor parameter values.				
Stereotype:	CodeList				
Class attribute:	Concrete				
Inheritance from:	(none)				
Association roles:	(none)				
Used by:	GIG_SensorParameterValues				
Public attributes:					
Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
estimated	estimated	CharacterString	C	1	Contains initial estimated values.
adjusted	adjusted	CharacterString	C	1	Contains computed adjusted values, not yet checked
validated	validated	CharacterString	C	1	Contains validated (or checked) adjusted values.
invalid	invalid	CharacterString	C	1	Contains values considered not valid for some reason (but object not yet deleted).
Condition: One and only one of the listed attributes shall be supplied.					

9.3 XML encoding

The GIG_SensorCalibration UML package is encoded in the igmSensorCalibration.xsd XML Schema Document, which imports the coordinateOperations.xsd XML Schema Document from GML 3.2.0. The ISO 19107, 19109, and 19111 classes used in the UML model are encoded by the GML 3.2 XML elements as listed in Table 3. The contents of the igmSensorCalibration.xsd XML Schema Document shall be as specified in the attached file.

Since the SensorParameterStatus is a UML <CodeList> stereotyped class, it is converted into XML Schema following the GML pattern for XML encoding of <CodeList> classes. That is, in igmSensorCalibration.xsd, the normal values of this class are NOT encoded in the XML Schema, but in a default-referenced GML dictionary. That dictionary document is named sensorParametersStatusValues.xml and is also attached to this document.

NOTE The following examples use "999" as a place holder for specific numerical values.

EXAMPLE 1 A simple example XML fragment for a specific ImageSensor element is:

```
<ImageSensor gml:id="imageSensor999">
   <gml:description>TBD</gml:description>
   <gml:identifier codeSpace="IGM">imageSensor999</gml:identifier>
   <sensorParameterValues
xlink:href="sensorParameterValuesFile999.999.xml#SensorParameterValues9
99.999"/>
</ImageSensor>
```


EXAMPLE 2 A simple example XML fragment for a specific SensorParameterValues element in the AbstractGeneralParameterValue substitutionGroup is:

```
<SensorParameterValues gml:id="SensorParameters999.999">
   <gml:identifier
codeSpace="IGM">SensorParameters999.999</gml:identifier>
   <igm:sensor> <!-- Association to ImageSensor element for image -->
      <igm:ImageSensor gml:id="ImageSensor999">
         <gml:description>TBD</gml:description>
         <gml:identifier
codeSpace="IGM">ImageSensor999</gml:identifier>
      </igm:ImageSensor>
   </igm:sensor>
   <igm:sensorParameterStatus
codeSpace="../imageParametersStatusValues.xml">initial</igm:sensorParam
eterStatus>
   <!-- ============================================== -->
   <focalLength>
      <FocalLength>
         <gml:value uom="mm">999</gml:value>
         <igm:operationParameter
xlink:href="frameImageMethodFile.xml#FocalLength"/> <!-- Reference to
correct OperationParameter element  -->
      </FocalLength>
   </focalLength>
   <!-- ============================================== -->
   <ppOffsetPosition>
      <PPoffset>
         <gml:valueList uom="mm">0.999 0.999</gml:valueList>
         <igm:operationParameter
xlink:href="frameImageMethodFile.xml#PPoffset"/> <!-- Reference to
correct OperationParameter element  -->
      </PPoffset>
   </ppOffsetPosition>
   <!-- ============================================== -->
   <igm:group
xlink:href="frameImageMethodFile.xml#OpticalPerspective"/> <!--
Reference to correct OperationParameterGroup element  -->
</SensorParameterValues>
```

10 Adjusted group

10.1 Introduction

The adjusted group part of the Image geopositioning metadata GML Application Schema supports recording multiple adjustments of image orientation data for groups of images that are adjusted together. The adjusted image orientation parameters can be for many possible image geometry (or sensor) models that can be encoded using extensions of this Application Schema.

10.2 UML model

The UML class diagram for the GAG_AdjustedGroup package is shown in Figure 4, together with the ISO 191XX UML classes that are directly inherited from. The new classes defined in this GAG_AdjustedGroup package are described in Table 16 through Table 19.

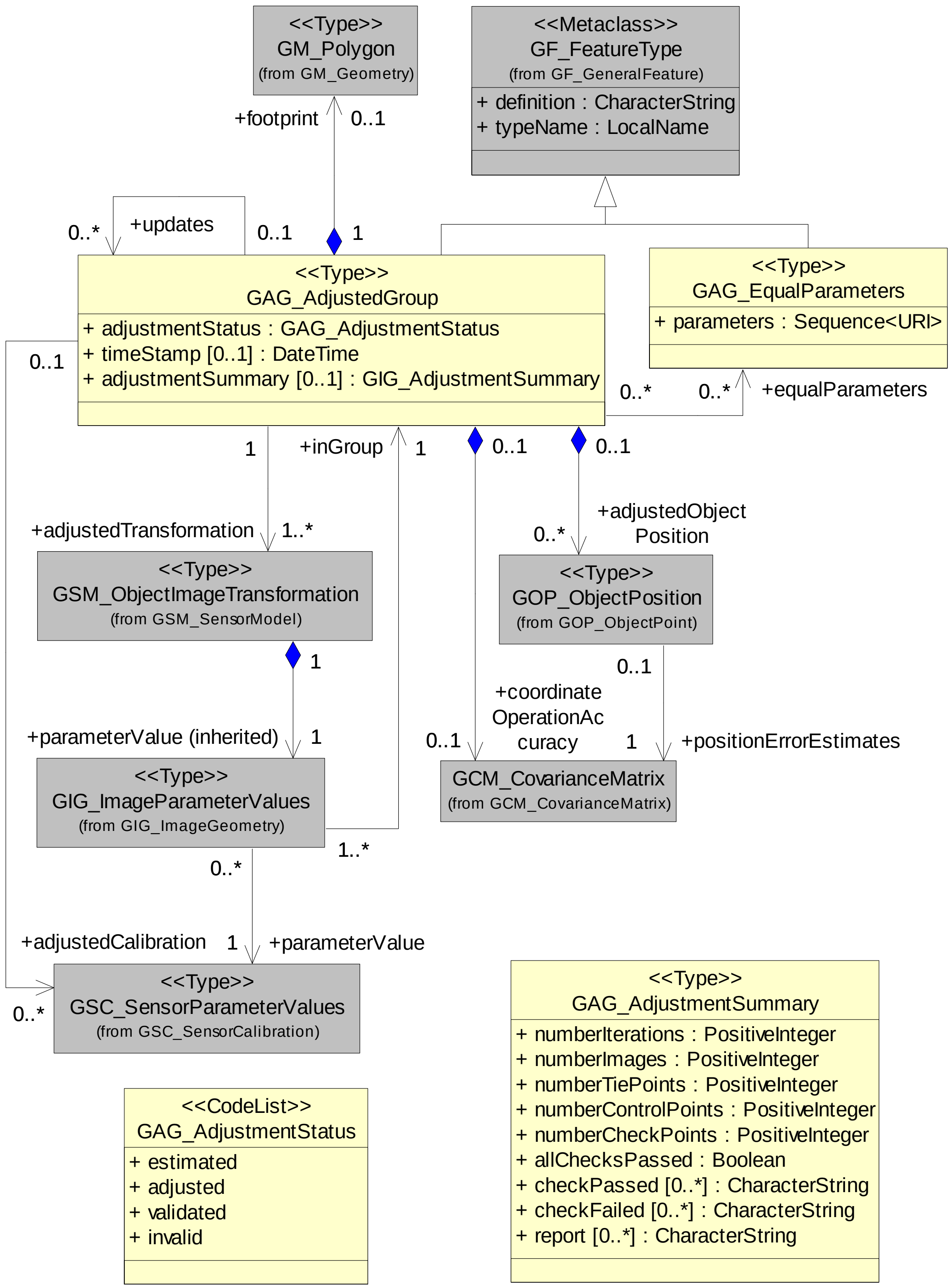

Figure 7 — GAG_AdjustedGroup package UML class diagram

This UML class diagram shows how metadata for multiple groups images adjusted together can be recorded. Multiple ObjectImageTransformation and ImageParameterValues objects can be associated with an AdjustedGroup object for one adjustment (or triangulation). As needed for different image types and purposes, different sensor models can be used by the images associated with the AdjustedGroup for one adjustment. Similarly, different image parameters can be adjusted in different ImageParameterValues objects.

The AdjustedGroup concrete class records the status of an adjustment, and has an association to the CovarianceMatrix object that records all the coordinateOperationAccuracy estimates resulting from that adjustment. Each AdjustedGroup object is associated to the ObjectPosition objects that were adjusted or computed by this adjustment. Each AdjustedGroup object also has an optional association to a Polygon giving the approximate footprint for that adjustment. Since the GM_Polygon is used, any holes in that polygon can be recorded.

Also, each AdjustmentGroup object can have associations to zero or more EqualParameters objects, which each identify two or more otherwise adjustable parameters that were considered to have the same values during that adjustment. Each EqualParameters object can be associated from multiple AdjustmentGroup objects which have the same set of equal parameters (as is common).

The AdjustedGroup package specializes the GF_FeatureType package in ISO 19109, by defining specific subclasses. By defining the AdjustedGroup class as a feature collection, identification metadata is inherited and additional properties for recording needed metadata can be easily added. By defining the EqualParameters as features, identification metadata is inherited and additional properties for recording needed metadata can be easily added.

