Open Geospatial Consortium

35 Main Street, Suite 5
Wayland, MA 01778
Telephone: +1-508-655-5858
Facsimile: +1-508-655-2237

Editor:
Telephone: +1-703-830-6516
Facsimile: +1-703-830-7096
editor@opengeospatial.org

The OpenGIS® Abstract Specification

Topic 16: Image Coordinate Transformation Services

Version 6

OpenGIS® Project Document Number 00-116.doc

Revision History

Date	Description
24 February 1999	New topic volume from project document 99-002r2.
30 March 1999	Update to use new document template following guidance of change proposal 99-010 w/ friendly amendments to remove Section 1 boilerplate from individual topic volumes, approved 9 February 1999.
22 June 1999	Update following sections with changes from change proposal 99-030r2 (w/ friendly changes), approved at April 1999 TC meeting: Sections 4.1.6 through 4.1.27, 4.3.5, 4.3.6, 4.3.7, 4.4.5, 4.4.6, 4.4.7, 4.5, 5 and 6.
14 December, 1999	Update with changes from change proposal 99-058r3 (with friendly changes), approved at the December 1999 TC meeting. Section 3.2 was added, and sections 2.5, 2.10, 5, and 6 were edited.

This page is intentionally left blank.

Table of Contents

1. Introduction

1.1. The Abstract Specification

The purpose of the Abstract Specification is to create and document a conceptual model sufficient enough to allow for the creation of Implementation Specifications. The Abstract Specification consists of two models derived from the Syntropy object analysis and design methodology [1].

The first and simpler model is called the Essential Model and its purpose is to establish the conceptual linkage of the software or system design to the real world. The Essential Model is a description of how the world works (or should work).

The second model, the meat of the Abstract Specification, is the Abstract Model that defines the eventual software system in an implementation neutral manner. The Abstract Model is a description of how software should work. The Abstract Model represents a compromise between the paradigms of the intended target implementation environments.

The Abstract Specification is organized into separate topic volumes in order to manage the complexity of the subject matter and to assist parallel development of work items by different Working Groups of the OGC Technical Committee. The topics are, in reality, dependent upon one another— each one begging to be written first. *Each topic must be read in the context of the entire Abstract Specification*.

The topic volumes are not all written at the same level of detail. Some are mature, and are the basis for Requests For Proposal (RFP). Others are immature, and require additional specification before RFPs can be issued. The level of maturity of a topic reflects the level of understanding and discussion occurring within the Technical Committee. Refer to the OGC Technical Committee Policies and Procedures [2] and Technology Development Process [3] documents for more information on the OGC OpenGIS™ standards development process.

Refer to Topic Volume 0: Abstract Specification Overview [4] for an introduction to all of the topic volumes comprising the Abstract Specification and for editorial guidance, rules and etiquette for authors (and readers) of OGC specifications.

1.2. Introduction to Image Coordinate Transformation Services

This topic volume is the portion of the OpenGIS™ Abstract Specification that covers image coordinate conversion services. That is, this part of the abstract specification describes services for transforming image position coordinates, to and from ground position coordinates. These services might alternately be called "Image Geometry Model Services."

1.3. References for Section 1

[1] Cook, Steve, and John Daniels, Designing Objects Systems: Object-Oriented Modeling with Syntropy, Prentice Hall, New York, 1994, xx + 389 pp.

[2] Open GIS Consortium, 1997. OGC Technical Committee Policies and Procedures, Wayland, Massachusetts. Available via the WWW as <http://www.opengis.org/techno/development.htm>.

[3] Open GIS Consortium, 1997. The OGC Technical Committee Technology Development Process, Wayland, Massachusetts. Available via the WWW as <http://www.opengis.org/techno/development.htm>.

[4] Open GIS Consortium, 1999. Topic 0, Abstract Specification Overview, Wayland, Massachusetts. Available via the WWW as <http://www.opengis.org/techno/specs.htm>.

2. Background for Image Coordinate Transformation Services

This topic volume is the portion of the OpenGIS™ Abstract Specification that covers image coordinate conversion services. That is, this part of the abstract specification describes services for transforming image position coordinates to and from ground position coordinates. These services might alternately be called “Image Geometry Model Services.”

This section provides background information useful in understanding the Image Coordinate Transformation Services discussed in later sections.

2.1. Image Coordinates

The position of a point in an image is specified in two-dimensional (2-D) image coordinates. A point of interest may be either a real or virtual point. The material in this topic volume is currently limited to 2-D images; it could be expanded in the future to include 3-D images.

For a digital image, the 2-D image coordinates are usually specified in the row and column directions of image pixels. The reference point for these image coordinates is usually the corner pixel of the image where the pixel row and column indices are either (1, 1) or (0, 0). Alternately, the reference point for image coordinates could be the center pixel, when the number of pixels is an odd number in each axis. When the number of pixels is even in each axis, the reference point for image coordinates could be the middle of the four most-center pixels.

Image coordinates could be specified as integers, referring to pixel indices. However, image coordinates are usually specified as floating point numbers, to allow coordinates to represent positions with fractional pixel spacings. The center of the corner pixel is often considered to have coordinates (1, 1). Alternately, the outside corner of the corner pixel could be considered to have coordinates (0, 0).

For a film or hardcopy image, the 2-D image coordinates are usually specified in two orthogonal axes, which could be called x and y or u and v. Various reference points can be used, such as the “center” of the image. These image coordinates are often specified as floating point numbers, in units of millimeters.

2.2. Ground Coordinates

The position of a point in ground space is specified in three-dimensional (3-D) coordinates. This document uses the term ground coordinates; photogrammetry often uses the alternate term “object coordinates.” Again, a point of interest may be a real or virtual point.

For Image Coordinate Transformation Services, ground coordinates are almost always 3-D, since the position of a point in all three dimensions affects the image position corresponding to that point. These ground coordinates are specified in some Spatial Reference System (SRS), such as any SRS defined in Abstract Specification Topic 2.

2.3. Position Accuracy

These Image Coordinate Transformation Services assume that most image and ground coordinates have a relatively high precision or accuracy. Image coordinates might be known or desired with an accuracy between a few tens of pixel spacings and a few tenths of one pixel spacing. Ground coordinates might be known or desired with an accuracy between a few tens of meters and a few millimeters.

The numerical values of image and ground point coordinates always have limited accuracies. If the accuracy were truly unknown, the numerical value would have little practical value. If the accuracy is known, that accuracy should be communicated from the data producer to the data user, so that the data user can properly interpret and use the numerical value. Recording of position accuracy information is therefore required by the OpenGIS Abstract Specification, Topic 1: Feature Geometry, in Figure 2-1 and Section 2.2.1.3. Topic Volume 9 specifies in considerable detail the accuracy data mentioned only briefly elsewhere in the Abstract Specification.

2.4. Corresponding Image and Ground Points

Any point on the ground normally will be imaged at only one point in any one image. Thus any point specified in ground coordinates has one corresponding point in image coordinates. Any point specified in image coordinates in one image usually can correspond to an infinite number of ground position coordinates. In most imaging geometries, these ground positions all lie along one imaging ray.

Obviously, the corresponding image coordinates and horizontal ground coordinates will (almost always) have different reference points and scales. Furthermore, the shapes of objects will be different in image and horizontal ground coordinates. Straight lines in ground coordinates will rarely be perfectly straight in image coordinates. Parallel lines in horizontal ground coordinates will usually not be parallel in image coordinates. Similarly, evenly spaced points along a line will usually not be evenly spaced in image coordinates.

Some of the possible shape differences are indicated in the diagram of Figure 1. Down the center is a cartoon showing a horizontal rectangular feature on the ground with two images of that ground area. The bottom left square indicates how this feature appears in horizontal ground coordinates. The top-left and top-right squares indicate how this feature could appear in two different images. This figure assumes that the two images are tilted significantly from vertical, as is typical with convergent stereoscopic images. Of course, many real images would be tilted less, producing less distortion of the ground rectangle. However, there is usually some significant shape distortion in an image of a ground feature.

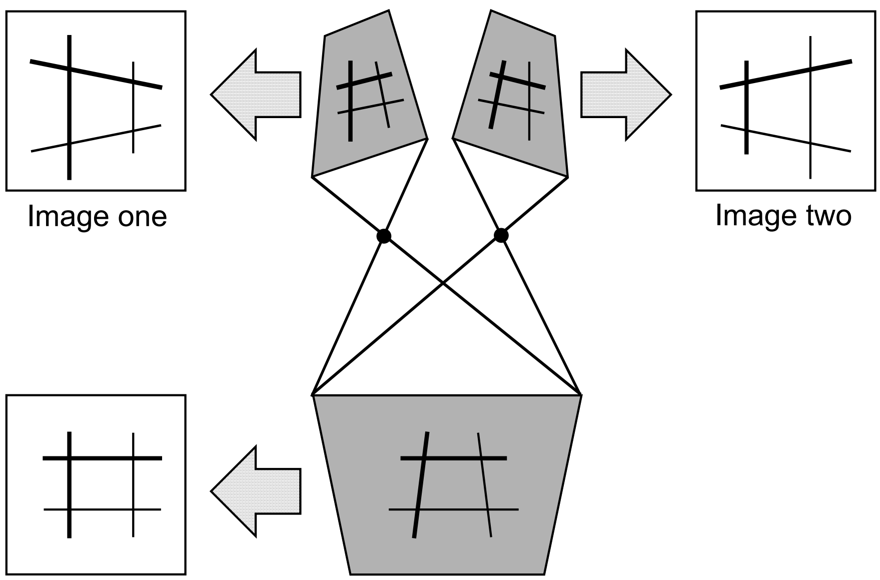

Figure 1. Corresponding Ground and Image Shapes

When high accuracy is needed, image exploitation must consider various image sensor (or camera) non-idealities from a perfect frame (or pinhole) camera. These non-idealities often include lens distortion, atmospheric refraction, and earth curvature, all of which tend to change straight lines into (slightly) curved lines.

2.5. Image Geometry Models

To determine the correct image position of a ground point, an image geometry mathematical model is used. Such an image geometry model relates 3–D ground position coordinates to the corresponding 2–D image position coordinates. Such an image geometry model is alternately called an image sensor model, sensor model, imaging model, or image mathematical model. The term "sensor" is often used when the camera is digital; the word "camera" is usually used when the image is captured on film. Of course, film images can be scanned or digitized and are then "digital". The data used by such an image geometry model is often called image support data.

An image geometry mathematical model can also be used to determine the correct ground position for an image position, if used with additional data. When a single (or monoscopic) image is used, this additional data normally defines the shape and position of the visible ground (or object) surface. For example, this additional data is often grid elevation data. Alternately, two or more stereoscopically overlapping images can be used, that show the same ground point viewed from different directions. In this case, the multiple image geometry mathematical models can be used, with the point coordinates in each image, to determine the corresponding 3–D ground position. (A single IFSAR image can be used to derive the corresponding 3–D ground position, because the gray value represents the distance from the camera to the ground point.)

An accurate image geometry model is more complex than a simple polynomial coordinate transformation, between ground coordinates and image coordinates. This document primarily considers high accuracy image geometry models for general images. Much simpler geometry models are adequate when low accuracy is sufficient, or for images that have been orthorectified. Also, much simpler geometry models can provide medium horizontal accuracy for rectified or nearly-vertical images, when the imaged ground is relatively flat. Many different high accuracy image geometry models can be used, as summarized in Topic Volume 7: The Earth Imagery Case [1].

There are multiple possible geometry models for an image, with different properties. For example, a rigorous geometry model can be accurately adjusted, but has high computation requirements. On the other hand, a non-rigorous or "approximate" geometry model has lesser computation requirements, but cannot be accurately adjusted. Conversion from a rigorous geometry model to a corresponding approximate geometry model for the same image is then sometimes required. However, conversion from an approximate geometry model to a rigorous geometry model is usually not possible or practical.