Table 16 — Defining elements of GAG_AdjustedGroup class

Description:	Definition of a group of one or more images that were adjusted as a group. [a]
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GF_FeatureType
Association roles:	adjustedTransformation to GSC_ObjectImageTransformation [1..*] (Association to object-image transformation determined by this adjusted group) adjustedObjectPosition to GOP_ObjectPosition [0..*] (Composition association to object positions determined by this adjusted group) adjustedCalibration to GSC_SensorParameterValues [0..*] (Association to group of sensor parameters determined by this adjusted group) coordinateOperationAccuracy to GCM_CovarianceMatrix [1] [c] (Composition (TBR) association to error estimates for all the image parameter and position values from this adjustment) footprint to GM_Polygon [0..1] (Composition association to approximate footprint of this adjusted group) equalParameters to GAG_EqualParameters [0..*] (Association to list of equal parameters for this adjusted group) updates to GAG_AdjustedGroup [0..*] (Association to previous adjusted group updated by this adjusted group) updates from GAG_AdjustedGroup [0..1] (Association from adjusted group that updated by this adjusted group) inGroup from GIG_ImageParameterValues [0..1] (Association from group of image parameter values adjusted by this adjusted group)
Public attributes:	Two attributes inherited from GF_FeatureType, plus:

Attribute name	UML identifier	Data type	Obligation	Maximum occurrence	Attribute description
Adjustment status	adjustmentStatus	GAG_AdjustmentStatus	M	1	Status of this adjustment of this group of images.
Time stamp	timeStamp	DateTime	C [b]	1	Time when this group adjustment was performed or last changed.
Adjustment Summary	adjustmentSummary	GAG_AdjustmentSummary	C [b]	1	Summary of results of this adjustment.

a A group of images can be either: 1) a strip of overlapping images collected in one pass, 2) a block of images covering a roughly rectangular area, 3) a group of blocks of images, or 4) a single image (for example, being adjusted to control points).

b Condition: Include after adjustment performed.

c The association “coordinateOperationAccuracy to GCM_CovarianceMatrix” is from GAG_AdjustedGroup instead of from GIG_ImageParameterValues or GSM_ObjectImageTransformation in order to support correlations between images.

NOTE 1 Each GM_Polygon object references the geodetic coordinate reference system of this polygon.

Table 17 — Defining elements of GAG_AdjustmentSummary class

Description:	Summary of results of this adjustment.
Stereotype:	DataType
Class attribute:	Concrete
Inheritance from:	(none)
Association roles:	(none)
Used by:	GAG_AdjustedGroup
Public attributes:	

Attribute name	UML identifier	Data type	Obligation	Maximum occurrence	Attribute description
Number of Iterations	numberIterations	PositiveInteger	M	1	Number of adjustment computation iterations actually executed.
Number of Images	numberImages	PositiveInteger	M	1	Number of images whose image parameters were adjusted.
Number of Tie Points	numberTiePoints	PositiveInteger	M	1	Number of tie points whose image position coordinates were used, and object positions estimated.
Number of Control Points	numberControlPoints	PositiveInteger	M	1	Number of control points whose object and image position coordinates were used, and object positions estimated.
Number of Check Points	numberCheckPoints	PositiveInteger	M	1	Number of check points whose object positions were estimated using image position coordinates with final image parameters (but were not used in adjustment).
All Checks Passed	allChecksPassed	Boolean	M	1	Adjustment results passed all required tests (or checks).
Check Passed	checkPassed	Character String	C [a]	N	Adjustment results passed identified test.
Check Failed	checkFailed	Character String	C [b]	N	Adjustment results failed identified test.
Report	report	Character String	O	N	Report describing adjustment results (without specified contents). [c]

a Condition: At least one identified test was passed by adjustment results.

b Condition: At least one identified test was failed by adjustment results.

c A report may include any (currently unspecified) information that might be useful to the client.

Table 18 — Defining elements of GAG_EqualParameters class

Description:	List of adjustable parameters in an adjusted group that are constrained to have equal values.				
Stereotype:	Type				
Class attribute:	Concrete				
Inheritance from:	GF_FeatureType				
Association roles:	equalParameters from GAG_AdjustedGroup [0..*] (Association from adjusted group using this list of equal parameters)				
Public attributes:	Two attributes inherited from GF_FeatureType, plus:				
Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
Name	name	Character String	M	N	Identifier of adjustable parameter constrained to have the same value as all others in this list.

Table 19 — Defining elements of GAG_AdjustmentStatus class

Description:	Status of this adjustment of this group of images.				
Stereotype:	CodeList				
Class attribute:	Concrete				
Inheritance from:	(none)				
Association roles:	(none)				
Used by:	GAG_AdjustedGroup				
Public attributes:					
Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
estimated	estimated	Character String	C	1	Contains initial estimated values.
adjusted	adjusted	Character String	C	1	Contains computed adjusted values, not yet checked
validated	validated	Character String	C	1	Contains validated (or checked) adjusted values.
invalid	invalid	Character String	C	1	Contains values considered not valid for some reason (but object not yet deleted).
Condition: One and only one of the listed attributes shall be supplied.					

10.3 XML encoding

The GAG_AdjustedGroup UML package is encoded in the igmAdjustedGroup.xsd XML Schema Document, which imports the coordinateOperations.xsd and geometryBasic2d.xsd XML Schema Document from GML 3.2.0. The ISO 19107, 19109, and 19111 classes used in the UML model are encoded by the GML 3.2 XML elements as listed in Table 3. The contents of the igmAdjustedGroup.xsd XML Schema Document shall be as specified in the attached file.

Since the AdjustmentStatus is a UML <CodeList> stereotyped classes, it is converted into XML Schema following the GML pattern for XML encoding of <CodeList> classes. That is, in igmAdjustedGroup.xsd, the normal values of these two classes are NOT encoded in the

XML Schema, but in a default-referenced GML dictionary. This dictionary document is named adjustmentStatusValues.xml, and is also attached to this document.

NOTE The following examples use “999” as a place holder for specific numerical values.

EXAMPLE 1 An example XML fragment for an AdjustedGroup element is:

```
<AdjustedGroup gml:id="AdjustedGroup999">
   <gml:identifier codeSpace="IGM">AdjustedGroup999</gml:identifier>
   <adjustmentStatus
codeSpace="../adjustmentStatusValues.xml">adjusted</adjustmentStatus>
   <timeStamp>2007-01-01T00:00:00</timeStamp>
   <adjustedTransformation
xlink:href="transformationfFle999.999.xml#ITransformation999.999"/> <!-
- Repeat for each image in adjusted group. Reference to concrete
element in AbstractImageParameterValues substitutionGroup -->
   <!-- =============================================== -->
   <adjustedObjectPosition> <!-- Repeated for each adjusted object
position -->
      <ObjectPosition gml:id="ObjectPosition999">
         <gml:identifier
codeSpace="IGM">ObjectPosition999</gml:identifier>
         <point xlink:href="objectPointFile999.xml#ObjectPoint999"/>
<!-- Reference to this ObjectPoint -->
         <objectPositionStatus
codeSpace="../objectPositionStatusValues.xml">estimated</objectPosition
Status>
         <dimension>3</dimension>
         <position>
            <gml:Point srsName="urn:ogc:crs:EPSG:6.0:XXXX"
gml:id="Point999">
               <gml:pos>999 999 999</gml:pos>
            </gml:Point>
         </position>
         <positionErrorEstimates>
            <CovarianceMatrix>
               <matrixSize>3</matrixSize>
               <adjustedParameters>X Y Z</adjustedParameters>
               <result>
                  <CompleteMatrix>
                     <valueUnit >m m m</valueUnit>
                     <valuesList>0.999 0.999 0.999 0.999 0.999
0.999</valuesList>
                  </CompleteMatrix>
               </result>
            </CovarianceMatrix>
         </positionErrorEstimates>
      </ObjectPosition>
   </adjustedObjectPosition>
   <!-- =============================================== -->
   <coordinateOperationAccuracy>
      <CovarianceMatrix>
         <matrixSize>6</matrixSize>
         <adjustedParameters>X Y Z Kappa Phi
Omega</adjustedParameters>
         <result>
            <BandedMatrix>
               <valueUnit>m m m r r r</valueUnit>
```



```
                <valuesList row="1">0.999 0.999 0.999</valuesList>
                <valuesList row="2">0.999 0.999 0.999</valuesList>
                <valuesList row="3">0.999 0.999 0.999</valuesList>
                <valuesList row="4">0.999 0.999 0.999</valuesList>
                <valuesList row="5">0.999 0.999</valuesList>
                <valuesList row="6">0</valuesList>
             </BandedMatrix>
          </result>
       </CovarianceMatrix>
    </coordinateOperationAccuracy>
    <!-- ============================================== -->
    <adjustmentSummary>
       <AdjustmentSummary>
          <numberIterations>999</numberIterations>
          <numberImages>999</numberImages>
          <numberTiePoints>999</numberTiePoints>
          <numberControlPoints>999</numberControlPoints>
          <numberCheckPoints>999</numberCheckPoints>
          <allChecksPassed>true</allChecksPassed>
       </AdjustmentSummary>
    </adjustmentSummary>
    <!-- ============================================== -->
</AdjustedGroup>
```

EXAMPLE 2 An example XML fragment for an EqualParameters element is:

```
<EqualParameters gml:id="EqualParameters999">
   <gml:description>TBD</gml:description>
   <gml:identifier codeSpace="IGM">EqualParameters999</gml:identifier>
   <parameters>TBD999 TBD999 TBD999 TBD999 TBD999 TBD999</parameters>
</EqualParameters>
```

11 Object point

11.1 Introduction

The object point part of the Image geopositioning metadata GML Application Schema records object (or ground) point metadata, including position coordinates in one of more images and optional object (or ground) coordinates. These object points can be tie points, control points, or check points. A control point or check point has a measured position with position error statistics in one or more images, and a known position with error statistics in some geodetic Coordinate Reference System (CRS). A tie point has a measured position with error statistics in two or more images, but not a known position in any geodetic CRS. All types of points can have the estimated object position computed from the measured positions in two or more images.

11.2 UML model

The UML class diagram for this package is shown in Figure 8 and Figure 9, together with the classes that are inherited from. The new classes defined in this GOP_ObjectPoint package are described in Table 20 through Table 27.

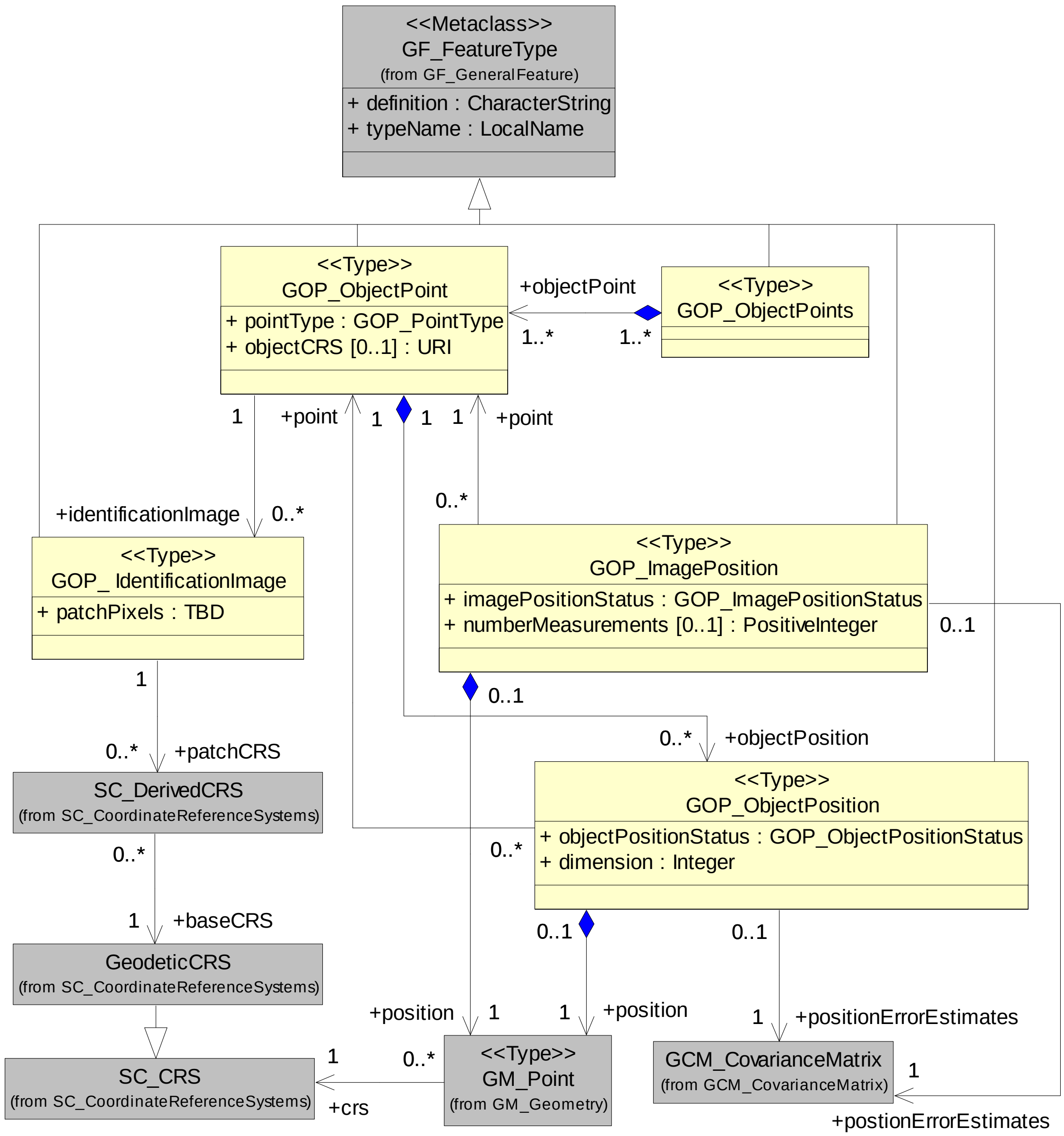

Figure 8 — GOP_ObjectPoint package UML class diagram, part 1

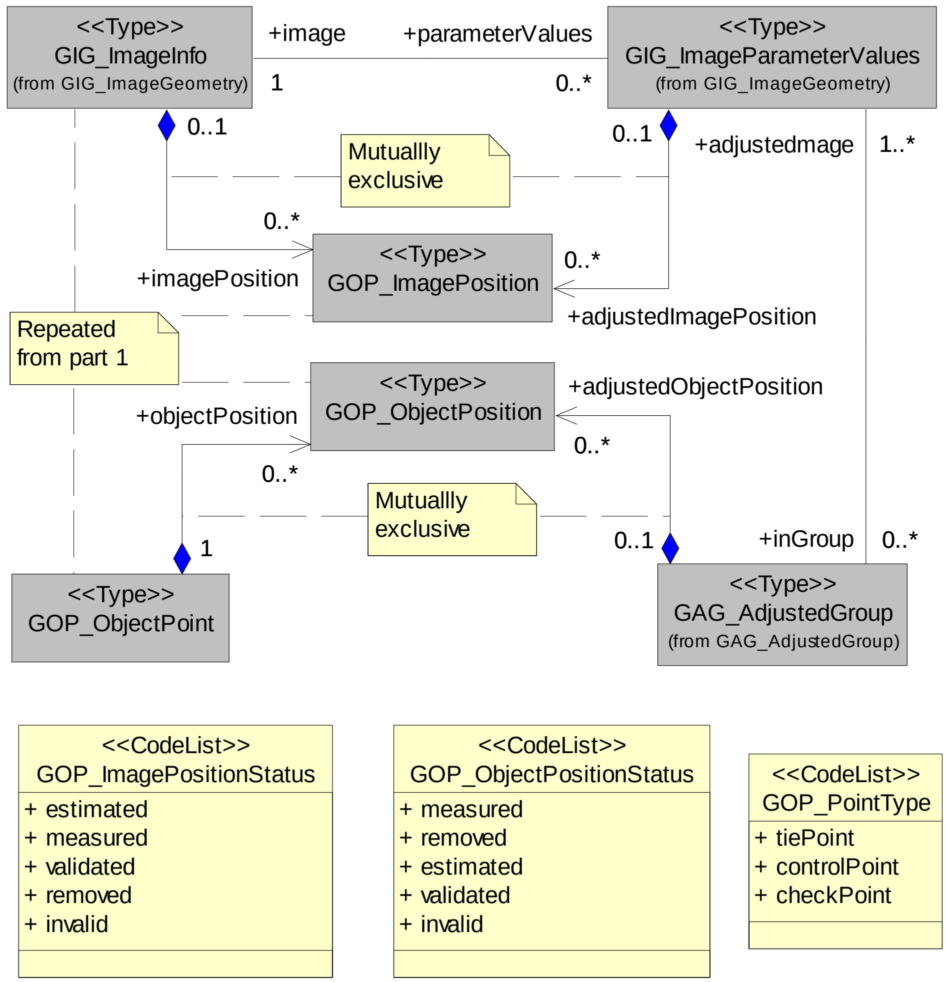

Figure 9 — GOP_ObjectPoint package UML class diagram, part 2

Part 1 of this UML class diagram shows how flexible object (or ground) point metadata is structured. An ObjectPoint object records the identification and type of each recorded point. Each ObjectPoint object is associated to zero or more ObjectPosition objects that record measured object position coordinates for this point, each using one GM_Point object. Similarly, each ObjectPoint object is associated from zero or more ImagePosition objects that record position coordinates of this point measured in different images, each using one GM_Point. Also, each ObjectPoint is associated to zero or more IdentificationImage objects that record image patches for use in point identification within another image.

Both ObjectPosition and ImagePosition objects are associated with the CovarianceMatrix that records position error estimates. Each GM_Point can reference its Coordinate Reference System (SC_CRS), when that CRS is not referenced by the ObjectPoint or ImageInfo class. More specifically, each GM_Point for an ObjectPosition references the GeodeticCRS for that point position, unless it is referenced by the associated ObjectPoint. Each GM_Point for an ImagePosition does NOT reference the ImageCRS, since it is referenced by the ImageInfo for all positions in that image. Each IdentificationImage references the image patch plus the SC_DerivedCRS for the image patch grid. That SC_DerivedCRS references its baseCRS, which is always a GeodeticCRS.

Part 1 of this UML class diagram also shows an ObjectPoints class that contains one or more ObjectPoint objects. This ObjectPoints allows combining multiple ObjectPoint objects in one XML-encoded file or data package.

Part 2 of this UML class diagram shows how ImagePosition and ObjectPosition objects are associated with ImageInfo, ImageParameterValues, and AdjustedGroup objects. The ImagePosition objects associated from an ImageInfo object record all the positions measured in that image. The ImagePosition objects associated from an ImageParameterValues object record all the positions computed in that image using that group of image parameter values. The ObjectPosition objects associated from an AdjustedGroup object record all the positions computed using the image parameters used by that adjusted group.

The GOP_ObjectPoint package specializes the GF_FeatureType package in ISO 19109, by defining more-specific subclasses. The GOP_ObjectPoints and GOP_ObjectPoint classes are feature collections, and the GOP_ImagePosition, GOP_ObjectPosition, and GOP_IdentificationImage classes are considered features. By subclassing GF_FeatureType, identification metadata is inherited and additional feature properties for recording needed metadata can be easily added. This package also uses the GM_Point class from ISO 19107, avoiding new definitions of point positions.

Table 20 — Defining elements of GOP_ObjectPoints class

Description:	Group of one or more object points.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GF_FeatureType
Association roles:	objectPoint to GOP_ObjectPoint [1..*] (Composition association to object point in this group)
Public attributes:	Two attributes inherited from GF_FeatureType

Table 21 — Defining elements of GOP_ObjectPoint class

Description:	Description of one object (or ground) point, including its identification.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GF_FeatureType
Association roles:	objectPosition to GOP_ObjectPosition [0..*] [a] (Composition association to measured object position for this point) identificationImage to GOP_IdentificationImage [0..*] [b] (Association to identification image for this object point) point from GOP_ObjectPosition [0..1] (Association from object position for this point) point from GOP_ImagePosition [0..*] (Association from image position of this point) objectPoint from GOP_ObjectPoints [1..*] (Composition association from group of abject points)
Public attributes:	Two attributes inherited from GF_FeatureType, plus:

Attribute name	**UML identifier**	**Data type**	**Oblig ation**	**Maximum occurrence**	**Attribute description**
Object Point Type	pointType	GOP_ObjectPoi ntType	M	1	Type of this object point.
Object CRS	objectCRS	URI	O	1	Reference to CRS for all associated object positions.

a Multiple GOP_ObjectPosition objects can be associated to record multiple measured positions.

b Multiple GOP_IdentificationImage objects can be associated to identify one object point, when those image patches have significantly different object (or ground) resolutions or other image appearance differences.