The term "approximate" image geometry model is used here as a synonym for a "non-rigorous" image geometry model. However, many degrees of accuracy are possible when considering the position errors introduced by the approximation, from large position errors through no detectable position errors. For example, the "Universal Approximate Image Geometry Model" described in Section 6.5 of Topic 7 can be fitted to almost all rigorous models with no significant position error. Similarly, the "Grid Interpolation Image Geometry Model" described in Section 6.3 of Topic 7 can approximate a rigorous model with no significant position errors, if enough grid points are used.

An approximate image geometry model has often been called a "real-time" geometry model, and that is the term now used in Topic Volume 7: The Earth Imagery Case [1]. The need for a fast "real-time" geometry model is decreasing, as computer speeds increase. However, a fast geometry model is still useful in some situations (such as generating complex feature graphics overlaid on an image). On the other hand, an "approximate" image geometry model can be needed or useful for other purposes, such as ignorance or inaccessibility of a rigorous geometry model.

Note: Although Abstract Specification Topic 7 defines both rigorous and "real-time" or approximate image geometry models, Topic 7 currently includes examples of only approximate image geometry models. It would be useful to add examples of rigorous image geometry models to Topic 7, but no volunteer has yet proposed a description of any rigorous model.

2.6. Multiple Image Versions

Multiple versions of the same original image with different image coordinates will usually exist, simultaneously and/or sequentially. Image exploitation can use any one of these image versions, not always the original image. However, there is usually only one basic image geometry model recorded for one image, usually for the original image.

The various versions of an image can have image coordinates that differ from the original image coordinates by:

1. Different origin and range of pixel indices, as a result of copying only a rectangular section of the original image.

2. Different pixel spacings, as a result of generating reduced resolution versions of the original image and/or of changing the apparent image magnification or demagnification. Reduced resolution versions often have pixel spacings that are powers of two times the original pixel spacing. Magnified or demagnified versions can have pixel spacings that are any number times the original pixel spacing. (For example, the magnification factor could be NN.NNN, where N is any decimal digit.)

3. Different rotation, as a result of image rotation by an angle.

4. Affine (first order polynomial) transformation of image coordinates, as a result of affine warping of the image pixels. Such an affine coordinate transformation might use the equations:

$$u = K1 + K2 * x + K3 * y$$

$$v = K4 + K5 * x + K6 * y$$

Where:

x, y = original image coordinate
u, v = transformed image coordinates
Kn = transformation parameter

(Note that this transformation includes, but is more general than, items 1 through 3 listed above.)

5. Image rectification, by mathematical projection from the actual imaging surface through the image perspective point onto another smooth surface. The second smooth surface is often a plane, selected to approximate the average surface of the earth in the region covered by the image. This rectification could include corrections for imaging non-idealities such as lens distortion and atmospheric refraction. (However, this rectification does not include corrections for an irregular shape of the ground, as represented by elevation data. Inclusion of corrections for an irregular ground surface is normally termed *orthorectification* or *differential rectification*, not rectification.)

6. Combination of the above

Some image versions are produced directly from the original image. Other image versions are produced from previously produced image versions. In this case, the image coordinates of the new image version might or might not be represented by a simple coordinate transformation from or to the original image coordinates.

2.7. Orthorectified Images

These image coordinate transformation services do not directly apply to orthorectified images. In an orthorectified image or orthophoto, the effects of the imaging geometry have been removed to the maximum practical extent. Orthorectified image coordinates are thus simply scaled and offset ground coordinates, using a selected ground coordinate reference system. Orthorectified image coordinates could also be rotated ground coordinates, but this is rare. On the other hand, image coordinate transformation services are needed to produce an orthorectified image.

2.8. Multiple Ground Coordinate Systems

Exploitation of an image can be into several different ground coordinate Spatial Reference Systems (SRSs), usually sequentially or perhaps simultaneously. However, the basic image geometry model is normally available and implemented relative to one specific ground SRS. This ground SRS of the image model may not be the only or primary ground SRS needed in exploiting this image.

Therefore, ground coordinate transformation services will often be needed, between the ground SRS used in the image geometry model and the one(s) desired for data from the exploited image, or data used in exploiting the image. These ground coordinate transformation services, and their interfaces, are being specified separately, by the Coordinate Transformation (CT) Working Group (WG) of the OGC. However, these ground coordinate transformation services need to be closely integrated with the image coordinate transformation services, and their interfaces, to effectively support image exploitation.

2.9. Similarities to Ground Coordinate Transformations

Image coordinate transformations have many commonalties with ground coordinate transformations, as discussed in Abstract Specification Topic 2. These commonalties include:

1. Use similar interface data, including position coordinates and metadata about these coordinates

2. Frequently use concatenated coordinate transformations

3. Need many different coordinate transformations

4. Need to use a common software framework, to simplify use of many different coordinate transformations

5. Need metadata about the coordinate transformations

6. Use ground coordinate transformations within image coordinate transformations

However, image coordinate transformations have certain differences from ground coordinate transformations. These differences include:

1. Decrease or increase the number of position coordinate dimensions, by an image coordinate transformation

2. Need many different image coordinate transformations for different image geometry types and different image versions

3. Less standardization now exists of image coordinate transformations, especially of image geometry models

4. Production (and use) of accuracy data is more important, because the errors are often more significant

2.10. Standardization of Image Geometry Models

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. Implementations of all these image geometry models should use one standardized Applications Programming Interface (API) for each distributed computing environment.

2.10.1. Multiple Image Geometry Models

Multiple different image geometry models are needed, for exploiting different image types under different conditions. There are many different types of imaging (or camera) geometries, including frame, panoramic, pushbroom, whiskbroom, Synthetic Aperture Radar (SAR), Interferometric SAR (IFSAR), X-Ray, and (laser or radar) altimetry profiles. Many of these imaging geometries have multiple subtypes, for example, multiple small images acquired simultaneously. These image geometries are sufficiently different that somewhat different rigorous image geometry models are required. Furthermore, different cameras of the same basic geometry can require different rigorous image geometry models.

Technical Question: How are IFSAR and X-Ray image geometries different from other geometry types?

Different image exploitation conditions can also need different image geometry models. For example, an approximate geometry model may be required for real-time use, which need not be rigorous. Fitted general image geometry models may be needed for use with multiple types of basic image geometries, where these general-purpose image geometry models cannot be rigorous.

Multiple different image geometry models need not be completely different. Multiple image geometry models can use common submodels where appropriate. For example, common submodels might be used for corrections that are common to multiple rigorous image geometry models, such as atmospheric refraction and lens distortion.

2.10.2. Standard Interfaces to Image Geometry Models

Although multiple different image geometry models may be needed, in general and within one image exploitation application, a standard Applications Programming Interface (API) is needed for all image geometry models. This standard interface may have to be somewhat different for each image geometry model, but should be the same across all image geometry models to the maximum practical extent. For example, the same interface operation(s) should transform ground to image coordinates for all image geometry models. Similarly, the same interface operation(s) should transform image to ground coordinates. Also, the same interface operation(s) should produce associated data, including absolute and relative accuracy data plus various partial derivatives.

A variety of types of partial derivative are of potential interest, including:

1. Partial derivatives of image coordinates with respect to ground coordinates
2. Partial derivatives of ground coordinates with respect to image coordinates
3. Partial derivatives of image and ground coordinates with respect to (adjustable) image geometry model parameter values

Author’s Note: I assume that such a standard image geometry model API is practical and commercially viable. The technical feasibility of such an API has been largely proven, at least by work done by BAE SYSTEMS (formerly GDE Systems, Inc.) Their SOCET SET® commercial photogrammetric software has used a standard image geometry model API for several years, using the same API for about 30 different image geometry models.

2.10.3. Standard Image Geometry Models

When inter-operation of software from multiple software producers is needed, standard image geometry models need to be specified and used. Such inter-operating software includes the software that produces a specific image geometry model for each image, and the software that uses this image geometry model for image coordinate transformation. In many cases, multiple different software packages must use the same image geometry model, in different image exploitation environments and/or to meet different image exploitation needs. In other cases, multiple different software packages must produce the same type of image geometry model, for use by the same image coordinate transformation software package(s).

A standard image geometry model must specify the form and format in which image geometry data is transferred between different software packages. A standard image geometry model must also specify the semantics of the data transferred. These semantics will often include a set of equations that specify how the transferred data must be interpreted for image coordinate transformation. Although the image coordinate transformation software might directly implement the specified equations, that software will often implement a different but equivalent set of computations. Such equivalent computations will be designed to optimize some quality of the software, such as computation speed.

In this context, an “image geometry model”, together with the values for all the parameters used in that model, completely defines the correct relationship between all possible image positions and the corresponding ground positions. Usually one 2-D image position corresponds to many possible 3-D ground positions.

Such an image geometry model is not a computation algorithm. However, a computer implementation of an image geometry model must use at least one algorithm. One image geometry model can (almost) always be implemented using any of several alternative algorithms. Indeed, different algorithms are normally used for each coordinate transformation direction between image and ground coordinates.

A common way to specify an image geometry model is by giving algebraic equations that could be used to calculate image coordinates from ground coordinates, or vice-versa. Direct implementation of those equations in any programming language comprises one possible algorithm for implementing coordinate transformation in that direction using that image geometry model. However, use of such equations to specify an image geometry model does not require use of the directly corresponding algorithm.

2.10.4. OGC Standardization of Image Geometry Models

The OGC should standardize the image geometry models that are expected to have widespread use by interoperable software. This means that the OGC should plan to standardize multiple different image geometry models, with the number of standardized models probably increasing over time.

In standardizing an image geometry model, the OGC should not standardize an algorithm for implementing that model. The OGC should allow a standard interface conforming service implementation to use any algorithm that always produces equivalent results.

To simplify these standard image geometry models and their implementations, the OGC should also standardize multiple submodels, that will each be used in multiple image geometry models. These submodels might be for common types of corrections, such as atmospheric refraction and lens distortion. The number of standardized submodels may also increase over time.

In defining standard image geometry models and submodels, the OGC probably should cooperate with the International Society of Photogrammetry and Remote Sensing (ISPRS). Several ISPRS commissions are doing related work, including:

Commission I: Sensors, Platforms and Imagery; Chair: G. Joseph (India)

Commission II: Systems for Data Processing, Analysis & Presentation; Chair: I. Dowman

Commission III: Theory and Algorithms; Chair: T. Schenk (USA)

Author's Note: The following text discusses several important questions about how the OGC should proceed toward standardizing image geometry models and geometry model data formats. For each of several needed technical decisions, the OGC TC might make the decision, before issuing the relevant RFP. Alternately, the OGC TC might ask teams responding to the relevant RFP to recommend each decision. Similarly, the OGC TC might let teams submitting RFCs recommend each decision. The OGC TC would then accept the recommended decision when it approves an Implementation Specification using that decision. This Section implies Arliss Whiteside's opinion on many of these needed decisions.

The set of image geometry models (or model types) to be standardized by the OGC might be completely selected by the OGC Technical Committee (TC). Probably better, specific image geometry models to be standardized might be proposed by vendors, and adopted by the OGC. The types of image geometry submodels to be standardized by the OGC might be more completely selected by the OGC TC. These submodels might be organized into categories, such as geometric model (e.g. earth curvature, lens distortion, principal point of autocollimation) and physical model (e.g. atmospheric refraction, radar or laser properties).

The specific image geometry models and submodels standardized might be (largely) specified by the OGC TC (with the ISPRS), perhaps in the form of model equations. Probably better, specific image geometry models might be specified by vendors in response to RFPs or in RFCs. Similarly, the forms used to transfer image geometry model data might be specified by the OGC TC or proposed by vendors. The complete format details for transferring data for each image geometry model should be specified by vendors, in response to a RFP or in a RFC.

2.10.5. Proprietary Image Geometry Models

Although multiple image geometry models should be standardized by the OGC, there will probably also be proprietary image geometry models that are not standardized. These proprietary image geometry models will be used only when the geometry model implementation software need not be interoperable. Although complete image geometry models may be proprietary, submodels are more likely to be proprietary. For example, camera internal geometry models are more likely to be proprietary.

These proprietary image geometry models and/or submodels are likely to be used by vendors who want to protect the nature of their image geometry and/or image geometry model. For these proprietary models, encryption of the model geometry data and/or the implementation software might be used to further conceal the details of the image geometry model.