Table 22 — Defining elements of GOP_ImagePosition class

Description:	Measured or computed image position for an object point.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GF_FeatureType
Association roles:	position to GM_Point [1] (Composition association to image position coordinates) positionErrorEstimates to GCM_CovarianceMatrix [1] (Association to absolute position error estimates for this image position) point to GOP_ObjectPoint [0..*] (Association to object point identifying this image position) imagePosition from GIG_ImageInfo [0..1] [a] (Composition association from image in which this position was measured) adjustedImagePosition from GIG_ImageParameterValue [0..1] [a] (Composition association from parameter group that adjusted this position)
Public attributes:	Two attributes inherited from GF_FeatureType, plus:

Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
Image Position Status	imagePositionStatus	GSC_ImagePositionStatus	M	1	Status of this image position.
Number of Measurements	numberMeasurements	PositiveInteger	O	1	Number of measurements averaged to produce recorded position.

a Each ImagePosition instance shall be associated from either one ImageInfo instance or one ImageParameterValue instance, but not both. Each ImagePosition instance associated from ImageInfo should have an Image Position Status value of “measured”, and shall not have the value “estimated”. Each ImagePosition instance associated from ImageParameterValue should have an Image Position Status value of “estimated”, shall not have the value “measured”.

Table 23 — Defining elements of GOP_ObjectPosition class

Description:	Object (or ground) position for an object point.				
Stereotype:	Type				
Class attribute:	Concrete				
Inheritance from:	GF_FeatureType				
Association roles:	point to GOP_ObjectPoint [1] (Association to description of this object point) position to GM_Point [1] (Composition association to object position coordinates) positionErrorEstimates to GCM_CovarianceMatrix [1] (Association to position error estimates for this object position, including absolute errors of this object position and errors relative to other object positions) objectPosition from GOP_ObjectPoint [0..*] (Composition association from object point that has this measured object position) adjustedObjectPosition from GAG_AdjustedGroup [0..1] [a] (Composition association from adjustment group that estimated this object position)				
Public attributes:	Two attributes inherited from GF_FeatureType, plus:				
Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
Object Position Status	ObjectPositionStatus	GSC_ObjectPositionStatus	M	1	Status of this object position.
Dimension	dimension	Integer	M	1	Number of object position coordinates that are measured or computed. [b]
a Each ObjectPosition instance associated from AdjustedGroup should have an Ground Position Status value of “estimated”, and shall not have the value “measured”. b For a control or check point, the value "3" shall be used for a 3D point, "2" shall be used for a horizontal point, and "1" shall be used for a vertical point. For a tie point between two or more images, "3" shall be used.					

Table 24 — Defining elements of GOP_IdentificationImage class

Description:	Image patch identifying one object point, located at the center of this patch.				
Stereotype:	Type				
Class attribute:	Concrete				
Inheritance from:	GF_FeatureType				
Association roles:	patchCRS to SC_DerivedCRS [1] (Association to grid CRS for this image patch) [a] identification from GOP_ObjectPoint [0..*] (Association from object point that uses this image patch for identification)				
Public attributes:	Two attributes inherited from GF_FeatureType, plus:				
Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
Patch Pixels	patchPixels	TBD	M	1	Image pixels in identification image patch.
a This DerivedCRS is used for a pixel grid that is defined in a geodetic CRS that is the baseCRS of this DerivedCRS.					

Table 25 — Defining elements of GOP_ObjectPointType class

Description:	Type of this object point.				
Stereotype:	CodeList				
Class attribute:	Concrete				
Inheritance from:	(none)				
Association roles:	(none)				
Used by:	GOP_ObjectPoint				
Public attributes:					
Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
Tie Point	tiePoint	CharacterString	C	1	Point with measured position in one or more images, used to geoposition those images.
Control Point	controlPont	CharacterString	C	1	Point with known object position and measured image position(s), used to geoposition one or more images.
Check Point	checkPoint	CharacterString	C	1	Point with known object position and measured image position(s), used to check the geopositioning of one or more images.
Condition: One and only one of the listed attributes shall be supplied.					

Table 26 — Defining elements of GOP_ImagePositionStatus class

Description:	Status of this image position.				
Stereotype:	CodeList				
Class attribute:	Concrete				
Inheritance from:	(none)				
Association roles:	(none)				
Used by:	GOP_ImagePosition				
Public attributes:					
Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
estimated	estimated	CharacterString	C	1	Contains position estimated by computation.
measured	measured	CharacterString	C	1	Contains position measured in image, not yet checked.
validated	validated	CharacterString	C	1	Contains validated (or checked) measured position.
removed	removed	CharacterString	C	1	Contains position automatically detected as having a blunder, and removed from use in later triangulation computation iterations.
invalid	invalid	CharacterString	C	1	Contains position considered not valid for some reason (but object not yet deleted).
Condition: One and only one of the listed attributes shall be supplied.					

Table 27 — Defining elements of GOP_ObjectPositionStatus class

Description:	Status of this object position.				
Stereotype:	CodeList				
Class attribute:	Concrete				
Inheritance from:	(none)				
Association roles:	(none)				
Used by:	GOP_ObjectPosition				
Public attributes:					
Attribute name	**UML identifier**	**Data type**	**Obligation**	**Maximum occurrence**	**Attribute description**
measured	measured	CharacterString	C	1	Contains measured and checked position.
removed	removed	CharacterString	C	1	Contains position automatically detected as having a blunder, and removed from use in later triangulation computation iterations.
estimated	estimated	CharacterString	C	1	Contains position estimated by computation.
validated	validated	CharacterString	C	1	Contains validated (or checked) estimated position.
invalid	invalid	CharacterString	C	1	Contains position considered not valid for some reason (but object not yet deleted).
Condition: One and only one of the listed attributes shall be supplied.					

11.3 XML encoding

The GOP_ObjectPoint UML package is encoded in the igmObjectPoint.xsd XML Schema Document, which imports the coordinateReferenceSystems.xsd XML Schema Document from GML 3.2.0. The ISO 19107 and 19109 classes used in the UML model are encoded by the GML 3.2 XML elements as listed in Table 3. The contents of the igmObjectPoint.xsd XML Schema Document shall be as specified in the attached file.

Since the ObjectPointType, ImagePositionStatus, and ObjectPositionStatus are UML <CodeList> stereotyped classes, they are converted into XML Schema following the GML pattern for XML encoding of <CodeList> classes. That is, in igmObjectPoint.xsd, the normal values of these three classes are NOT encoded in the XML Schema, but in default-referenced GML dictionaries. These dictionary documents are named objectPointTypeValues.xml, imagePositionStatusValues.xml, and objectPositionStatusValues.xml, and are also attached to this document.

NOTE The following examples use “999” as a place holder for specific numerical values.

EXAMPLE 1 An example XML fragment for an ImagePosition is:

```
<ImagePosition gml:id="ImagePosition999.999">
   <gml:identifier
codeSpace="IGM">ImagePosition999.999</gml:identifier>
   <point xlink:href="objectPointFile999.xml#ObjectPoint999"/> <!--
Reference to this object point -->
```



```
    <imagePositionStatus
codeSpace="../imagePostionStatusValues.xml">measured</imagePositionStat
us>
    <!-- ================================================ -->
    <position>
        <gml:Point srsName="OGC:0.0:ImageCRSpixelCenter:TBDimageID"
gml:id="Point999">
            <gml:pos>999 999</gml:pos>
        </gml:Point>
    </position>
    <!-- ================================================ -->
    <positionErrorEstimates>
        <CovarianceMatrix>
            <matrixSize>2</matrixSize>
            <adjustedParameters>row column</adjustedParameters>
            <result>
                <DiagonalMatrix>
                    <valueUnit>rowSpacing columnSpacing</valueUnit>
                    <valuesList>0.999 0.999</valuesList>
                </DiagonalMatrix>
            </result>
        </CovarianceMatrix>
    </positionErrorEstimates>
    <!-- ================================================ -->
</ImagePosition>
```

EXAMPLE 2   A simple example XML fragment for an ObjectPosition is:

```
<ObjectPosition gml:id="ObjectPosition999.999">
    <gml:identifier
codeSpace="IGM">ObjectPosition999.999</gml:identifier>
    <point xlink:href="objectPointFile999.xml#ObjectPoint999"/>
    <objectPositionStatus
codeSpace="../objectPositionStatusValues.xml">computed</objectPositionS
tatus>
    <dimension>3</dimension>
    <!-- ================================================ -->
    <position>
        <gml:Point gml:id="Point999">
            <gml:pos>999 999 999</gml:pos>
        </gml:Point>
    </position>
    <!-- ================================================ -->
    <positionErrorEstimates>
        <CovarianceMatrix>
            <matrixSize>3</matrixSize>
            <adjustedParameters>X Y Z</adjustedParameters>
            <result>
                <DiagonalMatrix>
                    <valueUnit>m m m</valueUnit>
                    <valuesList>0.999 0.999 0.999</valuesList>
                </DiagonalMatrix>
            </result>
        </CovarianceMatrix>
    </positionErrorEstimates>
    <!-- ================================================ -->
</ObjectPosition>
```


EXAMPLE 3 An example XML fragment for an ObjectPoint that is a control point is:

```
<ObjectPoint gml:id="ObjectPoint999">
   <gml:identifier codeSpace="IGM">ObjectPoint999</gml:identifier>
   <objectPointType
codeSpace="../objectPointTypeValues.xml">control</objectPointType>
   <objectCRS>urn:ogc:def:crs:EPSG:6.8:4979</objectCRS>
   <!-- ============================================== -->
   <objectPosition>
      <ObjectPosition gml:id="ObjectPosition999.999">
         <gml:identifier
codeSpace="IGM">ObjectPosition999.999</gml:identifier>
         <point xlink:href="#ObjectPoint999"/>
         <objectPositionStatus
codeSpace="../objectPositionStatusValues.xml">measured</objectPositionS
tatus>
         <dimension>3</dimension>
         <position>
            <gml:Point gml:id="Point999">
               <gml:pos>999 999 999</gml:pos>
            </gml:Point>
         </position>
         <positionErrorEstimates>
            <CovarianceMatrix>
               <matrixSize>3</matrixSize>
               <adjustedParameters>X Y Z</adjustedParameters>
               <result>
                  <DiagonalMatrix>
                     <valueUnit>m m m</valueUnit>
                     <valuesList>0.999 0.999 0.999</valuesList>
                  </DiagonalMatrix>
               </result>
            </CovarianceMatrix>
         </positionErrorEstimates>
      </ObjectPosition>
   </objectPosition>
   <!-- ============================================== -->
   <identificationImage
xlink:href="identificationImageFile999.999.xml#IdentificationImage999.9
99"/>
</ObjectPoint>
```

NOTE The above example includes inline encoding of the measured ObjectPosition, as expected for control and check points.

12 Covariance matrix

12.1 Introduction

The covariance matrix part of the Image geopositioning metadata GML Application Schema records combined absolute and relative position error estimates. The error statistics are in the form of covariance matrices, also called variance-covariance matrices, used with the (most-likely) values to which the covariance matrices apply. These covariance matrices normally represent correlations between the accuracies of different parameters, coordinates, and positions.

These covariance matrices contain the variance of each adjusted value estimated in the triangulation, and the covariances between these values. These estimated values include various parameters in georeferencing coordinate transformations, including the camera 3D position coordinates and 3D orientation angles. The estimated values also include point position coordinates in object CRSs.

NOTE Such covariance matrices for single point coordinates are described in Table D.33 of [ISO/TS 19138 draft n1934].

12.2 UML model

The UML class diagram for the GCM_CovarianceMatrix package is shown in Figure 10, together with the ISO 19115 classes that are inherited from. The new classes defined in this GCM_CovarianceMatrix package are described in through Table 28 through Table 37.

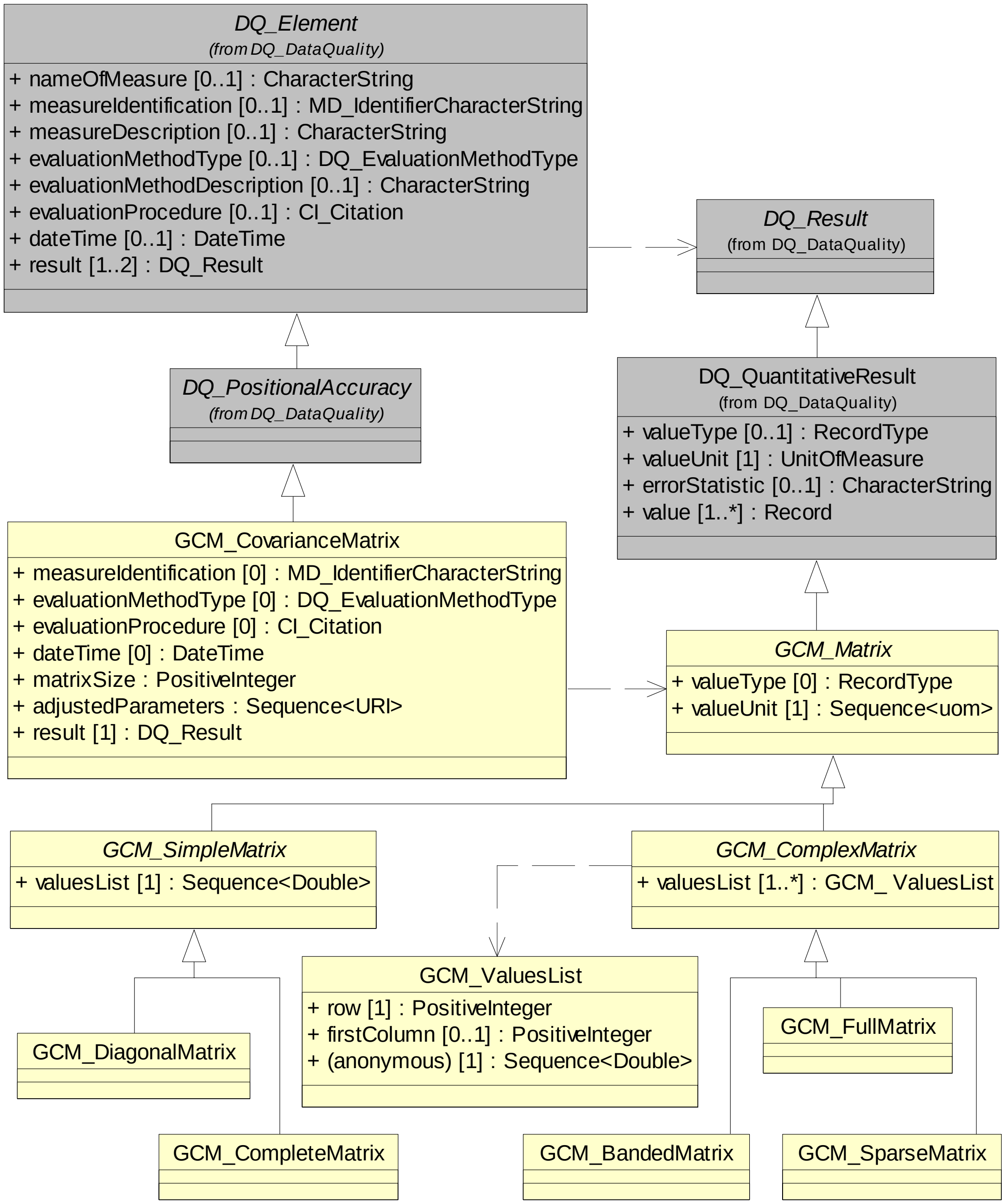

Figure 10 — GCM_CovarianceMatrix package UML class diagram

NOTE This GCM_CovarianceMatrix package class diagram shows all the attributes in the DQ_Element and DQ_QuantitativeResult classes. However, some of the optional attributes in each are not expected to be useful in this application, and are thus prohibited in the GCM_CovarianceMatrix and GCM_Matrix classes.

This UML class diagram shows how multiple types of covariance matrices can be efficiently recorded. The CovarianceMatrix package (for the Image geopositioning metadata GML Application Schema) specializes the DQ_DataQuality package specified in ISO 19115. This is done by defining more-specific subclasses of DQ_PositionalAccuracy and DQ_QuantitativeResult.

The CovarianceMatrix concrete class restricts and extends the DQ_PositionalAccuracy abstract class for combined absolute external and relative internal positional accuracies. This class adds the size of this covariance matrix and an ordered sequence of references to the variable parameters for this covariance matrix.

The Matrix, SimpleMatrix, and ComplexMatrix abstract classes restrict the DQ_QuantitativeResult concrete class for efficiently recording the values of covariance matrix cells. Instead of a general record, the “value” attribute is renamed valuesList, which contains the values of an ordered sequence of covariance matrix cells.

The SimpleMatrix class contains just one valuesList. The Diagonal- and Complete-Matrix classes are specialized subclasses of SimpleMatrix, which specify how the values in that one valuesList correspond to covariance matrix cells.

The ComplexMatrix class contains at least one valuesList for each row in the covariance matrix. The cell position in the covariance matrix of the first value in each of these valuesLists is specified by required “row” and optional firstColumn indices. The Full-, Sparse-, and Banded-Matrix concrete subclasses are specialized subclasses of ComplexMatrix, which specify how the multiple valuesList objects shall be used to encode covariance matrix cells.

NOTE Additional concrete subclasses of SimpleMatrix and ComplexMatrix can be added for more efficient encoding of covariance matrices with other properties.

Table 28 — Defining elements of GCM_CovarianceMatrix class

Description:	Covariance matrix form of error estimates, used for possibly statistically-correlated sets of parameters, containing ordered list of parameter names for matrix rows and columns.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	DQ_PositionalAccuracy
Association roles:	coordinateOperationAccuracy from GAG_AdjustedGroup [1] c (Composition (TBR) association from adjusted group that determined these error estimates) positionErrorEstimates from GOP_ImagePosition [1] (Association from image position for these absolute and relative position error estimates) positionErrorEstimates from GOP_ObjectPosition [1] (Association from absolute and relative position error estimates for this object position)
Public attributes:	(The first eight optional attributes listed below are inherited from DQ_Element through DQ_PositionalAccuracy, with four of these attributes being prohibited here.)

Attribute name	UML identifier	Data type	Obligation	Maximum occurrence	Attribute description
Name Of Measure	nameOfMeasure	CharacterString	O	N [b]	Name of the test applied to the data.
Measure Identification	measureIdentification	MD_Identifier	O	0 [a]	Code identifying a registered standard procedure.
Measure Description	measureDescription	CharacterString	O	1 [b]	Description of the measure.
Evaluation Method Type	evaluationMethodType	DQ_EvaluationMethodType	O	0 [a]	Type of method used to evaluate quality of the dataset.
Evaluation Method Description	evaluationMethodDescription	CharacterString	O	1	Description of the evaluation method. [b]
Evaluation Procedure	evaluationProcedure	CI_Citation	O	0 [a]	Reference to the procedure information.
Date and Time	dateTime	DateTime	O	0 [a]	Date or range of dates on which a data quality measure was applied.
Result	result	DQ_Result [c]	M	1	Value (or set of values) obtained from applying a data quality measure.
Matrix Size	matrixSize	PositiveInteger	M	1	Number or rows and columns in this covariance matrix.
Adjusted Parameter	adjustedParameter	URI	M	N	Reference to adjusted parameter whose estimated error is represented by one row and one column in covariance matrix.

a Inclusion of this attribute is prohibited in this GCM_CovarianceMatrix class subclass of the DQ_PositionalAccuracy subclass of the DQ_Element class.

b In this GCM_CovarianceMatrix class, this attribute is allowed but is not expected to be included.

c Limited to a concrete subclass of the abstract GCM_Matrix class, which specializes the DQ_QuantitativeResult subclass of the DQ_Result class.