In order to be useful and used, software implementing the needed image coordinate transformation capabilities using a proprietary image geometry model or submodel probably must:

1. Be provided at no extra cost by the supplier of the images using the model, or by the supplier of geometry model parameter data for specific images, so there is no need for competition between suppliers
2. Be available for all important computing platforms (such as Windows and UNIX)
3. Include capabilities and interfaces for all types of geometry model adjustment needed for image registration, if any such adjustment is needed. (These capabilities and interfaces for model adjustment could be designed to hide the specific nature of the model parameters being adjusted.)
4. Support standard interfaces for access to all provided image coordinate transformation functions, so this software can be easily used with other geometry models and software

To help ensure item 4 above, the OGC probably should work with prospective vendors of proprietary models in the development of standard image geometry model APIs. The OGC should work to ensure that OGC standardized APIs can be used with all anticipated proprietary image geometry models. For example, the standardized APIs should allow vendors to tailor the accuracy, computation speed, and other properties of their proprietary model implementations.

To simplify use and implementation, such proprietary image coordinate transformation software might also be:

1. Coded in a programming language that can be executed in many computing environments (such as JAVA)
2. Available for automatic download when needed, over the Internet or other network (like JAVA Applets)
3. Available on a publicly accessible server for remote execution, over the Internet or other network. (A charge could be levied for such execution.)

3. Essential Model for Image Coordinate Transformation Services

This section presents an essential model of Image Coordinate Transformation (ICT) Services, in the context of the use of these services within general image exploitation.

3.1. Image Coordinate Transformation Services

3.1.1. Image to Ground Coordinate Transformation

When the image positions of one or more points are identified during image exploitation, the ICT Services should transform the image coordinates to the corresponding ground coordinates. The image positions might be either manually or automatically identified, outside the ICT Services. The transformation to ground coordinates might be manually initiated, but will often be automatically initiated by other image exploitation software, when it is in certain operating modes.

The ground coordinates of each point are normally saved or recorded, not the image coordinates. Either the 3-D ground coordinates or the 2-D horizontal ground coordinates might be saved, depending on the application. However, 3-D ground coordinates are usually computed. In many cases, the 3-D or 2-D ground coordinates are saved as vertices of a feature geometry, and the image exploitation services could be directly connected to a vector GIS.

3.1.1.1. Single or Monoscopic Image

When a single (or monoscopic) image is being exploited (for each point), the image coordinates of each point are identified in one image. In order to determine the corresponding ground coordinates, additional information is needed. In general, this additional information must specify the position and shape of the visible ground surface. Often, this additional information is one elevation value, specifying the elevation of a horizontal surface in which the corresponding ground point coordinates (are assumed to) lie. Alternately, an elevation coverage can be used, that defines the (approximate) elevation of visible points at all horizontal positions within a region. Other additional information could alternately be used, such as radar range or laser profiling data.

The elevation information to be used is usually selected before image positions are identified. The same elevation information is often used for a series of image positions, until new elevation information is selected.

3.1.1.2. Multiple or Stereoscopic Images

When multiple (or stereoscopic) images are being exploited, the image coordinates of each point are identified in two or more images. Images of the same ground point in different images are often referred to as *conjugate points* or *homologue points*. If these images view each ground point from different viewing directions, sufficient information is available (or derivable) to determine the corresponding 3-D ground coordinates. Stereoscopic images are two images collected from different positions in space, that cover a common region on the ground. However, more than two images can be used together, to produce more accurate ground coordinates or otherwise better extracted data.

3.1.2. Ground to Image Coordinate Transformation

When the ground positions of one or more points are to be used during image exploitation, the ICT Services should transform the ground coordinates to the corresponding image coordinates, in one or more images. The transformation to image coordinates might be manually initiated, but will often be automatically initiated by other image exploitation software, when it is in certain operating modes.

The ground positions of interest might be manually or automatically identified. The one or more images being exploited are often identified before the desired ground positions. These images are often the same for a series of ground positions, until new images are selected. Alternately, the images in which a ground position appears might be automatically determined, especially if selecting from a limited set of images previously identified.

The image coordinates of each point are often used to display graphics overlaid on the displayed image. The overlaid graphics might represent a previously extracted feature or a measurement cursor. Alternately, the image coordinates can be used to extract one image pixel at (or near) that image position, or to extract a set of image pixels surrounding that image position. The pixels extracted might be displayed or used by some automated process. There are many such automated image exploitation processes, including:

1. Image resampling to produce an orthorectified image
2. Automated image matching

(Note: Most persons not familiar with image exploitation assume that transforming image to ground coordinates is most frequently used. In practice, transforming ground to image coordinates is more frequently used, especially in stereoscopic image exploitation.)

3.1.3. Output Coordinates Accuracy Data

In addition to producing ground or image coordinates, the ICT Services should output data describing the position accuracy of these output coordinates. Position accuracy output data is not useful in all image exploitation operations, and may be rarely needed for image coordinates. However, accuracy data should be automatically produced whenever it is needed by the current image exploitation operation.

This accuracy data should take the form of position error statistical estimates. Absolute error estimates should be available for single points, relative to the defined coordinate SRS. Relative error estimates should be available for pairs of points. These error estimates should be in the form of covariance matrices, as defined and specified in Abstract Specification Topic 9: Quality.

3.1.3.1. Output Accuracy Determination

Error estimates for coordinates output from a transformation should represent the combination of all significant error sources (or error components). In general, three types of error sources need to be considered: input coordinate errors, transformation parameter errors, and transformation computation errors.

The input coordinates used by the transformation will often contain errors, which propagate to the output coordinates. Error estimates for the input coordinates can be propagated to the resulting output error estimates by using the partial derivatives of the individual output coordinates with respect to the individual input coordinates.

The transformation parameters used will often contain errors, which propagate to the output coordinates. Error estimates for the transformation parameters can be propagated to the resulting output error estimates by using the partial derivatives of the individual output coordinates with respect to the individual transformation parameters.

The transformation computations will sometimes introduce significant errors, at several steps in the computation, which propagate to the output coordinates. Typical computation errors can be propagated to the resulting output error estimates by using the partial derivatives of the individual output coordinates with respect to the internal quantities where errors are committed.

3.1.3.2. Input Coordinates Accuracy Data

In order to produce output coordinates accuracy data, the ICT Services will usually need similar accuracy data for the input coordinates. For image coordinates, this input accuracy data should reflect the estimated errors in determining the correct image position. In some cases, the proper image coordinates error estimates will be zero.

3.1.3.3. Coordinate Transformation Accuracy Data

To produce output coordinates position accuracy data, the ICT Services will also generally need accuracy data for the transformation performed. This transformation accuracy data might be for the transformation as a whole, or for the various parameters used in the transformation.

A transformation between ground and image coordinates will have non-zero errors, except for synthetic images. A coordinate transformation between two different image versions will often have zero error estimates, except for computation errors. A coordinate transformation between different ground SRSs will sometimes have significant errors. However, ground coordinate "conversions" will have zero error estimates, except for computation errors.

Computation errors might or might not be negligible even when double precision floating point arithmetic is used in coordinate transformations. For example, the 48 bit precision of a double precision floating point number applied to a longitude of about 180 degrees corresponds to an error of about 8 micrometers. In performing coordinate transformation, several such errors can be committed. Although very rare, some ground coordinates could have similar size errors from all other sources.

Data about the inherent accuracy of a coordinate transformation should be recorded as metadata for that transformation. Data about the computation error in an implementation of a coordinate transformation could be either internal data of the implementation or metadata about that implementation.

3.1.4. Handle Image Versions

The ICT Services should automatically handle image coordinates in any version of the original image that has been created and is being used in image exploitation. The selection or identification of any image must include identification of the particular image version being exploited.

3.1.5. Handle Multiple Ground Coordinate Reference Systems

The ICT Services should automatically handle ground coordinates in any desired ground coordinate Spatial Reference System (SRS). The desired ground SRS for data being extracted from an image must be selected or identified. This will often be done once, for an entire image exploitation session. The ground SRS for each group of existing ground coordinates being used in image exploitation also needs to be identified. Some of this data may be in the same ground SRS as data being extracted. Other existing data will be in other ground SRSs, and must have metadata that specifies its ground SRS.

3.2. Interface Subset Purposes

An image coordinate transformation service should support external interfaces having multiple interface subsets that are designed and used for different purposes. (Indeed, most OGC standardized interfaces are likely to support multiple interface subsets designed for different uses.) These interface subsets for different purposes might be formally defined or not, and could overlap. Some interface subsets, with the associated service capabilities, will be optional and thus not provided by all service implementations.

Explicitly thinking about the possible interface subsets for different purposes appears useful in developing service interface abstract and implementation specifications. These interface subsets might be used by different actors, in one or multiple UML use cases. In some cases, different interface subsets will be used by different client software. In other cases, multiple interface subsets will be used by the same client software.

For an image coordinate transformation service, interface subsets should be designed to support several uses, including:

1. Performing image coordinate transformations, plus closely related computations
2. Importing the image support data needed to subsequently perform image coordinate transformations
3. Service administration
4. Accuracy data conversion
5. Service metadata access
6. Adjustment of the current image coordinate transformation(s)

Of course, the initial interfaces standardized by the OGC might not include all of these interface subsets. The interface purpose subsets listed above are discussed in the following subsections.

3.2.1. Perform Coordinate Transformations

For image coordinate transformation services, the most fundamental interface subset purpose is performing coordinate transformations. This interface subset probably should be considered to include some closely related computation functions. These related computations include computing error estimates for computed coordinates, and determining partial derivative matrices of the coordinate transformation (at specified positions). For image coordinate transformations, the relevant transformations include image-ground, ground-image and image-image transformations, plus concatenated transformations including any of these.

To separate usage and design concerns, this coordinate transformation interface subset assumes that the service has already been initialized and provided with all the (image) support data needed to perform these coordinate transformations. The interface subsets needed to do this are discussed below.

Because this is the most fundamental interface subset, almost all the interface operations defined later in the Abstract Model (see Sections 4.1.6 through 4.1.27) are in this interface subset.

3.2.2. Import Support Data

For image coordinate transformation services, the next critical interface subset purpose is importing the image support data that is required to subsequently perform image coordinate transformations. These interfaces are needed whenever separate software packages (or systems) produce and (later) use this image support data. This image support data includes all the information needed to define the correct relationship between corresponding image and ground coordinates. This interface subset probably should include interfaces for exporting the image support data for later use.

In many cases, the interface subset for importing image support data should be standardized by the OGC. Standardization is needed to allow interoperation between different software packages that may produce and use this image support data. Standardization is also needed to support interoperation between different software packages that may need to perform coordinate transformations for the same images. Standardization is also desirable to allow any of multiple software packages (and systems) to produce image support data for use by (one or more) image coordinate transformation packages.

On the other hand, (complete) OGC standardization of an interface subset to import image support data may not be possible or not be needed for image support data whose contents are proprietary.

To separate usage and design concerns, this import data interface subset may assume that the service has already been initialized. The interface subset needed to initialize the service is discussed below. Alternately, that interface subset might be intimately combined with this interface subset.

The “Image Support Data Conversion” interface operations defined later in the Abstract Model (see Sections 4.3.6 and 4.3.7) are in this interface subset.

3.2.3. Administer Service

In general, one needed interface subset purpose is administration of the service implementation software. Such administration functions might include starting and stopping service software: (a) readiness to execute, (b) communication connection to client software, and/or (c) usage session by client software. Other possible administration functions include setting and checking service operation properties, such as quality of service and service options.

To separate usage and design concerns, this interface subset may be separate from all others. Alternately, this interface subset could be partially or fully combined with the interface subset for importing image support data (see previous section).

3.2.4. Convert Accuracy Data

For image coordinate transformations, one possible interface subset purpose is conversion of accuracy data between different forms. Such conversions may be needed by clients before or after exercising other image coordinate transformation functions.

To separate usage and design concerns, this interface subset may be separate from all others. Alternately, this interface subset could be partially or fully combined with another the interface subset.

The “Accuracy Transformations” interface operations defined later in the Abstract Model (see Section 4.4.6 and 4.4.7) are in this interface subset.

3.2.5. Access Service Metadata

Another generally needed interface subset purpose is accessing metadata that describes the service implementation and interfaces. This metadata could include a variety of service metadata categories, including service location, service access, function description, interface description, input data, output data, output data quality, changeable properties, and metadata quality. For image coordinate transformation services, the available and useful metadata is likely to include metadata describing the general accuracy of the transformation, plus the identification of the specific image version supported. Both metadata retrieval and modification might be supported by this interface subset.

No specific metadata access capabilities are currently defining in the Abstract Model (see Section 4), but are the subject of needed future work (see Section 6).

3.2.6. Adjust Image Coordinate Transformation

For more complex image exploitation, image coordinate transformation services can provide an interface subset that supports effective adjustment of the current image-ground transformation. These interfaces allow changing the current (approximate) image-ground transformation of one or more images to better match each other and other geospatial datasets. This can be done by changing some current image-ground transformation parameters, using the derived from the current positions of one or more points in two or more datasets.

The service interfaces used for this interface subset purpose are likely to include some of the interfaces used for other purposes, especially the interfaces that Perform Coordinate Transformations (see Section 3.2.1). The “Geodata Registration Services” defined in Section 3.6 of Abstract Specification Topic 15: Image Exploitation Services would make use of this interface subset. No specific transformation adjustment operations are currently defining in the Abstract Model (see Section 4).

4. Abstract Specification for Image Coordinate Transformation Services

This section presents an abstract model of the Image Coordinate Transformation (ICT) Services. This abstract model of these services is divided into four parts, based on the similarity of the primary data produced and needed by each service part:

1. Image Coordinate Transformation Services. This primary part provides the basic image coordinate transformation services.
2. Imaging Time Determination Service. This expected future optional part provides access to imaging times.
3. Image Geometry Model Conversion Services. This optional part supports creating a new approximate image geometry model from an existing model.
4. Accuracy Conversion Services. This auxiliary part provides frequently-needed conversions of the form of accuracy data.

The following service descriptions do not describe the contents and formats of the "needed data" and "result data" of each service. The "needed data" could alternately be called inputs, and the "result data" could alternately be called outputs. The "needed data" and "result data" of multiple services are often identical or similar, so the possible contents and formats service of this data are discussed later in Section 5.

The image coordinate transformation services will often use and produce metadata about the coordinates that are manipulated. Metadata is the subject of the Metadata SIG and of Topic 11 of the Abstract Specification. To help define service interactions with metadata, the "needed data" and "result data" items listed are often annotated with "(is metadata for ...)".

4.1. Image Coordinate Transformation Services

4.1.1. Function

The Image Coordinate Transformation Services convert image position coordinates between different Spatial Reference Systems (SRSs). Some service operations may have operations that convert the positions of multiple points, not just one point at a time. An alternate name for these Image Coordinate Transformation Services would be "Image Geometry Model Services".

4.1.2. Service Subtypes

The expected sub-types of Image Coordinate Transformation Services include, but are not necessarily limited to:

1. Image-ground position transformation services:
2. Ground to image position conversion service (3-D to 2-D)
3. Stereoscopic images to ground position conversion service (multiple 2-D to one 3-D)
4. Monoscopic image plus elevation to ground position conversion service (2-D plus elevation to 3-D)
5. Monoscopic image plus other data to ground position conversion service (2-D plus other data to 3-D). (This other data might include laser profiling or radar range data.)

5. Image position transformation services: (2-D to 2-D)
6. Polynomial transformation (and conversion) service
7. Image to rectified image position conversion service
8. Rectified image to image position conversion service

6. Concatenated image coordinate transformation services (including two or more of the above image transformations plus ground coordinate transformations and conversions):
9. 3-D to 2-D concatenated transformation (ground to image)
10. 2-D to 3-D concatenated transformation (image to ground)
11. 2-D to 2-D concatenated transformation (image to image)

Note: Although different interfaces could be used for different numbers of input and output coordinates (as listed in item 3 above), such differentiation of interfaces appears undesirable.

4.1.3. Result Data

The data produced by these Image Coordinate Transformation Services includes:

1. Output point position coordinates, in desired SRS
2. Partial derivatives of output position coordinates with respect to input position coordinates (optional)
3. Metadata for output position coordinates, including: (optional)
4. Output SRS definition (is metadata for output positions)
5. Absolute accuracy estimates for output position coordinates (is metadata for output positions)
6. Relative accuracy estimates for output position coordinates (is metadata for output positions)

Result metadata is optionally returned to client software, depending on how the Image Coordinate Transformation service is called. Similarly, partial derivatives are optionally returned to client software, depending on how the service is called. However, the ability to produce result metadata and partial derivatives when requested are required capabilities of these services.

4.1.4. Needed Data

The data needed for use by these Image Coordinate Transformation Services includes:

1. Input point position coordinates, in another SRS
2. Output SRS definition (is metadata for output positions)
3. Image coordinate transformation parameters (optional) (is metadata for SRS or transformation)
4. Transformation accuracy estimates, for each SRS transformation (when output accuracy is needed) (is metadata for transformation)
5. Ground shape and position (or elevation) data (for monoscopic image to ground) (could be considered metadata for an image?)
6. Elevation accuracy estimates (when output accuracy is needed) (is metadata for elevation data)
7. Metadata for input position coordinates, including:
8. Input SRS definition (is metadata for input positions)
9. Absolute accuracy estimates for input position coordinates (when output absolute accuracy is needed) (is metadata for input positions)
10. Relative accuracy estimates for input position coordinates (when output relative accuracy is needed) (is metadata for input positions)

4.1.5. Discussion

These Image Coordinate Transformation Services are considered an Image Exploitation Services category separate from Ground Coordinate Transformation Services in order to limit the size of service categories. However, these two service categories require very similar interfaces, and they must be able to inter-operate easily. Specifically, the concatenated image coordinate transformation services (see item 3 in Section 4.1.2 above) must be able to include individual or concatenated Ground Coordinate Transformation Services.

When an Image Coordinate Transformation Service is needed, the corresponding Service for the opposite conversion direction will often also be needed. Instead of requiring a client to handle separate Image Coordinate Transformation Services for each direction, it appears desirable to automatically link the corresponding Services for the two directions. This linking might be done in several ways, including:

1. Have each Image Coordinate Transformation Services provide transformations in both directions. Different service operations or an additional input to certain operations would be used to select which transformation direction is requested.

2. Provide an Image Coordinate Transformation Service with an additional operation to obtain the reverse direction Service, or to obtain all the metadata needed by such a service.

4.1.6. Object Model

Figure 4-1 and Figure 4-2 are object models for the combination of Image Coordinate Transformation Services and the Ground Coordinate Transformation Services, in the form of a UML class diagram. The classes added to perform Image Coordinate Transformation Services are in the Image Transformations package, that uses the Coordinate Transform package which performs Ground Coordinate Transformation Services.