Table 29 — Defining elements of GCM_Matrix class

Description:	Covariance matrix cell values.
Stereotype:	Type
Class attribute:	Abstract
Inheritance from:	DQ_QuantitativeResult
Association roles:	(none)
Public attributes:	(All three listed attributes are inherited from DQ_QuantitativeResult, with two attributes being restricted here.)

Attribute name	UML identifier	Data type	Obligation	Maximum occurrence	Attribute description
Value Type	valueType	RecordType	0	0 [a]	Value type for reporting a data quality result.
Value Unit	valueUnit	Sequence<uom> [b]	M	1	Value unit for reporting a data quality result. [b]
Error Statistic	errorStatistic	CharacterString	0	1 [c]	Statistical method used to determine the value.

a Inclusion of the valueType attribute is prohibited in this GCM_ Matrix subclass of DQ_QuantitativeResult.

b In this GCM_Matrix class, the valueUnit attribute shall contain an ordered list of references to the units for each covariance matrix row, with the same units for each column, and with the unit for each matrix cell being the product of the corresponding row and column units.

c In this GCM_ Matrix class, this attribute is allowed but is not expected to be included.

Table 30 — Defining elements of GCM_SimpleMatrix class

Description:	Covariance matrix encoded using a one list of cell values.
Stereotype:	Type
Class attribute:	Abstract
Inheritance from:	GCM_Matrix
Association roles:	(none)
Public attributes:	Two attributes inherited from GCM_Matrix, plus one attribute inherited and restricted from DQ_QuantitativeResuilt as follows:

Attribute name	UML identifier	Data type	Obligation	Maximum occurrence	Attribute description
Values List [a]	valuesList	Sequence<Double> [a]	M	1 [b]	Ordered sequence of values for covariance matrix cells.

a The “value” attribute is renamed “valueList”, and the Record data type is here specialized to be an ordered sequence of Double typed values.

b Restricted to one occurrence here.

Table 31 — Defining elements of GCM_DiagonalMatrix class

Description:	Covariance matrix containing values for only the matrix diagonal cells. (Such a matrix is for a statistically-uncorrelated set of parameters). For this Diagonal Matrix, the "valuesList" shall contain an ordered list of the values of only the cells on the matrix diagonal.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GCM_SimpleMatrix
Association roles:	(none)
Public attributes:	Three attributes inherited from GCM_SimpleMatrix

Table 32 — Defining elements of GCM_CompleteMatrix class

Description:	Covariance matrix containing values for diagonal and upper-right cells of symmetrical covariance matrix. (Such a matrix is for a statistically-correlated set of parameters). For this CompleteMatrix, the "valuesList" element shall contain an ordered list of the values in each row starting with the diagonal cell in each row and continuing to the right. May be used for a 2D covariance matrix, and thus contain three values for cells (1 1), (1 2), and (2 2). May be used for a 3D covariance matrix, and thus contain six values for cells (1 1), (1 2), (1 3), (2 2), (2 3), and (3 3).
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GCM_SimpleMatrix
Association roles:	(none)
Public attributes:	Three attributes inherited from GCM_SimpleMatrix

Table 33 — Defining elements of GCM_ComplexMatrix class

Description:	Generic covariance matrix error estimates, for statistically-correlated sets of parameters.				
Stereotype:	Type				
Class attribute:	Abstract				
Inheritance from:	GCM_Matrix				
Association roles:	(none)				
Public attributes:	Two attributes inherited from GCM_Matrix, plus one attribute inherited and restricted from DQ_QuantitativeResuilt as follows:				
Attribute name	**UML identifier**	**Data type**	**Oblig ation**	**Maximum occurrence**	**Attribute description**
Values List [a]	valuesList	GCM_ValueList [a]	M	N	Ordered sequence of values for adjacent covariance matrix cells, with position of first cell.
a The "value" attribute is renamed "valueList", and the Record data type is here specialized to be an ordered sequence of GCM_ValueList typed values.					

Table 34 — Defining elements of GCM_ValueList class

Description:	Ordered sequence of one or more covariance matrix cell values, beginning at (row firstColumn) and continuing to the right.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	(none)
Association roles:	(none)
Public attributes:	

Attribute name	UML identifier	Data type	Obligation	Maximum occurrence	Attribute description
Row	row	Positive Integer	M	1	Index of row in covariance matrix of first cell value included.
First Column	firstColumn	Positive Integer	C [a]	1	Index of column in covariance matrix of first cell value included.
(anonymous)	(anonymous)	Sequence<Double>	M	1	Ordered sequence of values for adjacent covariance matrix cells.

a Default value is the value of associated "row". Attribute should be omitted when value would be same as associated "row". Value shall not be less than the value of associated "row".

Table 35 — Defining elements of GCM_FullMatrix class

Description:	Covariance matrix containing values for all matrix cells in a symmetric matrix. For this Full Matrix, one "valuesList" element (containing row and firstColumn attributes) shall be included for each matrix row. Each valuesList element shall contain the value of one diagonal matrix cell, followed by the values of all other cells to the right in that matrix row (and down in that matrix column). Each sequence shall end when the right edge of this matrix is reached.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GCM_ComplexMatrix
Association roles:	(none)
Public attributes:	Three attributes inherited from GCM_ComplexMatrix

Table 36 — Defining elements of GCM_SparseMatrix class

Description:	Covariance matrix containing only or mostly values of non-zero matrix cells, for a symmetric matrix. For this Sparse Matrix, at least one "valuesList" element shall be included for each matrix row. Each valuesList sequence shall end when several sequential zero cell values are reached or the right edge of this matrix is reached. Each valuesList element shall contain a list of the values of this sequence of (mostly non-zero) matrix cells.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GCM_ComplexMatrix
Association roles:	(none)
Public attributes:	Three attributes inherited from GCM_ComplexMatrix

Table 37 — Defining elements of GCM_BandedMatrix class

Description:	Covariance matrix containing values of all non-zero matrix cells, for a symmetric matrix with only zero values in the top-right and bottom-left corners. For this Banded Matrix, one "valuesList" element (containing row and firstColumn attributes) shall be included for each matrix row. Each valuesList element shall contain the value of one diagonal matrix cell, followed by the values of other cells to the right in that matrix row (and down in that matrix column). Each sequence shall end when there are no more non-zero cell values in that row or the right edge of this matrix is reached.
Stereotype:	Type
Class attribute:	Concrete
Inheritance from:	GCM_ComplexMatrix
Association roles:	(none)
Public attributes:	Three attributes inherited from GCM_ComplexMatrix

12.3 XML encoding

The GCM_CovarianceMatrix UML package is encoded in the igmCovarianceMatrix.xsd XML Schema Document, which imports the gmd/dataQuality.xsd and gco/gco.xsd XML Schema Documents from ISO 19139. The ISO 19115 classes used in the UML model are encoded by the ISO 19139 XML elements as listed in Table 3. The contents of the igmCovarianceMatrix.xsd XML Schema Document shall be as specified in the attached file.

NOTE The following examples use “999” as a place holder for specific numerical values.

EXAMPLE 1 An example XML fragment for a CovarianceMatrix using a 2 by 2 DiagonalMatrix, suitable for an ImagePosition, is:

```
<CovarianceMatrix id="CovarianceMatrix999>
    <matrixSize>2</matrixSize>
    <adjustedParameters>row column</adjustedParameters>
    <result>
        <DiagonalMatrix>
            <valueUnit>rowSpacing columnSpacing</valueUnit>
            <valuesList>0.999 0.999</valuesList>
        </DiagonalMatrix>
    </result>
</CovarianceMatrix>
```

EXAMPLE 2 An example XML fragment for a CovarianceMatrix using a 3 by 3 CompleteMatrix, suitable for an ObjectPosition, is:

```
<CovarianceMatrix id="CovarianceMatrix999>
    <matrixSize>3</matrixSize>
    <adjustedParameters>X Y Z</adjustedParameters>
    <result>
        <CompleteMatrix>
            <valueUnit >m m m</valueUnit>
            <valuesList>0.999 0.999 0.999 0.999 0.999 0.999</valuesList>
        </CompleteMatrix>
    </result>
</CovarianceMatrix>
```


EXAMPLE 3 An example XML fragment for a CovarianceMatrix using a 4 by 4 BandedMatrix is:

```
<CovarianceMatrix id="CovarianceMatrix999">
   <matrixSize>4</matrixSize>
   <adjustedParameters>X Y Z T</adjustedParameters>
   <result>
      <BandedMatrix>
         <valueUnit>m m m</valueUnit>
         <valuesList row="1">0.999 0.999 </valuesList>
         <valuesList row="2">0.999 0.999 </valuesList>
         <valuesList row="3">0.999 0.999 </valuesList>
         <valuesList row="4">0.999</valuesList>
   </BandedMatrix>
   </result>
</CovarianceMatrix>
```

13 IGM data packages

This clause specifies the types of data packages (or files) in which image geopositioning metadata (IGM) encoded using GML 3.2 should be packaged for effective data transfer and retrieval. As an XML document, each IGM file or data package shall contain one root XML element. This root is always a GML Object XML element, usually a GML Object specific to IGM.

NOTE This root XML element could be a gml:Dictionary which contains multiple other GML Objects.

The specified UML model and GML encoding of IGM are largely abstract, for flexibly encoding this metadata as GML objects and encoded associations between related objects. Over 20 new GML Objects are specified. However, multiple associated object instances can be encoded inline within one GML Object instance. Furthermore, an abstract GML object can be the basis for one data package type that can contain any of the concrete objects in that XML Schema substitution group.

The IGM data package types defined for transferring IGM Objects should include the package types specified in Table 38 through Table 47. Each of those tables lists the:

a) Root element, for an XML document which is a GML Object instance

b) Inline encode, listing associated GML Objects encoded within the root element

c) Number, of this data package type in a data base of IGM packages

d) Search by, values of listed XML elements to find desired data package in data base

e) Profile, of this application schema that can be used to encode this IGM data package type

f) Template, for an XML document of this data package type

g) File name, recommended for a file with this data package type (adapted from Annex C)

These IGM data package types are defined with the objective of minimizing normal client needs to assemble and disassemble data packages. The elements encoded inline are expected to be usually used and/or updated at the same time. Also, the elements encoded inline have same multiplicity, except for multiple points. There are now 10 different types of data packages.

NOTE All IGM elements might be transferred in one large data package, requiring clients to assemble and disassemble packages for each use. This would require using separate types of data packages for each use, or using one data package type with extremely general contents.

Each of these IGM data package types can be encoded using a (subset) profile of this Image geopositioning metadata GML 3.2 application schema. Most of these profiles are defined by a XML Schema Document that is referenced by table row e) and is attached to this specification. Also attached are template (or example) XML documents for most of these IGM data package types, referenced in row f) of these tables.

Table 38 — ObjectImageTransformation data package type

Root element	igm:ObjectImageTransformation
Inline encode	One non-abstract GML Object in igm:ImageParameterValues substitution group (There is one such non-abstract GML Object element for each image sensor type.) This object contains: One igm:ImageAccuracySummary Zero or one gml:Polygon Object, computed using this group of image parameter values, included when image footprint has been computed Zero or more igm:ImagePosition Objects, computed using this group of image parameter values, with each object containing: One gml:Point object Zero or one igm:CovariancceMatrix object for position error estimates, included unless this point is covered by covariance matrix with other point(s), containing: One igm:Matrix object concrete subtype All GML Objects used for image parameters of specific sensor type (For some of these image parameters, a file shall be attached providing parameter values that are not encoded in XML.)
Number	One for each adjustment of each image collected from different position or attitude
Search by	gml:identifier, gml:description for root element or igm:ImageParameterValues
Profile	(Depends on each specific supported image geometry model), which uses igmImagePositionPackagePart.xsd
Template	(Depends on each specific supported image geometry model)
File name	transformationFile999.999.xml

Table 39 — SensorParameterValues data package type

Root element	One non-abstract GML Object in igm:SensorParameterValues substitution group (There is one such non-abstract GML Object element for each image sensor type.)
Inline encode	All GML Objects used for sensor parameters of specific sensor type (For some of these sensor parameters, a file shall be attached providing parameter values that are not encoded in XML.)
Number	One for each adjustment of each image sensor (or sensor configuration for image sensors having multiple configurations)
Search by	gml:identifier, gml:description for root element
Profile	(Depends on each specific supported image geometry model)
Template	(Depends on each specific supported image geometry model)
File name	sensorParametersFile999.999.xml

Table 40 — AdjustedGroup data package type

Root element	igm:AdjustedGroup
Inline encode	One igm:CovarianceMatrix, for coordinateOperationAccuracy of adjusted group, containing: One non-abstract GML Object in igm:Matrix substitution group Zero or more igm:ObjectPosition, one for each object point whose position was computed with this adjusted group, each containing: One gml:Point giving the object position Zero or one igm:CovarianceMatrix for position error estimates, containing: One non-abstract Object in igm:Matrix substitution group One AdjustmentSummary Zero or one gml:Polygon, included when adjusted group footprint has been computed
Number	One for each adjustment of each adjusted group
Search by	gml:Identifier, timeStamp, gml:description of root element, gml:Polygon for footprint
Profile	igmAdjustedGroupPackage.xsd, which uses igmObjectPositionFilePackage.xsd
Template	templateAdjustedGroupPackage.xml
File name	adjustedGroupFile999.xsd

Table 41 — ImageInfo data package type

Root element	igm:ImageInfo
Inline encode	Zero or more igm:ImagePosition, for each object point whose position was measured in this image, each containing: One gml:Point giving image position, in each igm:ImagePosition One igm:CovariuanceMatrix, giving position error estimates, containing: One non-abstract GML Object in igm:Matrix substitution group
Number	One for each image collected from different position or attitude for each image sensor
Search by	gml:identifier, gml:description of root element
Profile	igmImageInfoPackage.xsd, which uses igmImagePositionFilePackage.xsd
Template	templateImageInfoPackage.xml
File name	imageInfoFile999.xsd

Table 42 — ImageSensor data package type

Root element	igm:ImageSensor
Inline encode	(none)
Number	One for each image sensor (or sensor configuration for image sensors having multiple configurations)
Search by	gml:identifier, gml:description
Profile	igmImageSensorPackage.xsd
Template	templateImageSensorPackage.xml
File name	imageSensorFile999.xsd

Table 43 — AdjustableParameters data package type

Root element	igm:AdjustableParameters
Inline encode	(none)
Number	One for each different set of adjusted parameters
Search by	gml:identifier, gml:description
Profile	igmAdjustableParametersPackage.xsd
Template	templateAdjustableParametersPackage.xml
File name	adjustableParametersFile999.xsd

Table 44 — EqualParameters data package type

Root element	igm:EqualParameters
Inline encode	(none)
Number	One for each different set of equal parameters
Search by	gml:identifier, gml:description
Profile	igmEqualParametersPackage.xsd
Template	templateEqualParametersPackage.xml
File name	equalParametersFile999.xsd

Table 45 — ObjectPoints data package type

Root element	igm:ObjectPoints
Inline encode	One or more igm:ObjectPoint, containing: Zero or more igm:ObjectPosition, included for a control or check point (with measured object position), containing: One gml:Point giving object position, in igm:ObjectPosition Zero or one igm:CovariuanceMatrix, giving position error estimates, containing: One non-abstract Object in igm:Matrix substitution group
Number	One for each selected group of tie, control, and check points
Search by	gml:identifier of igm:ObjectPoint, gml:pos with srsName in gml:Point of igm:ObjectPosition, gml:description of igm:ObjectPoint
Profile	igmObjectPointsPackage.xsd, which uses igmObjectPositionPackagePart.xsd
Template	templateObjectPointsPackage.xml
File name	objectPointsFile999.xsd

Table 46 — OperationMethod data package type

Root element	gml:OperationMethod
Inline encode	Two or more OperationParameterGroup, for each parameter group used by specific image sensor, each containing" Two or more OperationParameter, for each parameter used by parameter groups for specific image sensor
Number	One for each image sensor (or sensor configuration for image sensors having multiple configurations)
Search by	gml:identifier, gml:description of root element
Profile	TBD
Template	(Depends on each specific supported image geometry model)
File name	TBD

Table 47 — IdentificationImage data package type

Root element	igm:IdentificationImage
Inline encode	TBD
Number	One for each identification image of each control and check point
Search by	gml:identifier, gml:description of root element
Profile	igmIdentificationImagePackage.xsd
Template	TBD
File name	identificationImageFile999.999.xsd

Annex A
(normative)

Abstract test suite

An abstract test suite is not provided in this version of this Implementation Specification, but should be included in version 1.0.0.

Annex B
(normative)

XML Schema Documents

In addition to this document, this specification includes some normative XML Schema Documents. These XML Schema Documents are bundled in a zip file with the present document. After OGC acceptance of a Version 1.0.0 of this specification, these XML Schema Documents will also be posted online at the URL http://schemas.opengeospatial.net/igm/1.0.0. In the event of a discrepancy between the bundled and online versions of the XML Schema Documents, the online files shall be considered authoritative.

The Image geopositioning metadata GML Application Schema specified in this document use six specified XML Schema Documents included in the zip file with this document. These XML Schema Documents match the six UML packages described in Clauses 7 through 11, and are named:

igmSensorModel.xsd

igmImageGeometry.xsd

igmSensorCalibration.xsd

igmAdjustedGroup.xsd

igmObjectPoint.xsd

igmCovarianceMatrix.xsd

These XML Schema Documents import and build upon parts of many of the GML 3.2.0 XML Schema Documents specified in [OGC 05-105].

NOTE 1 The informative attachments to this document include the (subset) profiles XML Schema Documents referenced in Clause 13, namely:

igmAdjustableParametersPackage.xsd

igmAdjustedGroupPackage.xsd

igmEqualParametersPackage.xsd

igmIdentificationImagePackage.xsd

igmImageInfoPackage.xsd

igmImagePositionPackagePart.xsd

igmImageSensorPackage.xsd

igmObjectPointsPackage.xsd

igmObjectPositionPackagePart.xsd

All these XML Schema Documents contain documentation of the meaning of each element and attribute, and this documentation shall be considered normative as specified in Subclause 11.6.3 of [OGC 06-121r3].

NOTE 2 The informative attachments to this document include the template XML documents referenced in Clause 13, namely:

templateAdjustableParametersPackage.xml

templateAdjustedGroupPackage.xml

templateEqualParametersPackage.xml

templateImageInfoPackage.xml

templateImageSensorPackage.xml

templateObjectPointsPackage.xml

Annex C
(informative)

Identifiers and names

C.1 Introduction

This annex describes and discusses the identifier and name parameters used in GML and in this document, including their uses and values. Many of the GML Objects defined and used by this Image Geopositioning Metadata have three different parameters which can be used to identify or name that object:

a) The gml:id XML attribute with type xsd:ID, required in most objects

b) The gml:identifier XML element with type gml:CodeWithAuthorityType, required in many objects

c) The gml:name XML element with type gml:CodeType, optional and repeatable in many objects

In addition to these parameters, many objects also include parameters that can be used to describe that object and its uses, including:

d) The gml:description XML element with type xsd:string, optional in many objects

e) The gml:remarks XML element with type xsd:string, optional in many objects

NOTE The above list does not include the gml:metaDataProperty and gml:descriptionReference elements that can be included in many objects, usually referencing remote information. The identifiers and names uses and values discussed here do not reflect OGC best practices, since no OGC best practices currently exist. Furthermore, many of the possible URNs in the “ogc” URN namespace that might be used have not yet been specified.