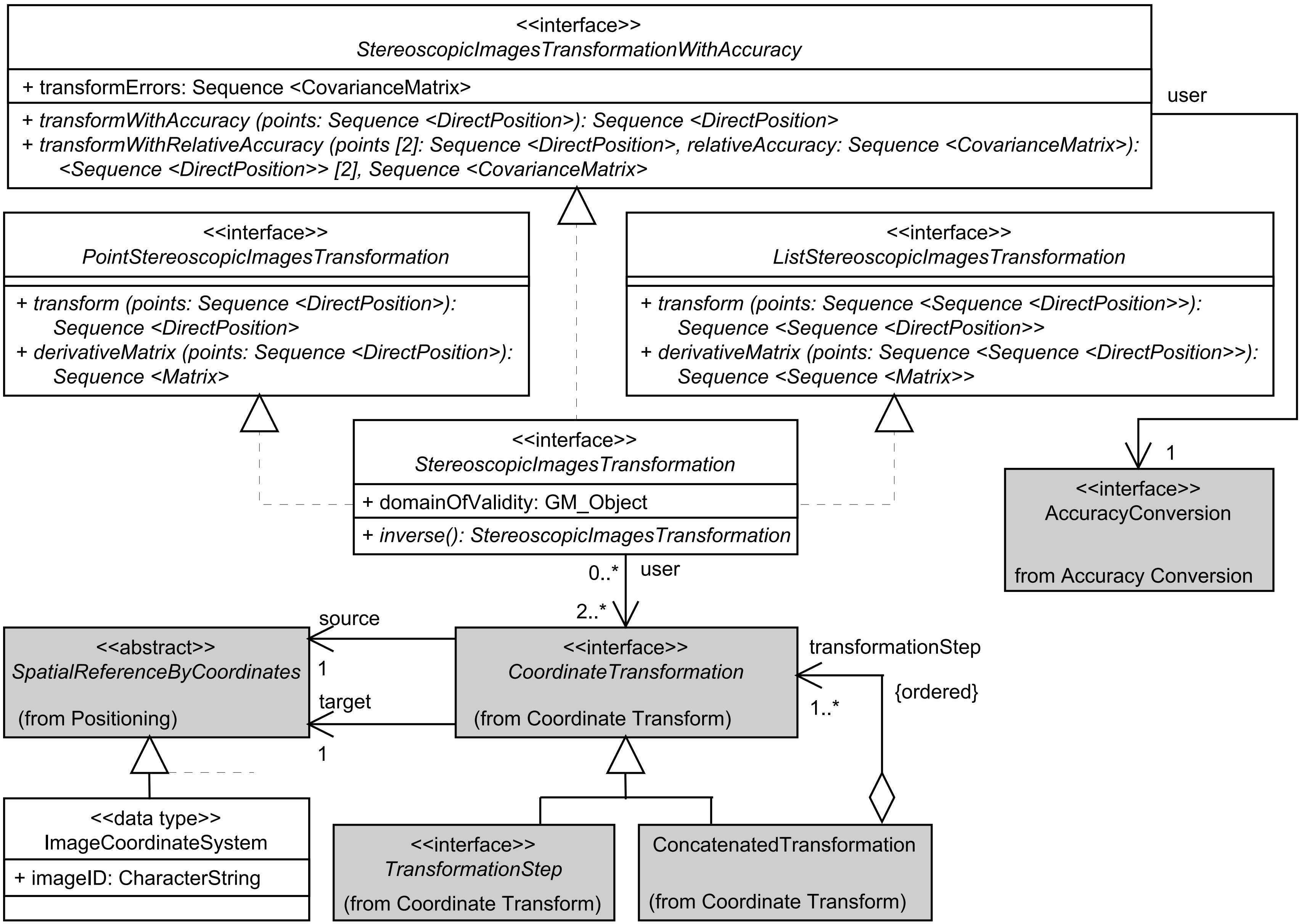

Figure 4-1 Object Model of Image Transformations Package

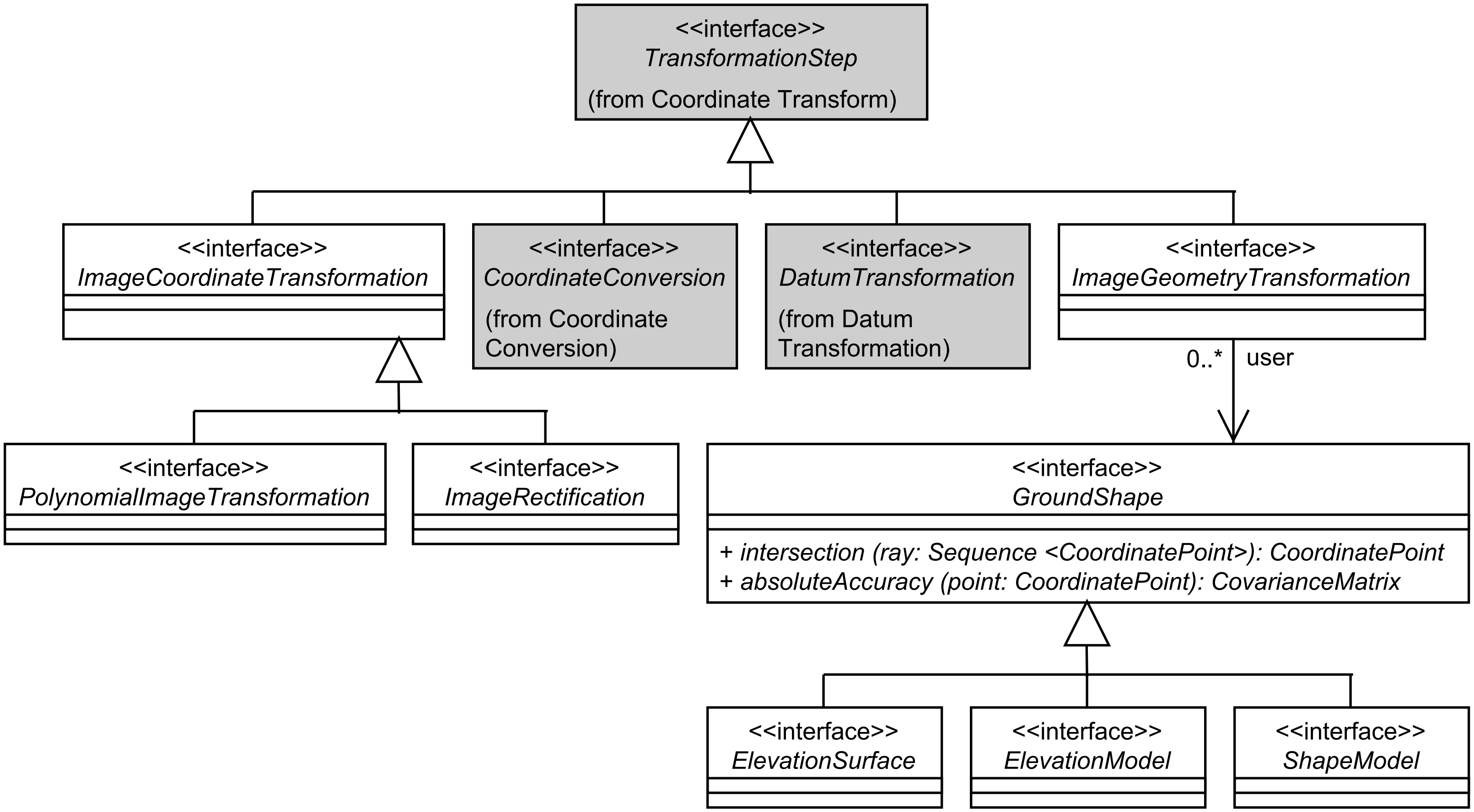

Figure 4-2. Object Model of Image Transformations Package, continued

This class diagram is based on the diagram for the Coordinate Transform package, developed by the Coordinate Transformation Working Group (CT WG). That class diagram has been augmented to show how the Image Coordinate Transformation Services should be combined with the Ground Coordinate Transformation Services. The key additions included in this diagram are the:

1. Stereoscopic Images Transformation class, whose objects each use two or more objects of the Coordinate Transformation class. This class provides capabilities for using two or more stereoscopic images together, to derive 3-D ground position coordinates without using ground elevation or shape data.
2. Point Stereoscopic Images Transformation, List Stereoscopic Images Transformation, and Stereoscopic Images Transformation With Accuracy classes. These classes define interfaces for different purposes, that are realized by the Stereoscopic Images Transformation class.
3. Image Coordinate System subclass of the Spatial Reference By Coordinates class, that defines the coordinate reference system for a Coordinate Transformation object which transforms coordinates to or from image coordinates. This Image Coordinate System class also defines the source and target coordinate reference systems for a Coordinate Transformation object that transforms between two different image coordinate systems.
4. Image Geometry Transformation subclass of the Transformation Step class, that provides transformations between image and ground coordinates.
5. Image Coordinate Transformation subclass of the Transformation Step class, that provides transformations between different image coordinates for the same original image.
6. Polynomial Transformation and Image Rectification subclasses of the Image Coordinate Transformation class.
7. Ground Shape class, that defines the shape and position of the ground surface for monoscopic image to ground transformation operations of the Image Geometry Transformation class.
8. Elevation Surface, Elevation Model, and Shape Model subclasses of the Ground Shape class, that use different forms of shape data.

Author's Note: The six shaded classes shown in Figure 2 are intended to be the same as the classes with the same names in the Transformation package of the CT abstract model, see Abstract Specification Topic 2. These classes include CoordinateTransformation, ConcatenatedTransformation, TransformationStep (shown twice), and SpatialReferenceByCoordinates. The IES SIG and CT WG plan to work together to modify this diagram and the Transformation package to make the shared classes essentially identical.

To keep this diagram simple and easier to understand, Figure 2 omits certain parts of the Transformation package diagram. The Transformation diagram elements omitted in this diagram include:

1. The MathTransform and Parameter classes, and their association with the TransformationStep class (that implement TransformationStep objects)

2. The association of the Parameter class to the CovarianceElement class (that holds absolute and relative accuracy data for the Parameters)

3. Two notes

Some of the classes shown in Figure 2 may have additional attributes and operations, not shown here. For example, all subclasses inherit all the abstract operations included in their ancestor classes, but those operations are not repeated in this diagram. Also, several classes are expected to have subclasses that are not specifically identified and not shown in this overview diagram. The classes expected to have (additional) subclasses include:

1. Stereoscopic Images Transformation, for different numbers of images and for different methods of computing absolute and relative accuracies

2. Image Geometry Transformation, for different image geometry models

3. Polynomial Transformation, using different orders of polynomials

4. Image Rectification, for different shapes of rectification surfaces

5. Elevation Surface, using different smooth elevation surfaces

6. Elevation Model, using different forms of elevation models

7. Shape Model, using different forms of additional shape data

The following sections provide more information about each image transformations package class shown in Figure 4-1 and Figure 4-2.

4.1.7. Class Name: StereoscopicImagesTransformation

Package Class Belongs to: Image Transformations

Documentation:

This service class determines the 3-D ground position corresponding to the specified positions in two or more stereoscopic images, by using the corresponding image geometry models. This class alternately determines the positions in two or more stereoscopic images corresponding to the specified 3-D ground position. This class provides a set of operations similar to the Coordinate Transformation class, and the three interface classes realized by that class, adapted to using two or more stereoscopic images. This class uses an object of the Coordinate Transformation class for each image, to transform between the corresponding image and ground positions.

Subclasses:

Subclasses are expected to be defined using different methods of computing absolute and relative accuracies. Subclasses could also be defined using different ray intersection algorithms and/or different numbers of stereoscopic images, such as only two images. The subclasses support identical interfaces except for any parameters needed by different methods of computing absolute and relative accuracies.

Superclass: none

Realizes Interfaces:

PointStereoscopicImagesTransformation

ListStereoscopicImagesTransformation

StereoscopicImagesTransformationWithAccuracy

Note: The use of realization is similar to but not equal to inheritance. An Interface does not constitute a complete class definition, but defines some basic class-like behavior, such as a set of attribute, association, and/or operation signatures that form a logically consistent group. In this model, realization is used to logically group signatures into three functionally related groups:

- transformations that transform one point at a time
- transformations that transform a list of points at a time
- transformations that determine the accuracy of transformed points

In any implementation of this model, these 3 groups shall be handled as units, either implementing all of the operations in a group or none of them.

Stereotype: interface

Associations: user (of 2..*) CoordinateTransformation

Attributes:

domainOfValidity: GM_Object
The domainOfValidity attribute is a geometry that defines the geographical region where this Stereoscopic Images Transformation object is appropriate to use. The coordinate reference system of this geometry may be a source, target, or any standard coordinate reference system.

This OGC geometry object supports the GM_Object interface. The GM_Object interface, as described in Topic 1: Feature Geometry, includes operations that will check whether or not a specified point or geometry is spatially within the domain of this GM_Object.

Operations:

inverse(): StereoscopicImagesTransformation
This operation returns another Stereoscopic Images Transformation object that reverses the actions of this Stereoscopic Images Transformation object. The target of the inverse transformation is the source of the original, and the source of the inverse transformation is the target of the original. Using the original transform followed by the inverse transform will result in an identity map on the source coordinate space, when allowances for error are made.

4.1.8. Class Name: PointStereoscopicImagesTransformation

Package Class Belongs to: Image Transformations

Documentation:

This class defines a coordinate transformation operation that determines one 3-D ground position corresponding to the specified positions in two or more stereoscopic images. This class alternately defines an operation that transforms the coordinates of one ground point position into the corresponding image coordinates in two or more images.

Note that the input and output positions are of type DirectPosition, in order to include the association to the proper SpatialReferenceByCoordinates (which is needed with some input and output positions). However, each DirectPosition includes position accuracy data, which is not needed with the input and output positions.

Superclass: none

Stereotype: interface

Associations: none

Attributes: none

Operations:

transform (points: Sequence <DirectPosition>): Sequence <DirectPosition>
This operation performs one of two similar functions, depending on whether the source SpatialReferenceByCoordinates is a ground or image coordinate system.

When the source is a ground coordinate system, this operation transforms the specified position of one point given in the source ground coordinate system to the corresponding positions of the point in two or more images.

When the source is an image coordinate system, this operation computes the one position in the (source) ground coordinate system that best fits the specified positions of this point in each of two or more images.

derivativeMatrix (points: Sequence <DirectPosition>): Sequence<Matrix>
This operation performs one of two similar functions, depending on whether the source SpatialReferenceByCoordinates is a ground or image coordinate system.

When the source is a ground coordinate system, this operation returns a list of matrices of the partial derivatives of the image coordinates with respect to the source ground coordinates. These partial derivatives are determined at the one specified position in the source ground coordinate reference system. The returned list of partial derivative matrices includes a matrix for each image used by this object.

When the source is an image coordinate system, this operation returns a list of matrices of the partial derivatives of the ground coordinates with respect to the image coordinates in two or more images. These partial derivatives are determined at the specified positions in these images, which are normally the positions of the same ground point in the different images.

4.1.9. Class Name: ListStereoscopicImagesTransformation

Package Class Belongs to: Image Transformations

Documentation:

This class defines a coordinate transformation operation that determines a list of one or more 3-D ground positions, each corresponding to the specified positions in two or more stereoscopic images. This class alternately defines an operation that transforms one or more ground point position coordinates, each point into the corresponding image coordinates in two or more images. The lists can contain one or more points.

Note that the input and output positions are of type DirectPosition, in order to include the association to the proper SpatialReferenceByCoordinates (which is needed with some input and output positions). However, each DirectPosition includes position accuracy data, which is not needed with the input and output positions.

Superclass: none

Stereotype: interface

Associations: none

Attributes: none

Operations:

transform (points: Sequence <Sequence <DirectPosition>>): Sequence <Sequence <DirectPosition>>
This operation performs one of two similar functions, depending on whether the source SpatialReferenceByCoordinates is a ground or image coordinate system. In both cases, the output list contains the same number of points in the same order as the input list(s) (no densification is performed).

When the source is a ground coordinate system, this operation transforms the specified list of one or more point positions in the source ground coordinate system to the corresponding positions of each point in two or more images.

When the source is an image coordinate system, this operation computes the list of one or more positions in the (source) ground coordinate system that best fit the specified positions of this point in each of two or more images.

derivativeMatrix (points: Sequence <Sequence <DirectPosition>>): Sequence <Sequence <Matrix>>
This operation performs one of two similar functions, depending on whether the source SpatialReferenceByCoordinates is a ground or image coordinate system.

When the source is a ground coordinate system, this operation returns a list of matrices of the partial derivatives of the image coordinates with respect to the source ground coordinates. These partial derivatives are determined at the one or more specified positions in the source ground coordinate reference system. For each ground point, the list of partial derivatives matrices includes a matrix for each image used by this object.

When the source is an image coordinate system, this operation returns a list of matrices of the partial derivatives of the ground coordinates with respect to the image coordinates in two or more images. These partial derivatives are determined at the specified positions in these images, which are normally the positions of the same ground point in the different images. This is done for one or more points in the ground coordinate system.

4.1.10. Class: StereoscopicImagesTransformationWithAccuracy

Package Class Belongs to: Image Transformations

Documentation:

This class defines coordinate transformation operations that also compute error estimates for transformed point position coordinates. For image coordinates, these operations accept or produce a list of image coordinates in two or more stereoscopic images.

The output accuracy data always combines the error contributions from all known error sources, including:

- Position errors in the input point coordinates, input in “DirectPosition”
- Errors in the transformation parameters, recorded in Covariance Element objects associated with objects of the “Parameter” class

- Errors in the transformation computations, recorded in the transformErrors attribute of this class

Note that the input and output positions are of type DirectPosition, in order to include position accuracy data and the association to the proper SpatialReferenceByCoordinates (which is needed with the input and output positions).

Superclass: none

Stereotype: interface

Associations: none

Attributes:

transformErrors: Sequence <CovarianceMatrix>
Estimates of the computation errors committed in performing the various "transform" operations, by the specific implementation of the specific subclass of this class. These Covariance Matrices reflect the errors in various coordinate systems, including the source and target coordinate systems.

Note: In some cases, these error estimates could include the combined effects of the error estimates in some or all the parameter values within a set of "Parameter" objects. This set is one used by one or more objects of the TransformationStep class, such as for a set of images that were triangulated together. Such inclusion would replace the CovarianceElement accuracy data otherwise associated with objects of the "Parameter" class.

Operations:

transformWithAccuracy (points: Sequence <DirectPosition>): Sequence <DirectPosition>
This operation performs one of two similar functions, depending on whether the source SpatialReferenceByCoordinates is a ground or image coordinate system.

When the source is a ground coordinate system, this operation transforms the specified position of one point in the source ground coordinate system to the corresponding positions of the point in two or more images. This operation also computes absolute position error estimates (or absolute accuracy) for each set of image coordinates returned by this operation.

When the source is an image coordinate system, this operation computes the one position in the (source) ground coordinate system that best fits the specified positions of this point in each of two or more images. This operation also computes absolute position error estimates (or absolute accuracy) for the ground coordinates returned by this operation.

transformWithRelativeAccuracy (points [2]: Sequence <DirectPosition>, relativeAccuracy: Sequence <CovarianceMatrix>):
<Sequence <DirectPosition>> [2], Sequence <CovarianceMatrix>
This operation performs one of two similar functions, depending on whether the source SpatialReferenceByCoordinates is a ground or image coordinate system.

When the source is a ground coordinate system, this operation transforms the specified positions of two points in the source ground coordinate system to the corresponding positions of each point in two or more images. This operation also computes relative position error estimates (or relative accuracy) between the corresponding pairs of image CoordinatePoints returned by this operation. The input and output Covariance Matrices specify the relative accuracies between the specified pair of points.

When the source is an image coordinate system, this operation computes two positions in the (source) ground coordinate system that best fit the specified image positions of these points, each point in two or more images. This operation also computes relative position error estimates (or relative accuracy) between the two CoordinatePoints returned by this operation. The input and output Covariance Matrices specify the relative accuracies between the pairs of points.

4.1.11. Class Name: ImageCoordinateSystem

Package Class Belongs to: Image Transformations

Documentation:

Defines the image coordinate reference system for a specific version of a specific image, and (probably) associates this image coordinate system with that version of that image.

Superclass: SpatialReferenceByCoordinates

Stereotype: data type

Associations: only inherited associations

Attributes:

imageID: CharacterString
Identifies the specific version of a specific image.

Operations: none

4.1.12. Class Name: ImageCoordinateTransformation

Package Class Belongs to: Image Transformations

Documentation:

This class transforms image position coordinates between two different image coordinate reference systems, usually based on the same original image. Such transformations are required when different images are derived from an original image, such as by image cutting, magnification, rotation, and/or rectification. This class is applicable to 2-D images (not to 3-D images).

Superclass: TransformationStep

Stereotype: interface

Associations: only inherited associations

Attributes: only inherited attributes

Operations: only inherited operations

Note that the CoordinatePoint and DirectPosition inputs to and outputs from all inherited operations are 2-D (not 3-D).

4.1.13. Class Name: PolynomialIMageTransformation

Package Class Belongs to: Image Transformations

Documentation:

This class transforms image positions in one image version to the corresponding position in a different image version, where the transformation is based on low order polynomial functions of two image axes. This class supports image versions that are reduced resolutions, patches, segments, magnifications, rotations, and/or more complex changes.

Subclasses:

Subclasses are expected to be defined using different orders of polynomials, such as first degree and second degree polynomials. The subclasses support identical interfaces except for the parameters needed by different orders of polynomials.

Superclass: ImageCoordinateTransformation

Stereotype: interface

Associations: only inherited associations

Attributes: only inherited attributes

Operations: only inherited operations

Note that the CoordinatePoint and DirectPosition inputs to and outputs from all inherited operations are 2-D (not 3-D).

4.1.14. Class Name: ImageRectification

Package Class Belongs to: Image Transformations

Documentation:

This class transforms image positions in one image version to the corresponding position in a different image version, where the transformation is based on image rectification.

Subclasses:

Subclasses are expected to be defined using different types of rectification surfaces, such as horizontal plane, tilted plane, and spheroid surfaces. The subclasses support identical interfaces except for any parameters needed by different types of rectification surfaces.

Superclass: ImageCoordinateTransformation

Stereotype: interface

Associations: only inherited associations

Note that this class might use the Image Geometry Transformation class, but such use is not required and is not visible to the client of this class.

Attributes: only inherited attributes

Operations: only inherited operations

Note that the CoordinatePoint and DirectPosition inputs to and outputs from all inherited operations are 2-D (not 3-D).

4.1.15. Class Name: ImageGeometryTransformation

Package Class Belongs to: Image Transformations

Documentation:

This class transforms position coordinates between an image coordinate system and a three-dimensional ground coordinate reference system. The image coordinate system is usually two-dimensional, but might be three-dimensional in the future. Three-dimensional ground coordinates can be transformed to image coordinates (either 2-D or 3-D). (However, 2-D ground coordinates cannot be transformed to either 2-D to 3-D image coordinates.) Two-dimensional image coordinates can be transformed to ground coordinates (2-D to 3-D) only when additional information is supplied in an associated Ground Shape object.

Subclasses:

Subclasses are expected to be defined using different image geometry models. The subclasses support identical interfaces except for the parameters needed by different image geometry models.

Superclass: TransformationStep

Stereotype: interface

Associations: user (of 0..1) GroundShape

Attributes: only inherited attributes

Operations: only inherited operations

Note that the CoordinatePoint and DirectPosition inputs to and outputs from all inherited operations are usually 2-D (not 3-D) when they refer to image coordinates.

4.1.16. Class Name: GroundShape

Package Class Belongs to: Image Transformations

Documentation:

This service class provides access to data describing the shape and position of the visible ground surface. This data is used to determine the ground position corresponding to the image position in one (monoscopic) image. Two-dimensional image coordinates can be transformed to ground coordinates (2-D to 3-D) only when additional information is supplied in an associated GroundShape object. This additional information can take the form of an elevation surface, elevation model coverage, or other ground shape (and position) data. This additional data is provided by or stored in more specific subclasses of this class.

Superclass: none

Stereotype: interface

Associations: used (by 0..*) ImageGeometryTransformation

Attributes: none

Operations:

intersection (ray: Sequence <CoordinatePoint>): CoordinatePoint
Compute the ground position at which the specified imaging ray intersects the ground shape data. The imaging ray is defined by a list of two or more points lying along that ray.

absoluteAccuracy (point: CoordinatePoint): CovarianceMatrix
Obtain absolute position error estimates for the specified ground position previously determined. This elevation accuracy data is in the form of the applicable elements of a covariance matrix for the ground position where the imaging ray intersects the GroundShape.

4.1.17. Class Name: ElevationSurface

Package Class Belongs to: Image Transformations

Documentation:

This service class provides access to the elevation of a smooth surface being used to define the approximate shape and position of the surface visible in an image.

Subclasses:

Subclasses are expected to be defined using different elevation surfaces, such as horizontal plane, tilted plane, and spheroid. The subclasses support identical interfaces except for any parameters needed by different forms of elevation models.

Superclass: GroundShape

Stereotype: interface

Associations: only inherited associations

Attributes: none

Operations: only inherited operations

4.1.18. Class Name: ElevationModel

Package Class Belongs to: Image Transformations

Documentation:

This service class provides access to an elevation model coverage being used to define the (approximate) shape and position of the surface visible in an image. This class is expected to use an OGC coverage object (or to be a coverage object).

Subclasses:

Subclasses are expected to be defined using different forms of elevation model, such as grid elevation, TIN, and 3-D feature geometries. The subclasses support identical interfaces except for any parameters needed by different forms of elevation models.

Superclass: GroundShape

Stereotype: interface

Associations: only inherited associations

Attributes: none

Operations: only inherited operations

4.1.19. Class Name: ShapeModel

Package Class Belongs to: Image Transformations

Documentation:

This service class provides access to a ground shape model being used to define the (approximate) shape and position of the surface visible in an image.

Subclasses:

Subclasses are expected to be defined using different forms of additional data, such as IFSAR and radar or laser profiles. The subclasses support identical interfaces except for any parameters needed by the different forms of shape models.

Superclass: GroundShape

Stereotype: interface

Associations: only inherited associations

Attributes: none

Operations: only inherited operations

> Note: The following classes, in subsections 4.1.20 through 4.1.27, are intended to be essentially the same as the classes with the same names in the Transformation package of Topic 2. The descriptions of these classes are included here for easy reference by readers of this document. Some of the text descriptions have been edited in an attempt to make them more understandable or more precise.

4.1.20. Class Name: CoordinateTransformation

Package Class Belongs to: Coordinate Transform

Documentation:

This service interface transforms a coordinate point position between two different coordinate reference systems. A coordinate transformation object establishes an association between a source and a target coordinate reference system, and provides operations for transforming coordinates in the source coordinate reference system to coordinates in the target coordinate reference system. These coordinate systems can be ground or image coordinates. In general mathematics, "transformation" is the general term for mappings between coordinate systems (see tensor analysis).

For ground coordinate points, if the transformation depends only on mathematically derived parameters (as in a cartographic projection), then this is an ISO conversion. If the transformation depends on empirically derived parameters (as in datum transformations), then this is an ISO transformation.

Superclass: none

Realizes Interfaces:

PointTransformation

ListTransformation

TransformationWithAccuracy

Note: The use of realization is similar to but not equal to inheritance. An Interface does not constitute a complete class definition, but defines some basic class-like behavior, such as a set of attribute, association, and/or operation signatures that form a logically consistent group. In this model, realization is used to logically group signatures into three functionally related groups:

- transformations that transform one point at a time

- transformations that transform a list of points at a time
- transformations that determine the accuracy of transformed points

In any implementation of this model, these 3 groups shall be handled as units, either implementing all of the operations in a group or none of them.

Stereotype: interface

Associations:

from (1) SpatialReferenceByCoordinates in association: source

to (1) SpatialReferenceByCoordinates in association: target

used by (0..*) ConcatenatedTransformation

Attributes:

transformationName: Name
This attribute is the identifier of this coordinate transformation.

domainOfValidity: GM_Object
The domainOfValidity attribute is a geometry that defines the geographical (or image) region where this Coordinate Transformation object is appropriate to use. The coordinate reference system of this geometry may be the source, target, or any standard coordinate reference system. (When image coordinates are the source SpatialReferenceByCoordinates, the coordinate reference system will usually be the image coordinates reference system.)

This OGC geometry object supports the GM_Object interface. The GM_Object interface, as described in Topic 1: Feature Geometry, includes operations that will check whether or not a specified point or geometry is spatially within the domain of this GM_Object.

Operations:

inverse (): CoordinateTransformation
This operation returns another CoordinateTransformation object that reverses the actions of this CoordinateTransformation object. The target of the inverse transformation is the source of the original, and the source of the inverse transformation is the target of the original. Using the original transform followed by the inverse transform will result in an identity map on the source coordinate space, when allowances for error are made.