C.2 GML intended uses

The uses of these parameters intended in GML 3.2 are summarized by these quotes:

a) gml:id: “The attribute gml:id supports provision of a handle for the XML element representing a GML Object. Its use is mandatory for all GML objects. It is of XML type ID, so is constrained to be unique in the XML document within which it occurs.”

b) gml:identifier: “Often, a special identifier is assigned to an object by the maintaining authority with the intention that it is used in references to the object. For such cases, the codeSpace shall be provided. That identifier is usually unique either globally or within an application domain. gml:identifier is a pre-defined property for such identifiers.”

c) gml:name: “The gml:name property provides a label or identifier for the object, commonly a descriptive name. An object may have several names, typically assigned by different authorities. gml:name uses the gml:CodeType content model. The authority for a name is indicated by the value of its (optional) codeSpace attribute. The name may or may not be unique, as determined by the rules of the organization responsible for the codeSpace. In common usage there will be one name per authority, so a processing application may select the name from its preferred codeSpace."

d) gml:description: "The value of this property is a text description of the object."

e) gml:remarks: "The gml:remarks element shall be used to hold additional textual information that is not conceptually part of the definition but is useful in understanding the definition."

Simon Cox has posted a web page discussing this subject, publicly available at https://www.seegrid.csiro.au/twiki/bin/view/AppSchemas/LabelsAndHandles. Three excerpts from that page are quoted below:

"Names and identifiers is a sometimes confusing topic. In some contexts there is an apparently clear distinction, with names being "words" and identifiers being numbers or codes. However, in practice a "name" is just a memorable identifier, and these concepts slide into each other. Thus it is appropriate to consider them together and use a unified approach."

"ID value - system assigned handle for XML element

The value of gml:id will normally be assigned according to rules determined by the service provider responsible for generating the XML representation of the data. The rules for assigning document identifiers are, in general, outside the scope of an *encoding* standard.

The value may be opaque, and should not be assumed to have any significance outside the context of the document or system of documents, so caution should be applied in using it as a persistent identifier. Note also that strictly speaking the id applies to the XML element in its document context, rather than the real-world object being described. It has similar significance to a "Primary Key" in a RDBMS system.

It may be *convenient* for the value to have some mnemonic significance, but this is *not required*. It is merely necessary that the value of the gml:id attribute is unique within the current context (i.e. the XML document). A particular service may use locally convenient rules to generate unique id's: these may use timestamps, sequence numbers, even random numbers as inputs."

"Name - externally assigned designator for the real world object

The label or name is usually assigned by an external agency to the real-world object under consideration, according to the relevant business rules of that organisation. In many cases the rules will specify that the name is unique and has continuing significance in the external context, so *if present* it should be treated as *persistent*. Instead of "organisation", substitute "family" or "registry of births, deaths and marriages" and the rule still holds up.

Names may be any combination of characters, including whitespace. The value will often be "human readable", possibly with semantic significance. However, gml:name elements should (be) used to carry *all* externally-significant designators (names, labels, identifiers).

Some objects may have multiple names, which may be assigned by different authorities. In GML (and XMML) these are disambiguated by using the **codeSpace** attribute on the name element. The convention is that the value of a name element with no codeSpace is the canonical name in the context of the service providing the document. Aliases from other authorities are given in additional name elements which do carry a codeSpace."

The above quote about gml:id does not make clear that it will be used to find an element within an XML document, when an element is referenced using a URI in the same or different XML document. For this use, a gml:id cannot contain any spaces, or start with any digit.

Simon does not similarly describe the gml:identifier. As I understand his complete discussion, the gml:identifier should be used for the primary label that would otherwise be encoded as a gml:name.

C.3 IGM recommended uses

The currently recommended uses of these identification parameters in IGM are:

a) gml:id: The required gml:id attribute supports referencing that object within an XML document, when an element is referenced using a URI in the same or different XML document. For this use, a gml:id cannot contain any spaces, or start with any digit. In IGM, a gml:id should be human readable, and should name the type of IGM object. In addition, a gml:id should be searchable in an IGM database, and thus unique in that IGM database (not just unique in one XML document).

b) gml:identifier: The required gml:identifier element should be a human readable descriptive name, and should name the type of IGM object provided. In addition, a gml:identifier should be globally unique across all GML applications, including multiple IGM databases. For this use, the codeSpace should identify a specific IGM application, such as a specific IGM database.

c) gml:name: The optional and repeatable gml:name elements should also be human readable descriptive names, possible using different languages or different terms. An object may have several names, sometimes assigned by different authorities. The authority for a name is indicated by the value of its (optional) codeSpace attribute. An object name should be unique in a codeSpace.

d) gml:description: The optional gml:description element should be included when a (human readable) text description of the object would be useful. For example, the gml:description may be used to describe the appearance of a control or check point.

e) gml:remarks: The optional gml:remarks element should be included if and when more (human readable) text about an object would be useful, which is not conceptually part of the description.

The currently specified IGM uses two types of referencing of other objects and parts of objects:

a) Object associations encoded as references (not encoded inline), using the xlink:href XML attribute with type xsd:anyURI. There are numerous association references to objects allowed and required in IGM. A gml:id parameter must be referenced in each xlink:href, as a fragment identifier following a number sign ("#") character. To reference an object in the same XML document, the xlink:href value needs to contain only that fragment identifier. To reference an object in the different XML document, the xlink:href value shall reference that file preceding the fragment identifier.

b) References to adjustable parameters encoded within concrete elements in the substitution groups of the igm:ImageParameterValue and igm:SensorParameterValue abstract XML elements. A list of such references is contained in the “parameters” element within each igm:AdjustableParameters and igm:EqualParameters element, and in the adjustedParameters element within each igm:CovarianceMatrix element. Sometimes the referenced (adjustable) parameter is a GML object that contains only one parameter value. In these cases, the gml:id of the referenced object should be used. However, sometimes the referenced parameter is one of several parameter values in one GML object, where the multiple values are encoded using the gml:doubleList data type. In these cases, the gml:id of the referenced object should be used followed by the integer index of the desired value in the gml:doubleList. To reference any gml:id with the same object type prefix, that prefix may be followed by “Any”.

EXAMPLE To reference the first item in a data structure with any gml:id of the form “SensorPosition999.999”, the gml:id may be “SensorPositionAny1”.

C.4 Recommended values

The currently recommended pattern for gml:id and gml:identifier values is to use a text name for the type of object, which is the same as or part of the name of that XML element (without the “igm:” prefix). This name is immediately followed (without a space or punctuation mark) by an alphanumeric item identifier, which is often an integer number. That identifier may be followed by a period “.” and a second identifier. Together, these two identifiers identify each instance of that object type. These recommended gml:id (or gco:id) values are thus as listed in Table C.1.

These gml:id values are also used as the gml:identifier string values, preceded by the fragment identifier symbol “#” followed (without a space).

Table C.1 — Recommended gml:id values

Object XML element name	gml:id value [a]	Use for 999 or 999.999 value
igm:ObjectImageTransformation	Transformation999.999	Image identifier and adjustment identifier
igm:ImageInfo	ImageInfo999	Image identifier
igm:AbstractImageParameterValues	(none)	(none)
igm:ImageOrientation	(none)	(none)
igm:SensorPosition	SensorPosition999.999 [c]	Image identifier and adjustment identifier
igm:SensorAttitude	SensorAttitude999.999 [c]	Image identifier and adjustment identifier
igm:AdjustableParameters	AdjustableParameters999	AdjustableParameters identifier
igm:ImageAccuracySummary	(none)	(none)
igm:ImageSensor	ImageSensor999	Sensor identifier
igm:AbstractSensorParameterValues [b]	SensorParameters999.999	Sensor identifier and adjustment identifier
igm:AdjustedGroup	AdjustedGroup999	Adjustment identifier
igm:EqualParameters	EqualParameters999	EqualParameters identifier
igm:AdjustmentSummary	(none)	(none)
igm:ObjectPointsPackage	ObjectPointsPackage999	ObjectPoints package identifier
igm:ObjectPoint	ObjectPoint999	ObjectPoint identifier
igm:ImagePosition	ImagePosition999.999 [c]	Image identifier and point identifier
igm:ObjectPosition	ObjectPosition999.999 [c]	Point identifier and adjustment identifier
igm:IdentificationImage	IdentificationImage999.999	Point identifier and image identifier
igm:CovarianceMatrix	CovarianceMatrix999	CovarianceMatrix identifier
igm:DiagonalMatrix	DiagonalMatrix999 [c, d]	Matrix identifier
igm:CompleteMatrix	CompleteMatrix999 [c, d]	Matrix identifier
igm:FullMatrix	FullMatrix999 [c, d]	Matrix identifier
igm:BandedMatrix	BandedMatrix999 [c, d]	Matrix identifier
igm:SparseMatrix	SparseMatrix999 [c, d]	Matrix identifier
gml:Point	Point999	Point identifier
gml:Vector	Vector999	Vector identifier
gml:Polygon	Polygon999	Polygon identifier

a The “999”or “999.999” in each value is replaced by an item identifier, often integer number(s).

b The same gml:id values can be used for all concrete elements in the AbstractSensorParameterValues substitution group.

c This object is not expected to be referenced remotely.

d This is a concrete element in the igm:Matrix substitution group.

Bibliography

[1] ASPRS 2004, Manual of Photogrammetry, Fifth Edition, American Society for Photogrammetry and Remote Sensing, 2004

[2] ISO 19105:2000, Geographic information — Conformance and Testing

[3] ISO CD 19130-2 (n1772), Geographic information - Sensor data model for imagery and gridded data

[4] ISO/TS 19138 draft n2027, Geographic information — Data quality measures

[5] LabelsAndHandles, Names, Identifiers, Labels and Handles, by Simon Cox, available at https://www.seegrid.csiro.au/twiki/bin/view/AppSchemas/LabelsAndHandles

[6] OGC 00-115, OGC Abstract Specification Topic 15: Image Exploitation Services

[7] OGC 00-116, OGC Abstract Specification Topic 16: Image Coordinate Transformation Services

[8] OGC 05-094r1, GML 3.1.1 CRS support profile

[9] OGC 07-030, OpenGIS® Image Geopositioning Service Implementation Specification

[10] OGC 05-086r2, Sensor Markup Language (SensorML)

[11] OGC 06-010r4, Transducer Markup Language (TML)

[12] USIGS Glossary