4.1.21. Class Name: PointTransformation

Package Class Belongs to: Coordinate Transform

Documentation:

This interface class defines coordinate transformation operations that transform single point position coordinates between two different coordinate reference systems.

Superclass: none

Stereotype: interface

Associations: none

Attributes: none

Operations:

transform (point: CoordinatePoint): CoordinatePoint
This operation transforms the coordinates of one point given in the source coordinate system to the coordinates of the same point in the target coordinate system.

derivativeMatrix (point: CoordinatePoint): Matrix
This operation determines the matrix of partial derivatives of the transformation result coordinates with respect to the input coordinates, at the specified position in the source coordinate reference system. This matrix is the multi-dimensional equivalent of a derivative or a tangent.

4.1.22. Class Name: ListTransformation

Package Class Belongs to: Coordinate Transform

Documentation:

This interface class defines coordinate transformation operations that transform a list of point position coordinates for each call, between two different coordinate reference systems. The list can contain one or more points.

Superclass: none

Stereotype: interface

Associations: none

Attributes: none

Operations:

transform (points: Sequence <CoordinatePoint>): Sequence <CoordinatePoint>
This operation transforms the coordinates of a list of one or more points given in the source coordinate system to the corresponding list of positions in the target coordinate system. The returned list contains the same number of points in the same order as the input list (no densification is performed).

derivativeMatrix (points: Sequence <CoordinatePoint>): Sequence <Matrix>
This operation determines matrices of the partial derivatives of the transformation result coordinates with respect to the input coordinates. Each matrix is the multi-dimensional equivalent of a derivative or a tangent. This operation is similar to the corresponding operation in the PointTransformation Interface, but works on a specified list of positions in the source coordinate reference system.

4.1.23. Class Name: TransformationWithAccuracy

Package Class Belongs to: Coordinate Transform

Documentation:

This interface class defines operations that compute error estimates for transformed point position coordinates. These operations also transform the point position coordinates. The input and output positions are of type DirectPosition, since this type includes position error data. However, each DirectPosition is associated with a SpatialReferenceByCoordinates, which is not needed with the input and output positions.

The output accuracy data always combines the error contributions from all known error sources, including:

- Position errors in the input point coordinates, input in "DirectPosition" objects
- Errors in the transformation parameters, recorded in Covariance Element objects associated with objects of the "Parameter" class
- Errors in the transformation computations, recorded in the transformErrors attribute of this class

Superclass: None

Stereotype: interface

Associations: user (1) AccuracyConversion

Attributes:

transformErrors: Sequence <CovarianceMatrix>
This attribute contains estimates of the computation errors committed in performing the "transform..." operations, by the specific implementation of the specific subclass of the Coordinate Transformation class. These Covariance Matrices reflect the errors in various coordinate systems, including the source and target coordinate systems.

Note: In some cases, these error estimates could include the combined effects of the error estimates in some or all the parameter values within a set of "Parameter" objects. This set is one used by one or more objects of the TransformationStep class, such as for a set of images that were triangulated together. Such inclusion would replace the Covariance Element accuracy data otherwise associated with objects of the "Parameter" class.

Operations:

transformWithAccuracy (point: DirectPosition): DirectPosition
This operation transforms the specified position of a point in the source coordinate system to the corresponding position in the target coordinate system. This operation also computes absolute position error estimates (or absolute accuracy) for the transformed position returned by this operation.

transformWithRelativeAccuracy (points [2]: DirectPosition, relativeAccuracy: CovarianceMatrix): DirectPosition [2], CovarianceMatrix
This operation transforms the specified positions of two points in the source coordinate system to the corresponding two positions in the target coordinate system. This operation also computes relative position error estimates (or relative accuracy) between the two transformed positions returned by this operation. The input and output Covariance Matrices specify the relative accuracies between the specified pair of points.

4.1.24. Class Name: ConcatenatedTransformation

Package Class Belongs to: Coordinate Transform

Documentation:

Transforms position coordinates of one or more points from a source coordinate reference system to a target coordinate reference system using a sequence of two or more transformation steps.

For converting between image and ground coordinates, a concatenated transformation combines zero or more Image Coordinate Transformations, exactly one Image Geometry Transformation, and zero or more ground coordinate transformations.

Superclass: CoordinateTransformation

Stereotype: none

Associations:

transformationStep (1..) {ordered}* *CoordinateTransformation*
A ConcatenatedTransformation is a sequence of transformationSteps, each being either a single step or another ConcatenatedTransformation. Since 1-long sequences do not cause any logical inconsistency at the behavioral level, this association was not restricted to "2 or more steps," even though that would be the naïve conclusion from the class name. The current multiplicity should also make editing of the sequence easier, since during an edit the list may well drop to 1-long. Viable implementation could allow the list to drop to 0-long with the assumption that that would constitute the identity transformation.

Attributes: only inherited attributes

Operations: only inherited operations

4.1.25. Class Name: TransformationStep

Package Class Belongs to: Coordinate Transform

Documentation:

This class transforms the position coordinates of one or more points from a source coordinate reference system to a target coordinate reference system in one step. This transformation or conversion is often used as one step in a more complex transformation used to transform a set of coordinates.

Superclass: CoordinateTransformation

Stereotype: interface

Associations:

mathTransform (1) *MathTransform in association: uses parameters (0..*) Parameter*

Attributes: only inherited attributes

Operations: only inherited operations

4.1.26. Class Name: MathTransform

Package Class Belongs to: Stored Functions

Note: The Stored Functions package contains classes for implementation of Stored Functions. A Stored Function is an object that uses a finite set of data points or values and a interpolation scheme to implement a function that potentially has an infinite domain.

Documentation:

This service class transforms one set of coordinates to another set of coordinates using a mathematical or stored function. A MathTransform can support both transformation and conversion (without error) operations. The same MathTransform can be used by multiple TransformationStep objects, using different sets of parameter values for each.

Superclass: none

Stereotype: interface

Associations:

user (1) TransformationStep in association: uses
parameter (0..*) Parameter

Attributes:

inputDimension : Integer
Number of dimensions in source (input) coordinates.

outputDimension : Integer
Number of dimensions in target (output) coordinates.

Operations:

initialize (parameters: Sequence <Parameter>): void
This operation initializes the Math Transformation object so it can be used correctly. The initialize operation is often called from the object constructor to fill in appropriate attribute values based on the constructor's parameters.

An uninitialized MathTransform will use default values for all parameters. During initialize, the MathTransform object will change the values of all parameters passed in, and all parameters dependent on the passed parameters. Unreferenced parameters will retain their previous values (which may or may not be the default, if this object was previously initialized).

For example, if the transform is an Affine Transformation, and the parameters are the various rotations of a unit matrix, then initialize will calculate the elements of the Affine Transformation matrix.

One use of the initialize operation is to reuse a MathTransform object without reconstruction, such as changing a single parameter to obtain the inverse MathTransform. (This operation may be used in conjunction with a copy constructor when the original MathTransform object is still needed in its current state.)

transform (source: Vector): Vector
This operation transforms the coordinates of one point in the source coordinate reference system to the coordinates of the same point in the target coordinate reference system.

derivativeMatrix (point: Vector): Matrix
This operation determines the matrix of partial derivatives of the transformation result

coordinates with respect to the input coordinates, at the specified input position. This matrix is the multi-dimensional equivalent of a derivative or a tangent.

inverse ():MathTransform
This operation returns a new MathTransform object for the inverse transformation direction, that is already initialized with the appropriate parameters. This new object has a transform operation that reverses the transform operation of this (the original) MathTransform object. For example, if the original MathTransform was a Matrix multiply, the inverse transform is multiplication by the inverse of the original Matrix.

4.1.27. Class Name: Parameter

Package Class Belongs to: Coordinate Transform

Documentation:

This class contains numerical values that are used by a coordinate transformation or conversion as control parameters (such as polynomial coefficients or physical measurements). These parameters are generally used in the construction or initialization of the MathTransform object used by one TransformationStep.

Superclass: none

Stereotype: data type

Associations:

user (1) TransformationStep

user (1) MathTransform

error (0..*) CovarianceElement in association: describes

Note: A single Covariance Element object can define the variance in the value of the associated Parameter. A set of Covariance Element objects can define a Covariance Matrix across a set of parameters whose values have correlated errors. Such a Covariance Matrix can be across multiple parameters, but not necessarily all the parameters used by one Transformation Step subclass or object. Such a Covariance Matrix can also be across multiple parameters used by multiple Transformation Step objects of the Image Geometry Transformation subclass. The error in the value of a Parameter object is assumed to be zero when there is no Covariance Element object associated with that Parameter object.

Attributes:

parameterName : CharacterString
The parameterName is a common name given to this parameter used in a coordinate transformation. Examples of names commonly given to parameters used in coordinate conversions include: False Northing, False Easting, Latitude of the Origin, Longitude of the Origin, and Scale Factor. Examples of parameter names for Datum Transformations include: X Axis Translation, Y Axis Rotation, and Scale Differences.

parameterValue : Measure
The numerical value of this parameter used to describe the transformation between two coordinate reference systems, plus a specification of the physical units when applicable.

Operations: none

4.2. Imaging Time Determination Service

4.2.1. Function

The Imaging Time Determination Service determines the imaging time of points in an image.

For a “frame” type of image, all points are imaged at the same time. For pushbroom, whiskbroom, many panoramic, SAR, and other types of images, different points in one image are imaged at somewhat different times. The imaging time differences within one image can be important for some image exploitation purposes, such as estimating the velocity of imaged objects.

This imaging time determination service is included under Image Coordinate Transformation Services because time can be considered to be another dimension of a point coordinate, or the replacement for one axis of a point's image coordinates. Furthermore, the data needed to determine imaging time is closely related to the data needed to convert image positions.

Specification and implementation of this imaging time determination service probably depends on specification of temporal reference systems. Imaging time determination also depends on the temporal part of image geometry models. Complete specification of temporal reference systems has been put off for future work by the OGC. Therefore, complete specification and implementation of this imaging time determination service may have to be delayed.

4.2.2. Service Subtypes

The one currently expected Imaging Time Determination Service is:

1. Determine imaging times for one or more image positions

4.2.3. Result Data

The data produced by the Imaging Time Determination Service includes:

1. Imaging time, for each point
2. Metadata for imaging times, including: (optional)
3. Temporal SRS definition (is metadata for output times)
4. Absolute accuracy estimates for imaging times
(is metadata for output times)
5. Relative accuracy estimates for imaging times
(is metadata for output times)

Result metadata is optionally returned to client software, depending on how the service is called. The ability to produce result metadata when requested is a required capability of the service.

4.2.4. Needed Data

The data needed for use by the Imaging Time Determination Service includes:

1. Image position coordinates, for each point
2. Output temporal SRS definition (is metadata for output times)
3. Temporal SRS transformation parameters (optional)
(is metadata for temporal SRS or transformation)
4. Temporal transformation accuracy estimates (when output accuracy is needed) (is metadata for temporal transformation)
5. Metadata for input position coordinates, including:
6. Input SRS definition (is metadata for input positions)
7. Absolute accuracy estimates for input position coordinates (when output absolute accuracy is needed) (is metadata for input positions)
8. Relative accuracy estimates for input position coordinates (when output relative accuracy is needed) (is metadata for input positions)

4.2.5. Discussion

The Temporal SRS transformation parameters (item 3 in Section 4.2.4) must include or imply all the information needed to determine the imaging time for any point. For images collected over some time, the Image coordinate transformation parameters (item 3 in Section 4.1.4) must also imply or include all the information needed to determine the imaging time for any point. This information may include the position interpolation algorithm to be used between discrete positions recorded along the sensor flight path, to determine intermediate positions or times. (It is possible to record only discrete sensor positions, not a continuous record of the positions. In that case, an interpolation algorithm is needed to interpolate between the stored positions.)

4.2.6. Object Model

TBD

Figure 3. Object Model of TBD

4.2.7. Class Name: TBD

Package Class Belongs to:

Documentation:

Superclass: none

Stereotype: none

Associations:

Public Interface:
Attributes:

Operations:

Operation name:
Return type:
Arguments:
Documentation:

4.3. Image Geometry Model Conversion Services

4.3.1. Function

The Image Geometry Model Conversion Services produce a different geometry model for an image, or metadata for a different image geometry model. An alternate name for these Image Geometry Model Conversion Services would be "Image Geometry Model Fitting Services".

There are multiple possible geometry models for an image, with different properties. For example, a rigorous geometry model can be accurately adjusted, but has high computation requirements. On the other hand, an "approximate" geometry model has lesser computation requirements, but cannot be accurately adjusted. Conversion from a rigorous geometry model to a corresponding approximate geometry model for an image is then sometimes required. (Conversion from an "approximate" geometry model to a rigorous geometry model is usually not possible or practical.)

These Image Geometry Model Conversion Services are described here because they support the Image Coordinate Transformation Services. However, these Image Geometry Model Conversion Services have significantly different interfaces.

4.3.2. Service Subtypes

The expected sub-types of Image Geometry Model Conversion Services include, but are not limited to:

1. Fit approximate image geometry model to point positions computed using existing image geometry model

2. Convert image geometry model to different, mathematically equivalent model, by converting geometry parameters of existing image geometry model

4.3.3. Result Data

The data produced by these Image Geometry Model Conversion Services includes:

1. New image geometry model, for entire image or selected section of image (is modified metadata for image)
2. Metadata for new image geometry model, including:
3. Absolute accuracy estimates for new model (is metadata for new model)
4. Relative accuracy estimates for new model (is metadata for new model)
5. Modified image SRS definition (is metadata for new model)
6. Identification of applicable image section (is metadata for new model)
7. Model fitting error estimates (is metadata for new model)

These services should always return result metadata to client software (result metadata is not optional).

4.3.4. Needed Data

The data needed for use by these Image Geometry Model Conversion Services includes:

1. Existing image geometry model (is metadata for image)
2. Desired accuracy of geometry model conversion
3. Values of parameters required and useful to control conversion processes (sometimes called strategy parameters) (is metadata for conversion process)
4. Selection of desired image section (is metadata for new model)
5. Metadata for existing image geometry model, including:
6. Absolute accuracy estimates for model (is metadata for model)
7. Relative accuracy estimates for model (is metadata for model)
8. Existing image SRS definition (is metadata for new model)

4.3.5. Object Model

Figure 4-3 is a (draft) object model for the Image Geometry Model Conversion Services, in the form of a UML class diagram. The new classes needed to perform Image Geometry Model Conversion Services are in the Image Geometry Model Conversion package. All classes shown are expected to have multiple subclasses, not shown in this overview diagram. Also, some classes may have additional attributes and operations, not shown here.

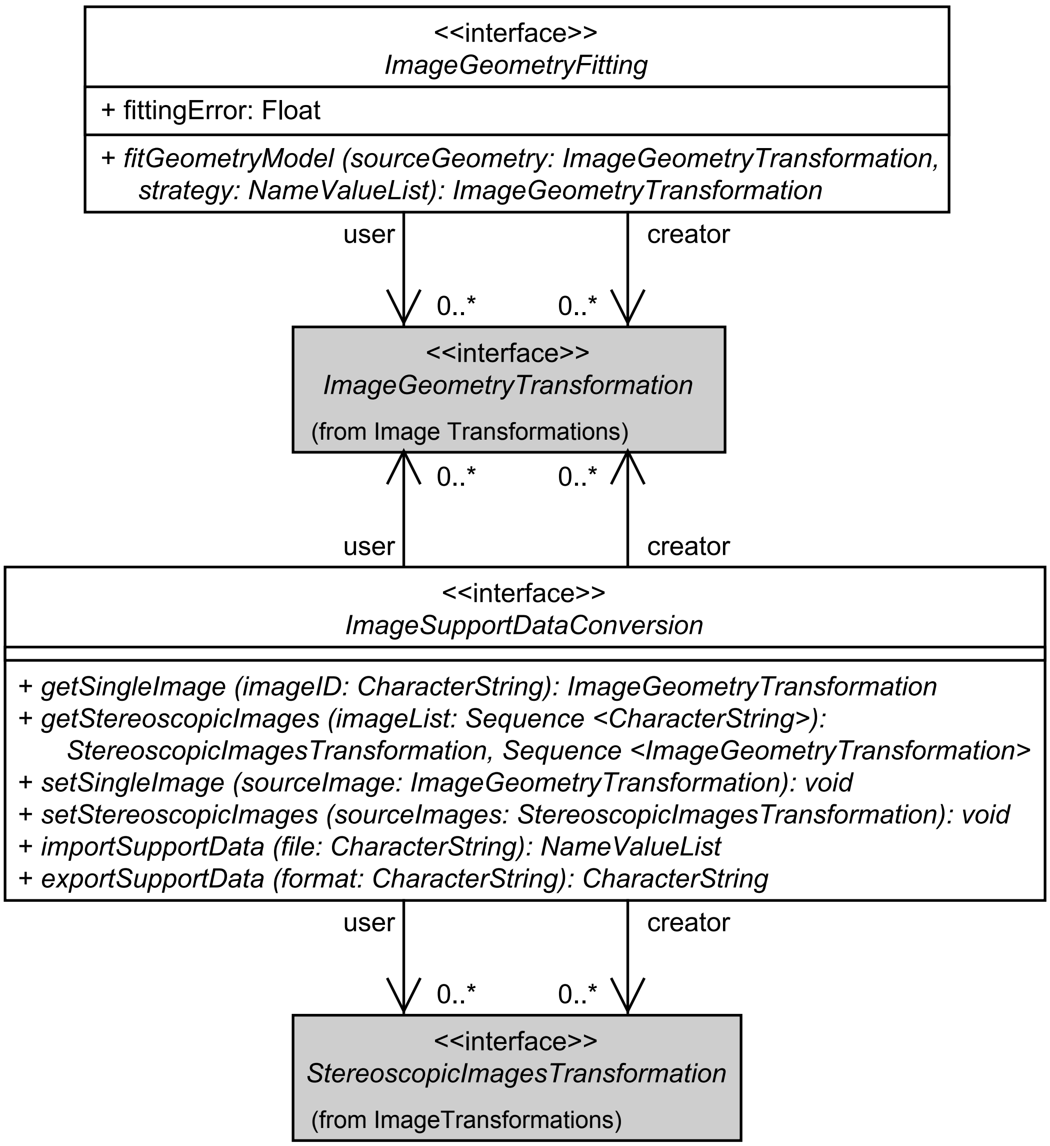

Figure 4-3. Object Model of Image Geometry Model Conversion Package

The following sections provide more information about each new class shown in Figure 4-3.

4.3.6. Class Name: ImageGeometryFitting

Package Class Belongs to: Image Geometry Model Conversion

Documentation:

This service class produces a different type of image geometry model by fitting a specific model type to any existing image geometry model. The fitted geometry type is usually approximate (not rigorous).

Subclasses:

Subclasses are expected to be defined for fitting different image geometry models, such as the various approximate models now described in Topic Volume 7: The Earth Imagery Case [1]. The subclasses support identical interfaces except for any different geometry parameters that may be needed for the fitted image geometry models.

Superclass: none

Stereotype: interface

Associations:	user (of 0..*)	ImageGeometryTransformation
	creator (of 0..*)	ImageGeometryTransformation

Attributes:

fittingError: Float
Desired maximum fitting error, in pixel spacings.

Operations:

fitGeometryModel (sourceGeometry: ImageGeometryTransformation, strategy: NameValueList): ImageGeometryTransformation

Creates a new image geometry model that is fitted to accurately approximate an existing image geometry model, within the desired maximum fitting error. Can use strategy parameters that control the fitting process which are specific to each subclass.

4.3.7. Class Name: ImageSupportDataConversion

Package Class Belongs to: Image Geometry Model Conversion

Documentation:

This service class converts image geometry support data from a specific image geometry model and/or data format to a different image geometry model and/or data format. This different image geometry model and/or data format is (normally) the one used by a specified subtype of the Image Geometry Transformation class, and perhaps also the corresponding subtype of the Stereoscopic Images Transformation class.

Subclasses:

Subclasses are expected to be defined for each image support data format and for each image geometry model supported by that format. The subclasses support identical interfaces except for any different geometry parameters that may be needed for the each specific image geometry model and/or data format.

Superclass: none

Stereotype: interface

Associations:	creator (of 0..*)	ImageGeometryTransformation
	creator (of 0..*)	StereoscopicImagesTransformation
	user (of 0..*)	ImageGeometryTransformation
	user (of 0..*)	StereoscopicImagesTransformation

Attributes: none

Operations:

getSingleImage (imageID: CharacterString): ImageGeometryTransformation
Retrieve the image geometry support data for the specified image from the image support data file(s), and create an Image Geometry Transformation object.

getStereoscopicImages (imageList: Sequence <CharacterString>): StereoscopicImagesTransformation, Sequence <ImageGeometryTransformation>
Retrieve the image geometry support data for the specified list of images from the image geometry support data file(s), and create one Stereoscopic Images Transformation object plus two or more ImageGeometryTransformation objects.

setSingleImage (sourceImage: ImageGeometryTransformation): void
Convert image geometry support data from the specified Image Geometry Transformation object to external format, and store converted data in an image geometry support data file.

setStereoscopicImages (sourceImages: StereoscopicImagesTransformation): void
Convert image geometry support data from the specified Stereoscopic Images Transformation object to external format, and store converted data in an image geometry support data file.

importSupportData (file: CharacterString): NameValueList
Import the specified image geometry support data file. Also, check this image support data and return a Name Value List describing any "errors" or missing data detected.

exportSupportData (format: CharacterString): CharacterString
Obtain current image geometry support data as a character string in the specified external format.

4.4. Accuracy Conversion Services

4.4.1. Function

The Accuracy Conversion Services convert position accuracy estimates between error covariance matrix form and Circular Error (CE) plus Linear Error (LE) or other forms. These accuracy conversions are also applicable to linear dimensions and perhaps other accuracy estimates.

When converting most other forms of accuracy data to covariance matrices, correct values will not be known for some covariance matrix cells. Specifically, the off-diagonal cells for covariances between coordinates will not be known. As specified in Abstract Specification Topic 9: Quality, the values of these covariance matrix cells should be null or missing. (Alternately, these covariances values would be assumed to be zero.)

When converting covariance matrices to other forms of accuracy data, some cells of a covariance matrix may be unknown. When this occurs, the unknown matrix cells must be handled in an appropriate manner.

4.4.2. Service Subtypes

The expected sub-types of Accuracy Conversion Services include, but are not limited to:

1. Convert covariance matrices to other forms:
2. Convert 3-D covariances to CE plus LE
3. Convert 2-D covariances to CE
4. Convert 1-D variance to LE
5. Convert 3-D covariances to Spherical Error (SE)
6. Convert 1-D variance to Standard Deviation
7. Convert other forms to covariance matrices:
8. Convert CE plus LE to 3-D covariances
9. Convert CE to 2-D covariances
10. Convert LE to 1-D variance
11. Convert Spherical Error (SE) to 3-D covariances
12. Convert Standard Deviation to 1-D variance

4.4.3. Result Data

The data produced by the Accuracy Conversion Services includes:

1. Accuracy estimates in desired form (is metadata for some point positions)
2. Confidence level(s) for output CE, LE, and SE values
   (is metadata for CE, LE, and SE values?)

4.4.4. Needed Data

The data needed for use by the Accuracy Conversion Services includes:

1. Accuracy estimates in existing form (is metadata for same point positions)

2. Selection of accuracy estimate form desired

3. Confidence level(s) for input CE, LE, and SE values
(is metadata for CE, LE, and SE values?)

4. Accuracy conversion parameters (is metadata for accuracy conversions?)

4.4.5. Object Model

Figure 4-4 is an object model for the Accuracy Conversion package, in the form of an UML class diagram. The new classes needed to perform Accuracy Conversion services are in the Accuracy Conversion package. Note that the Accuracy Conversion package depends on the Accuracy and the Unit Of Measure packages, both defined in Topic Volume 0.

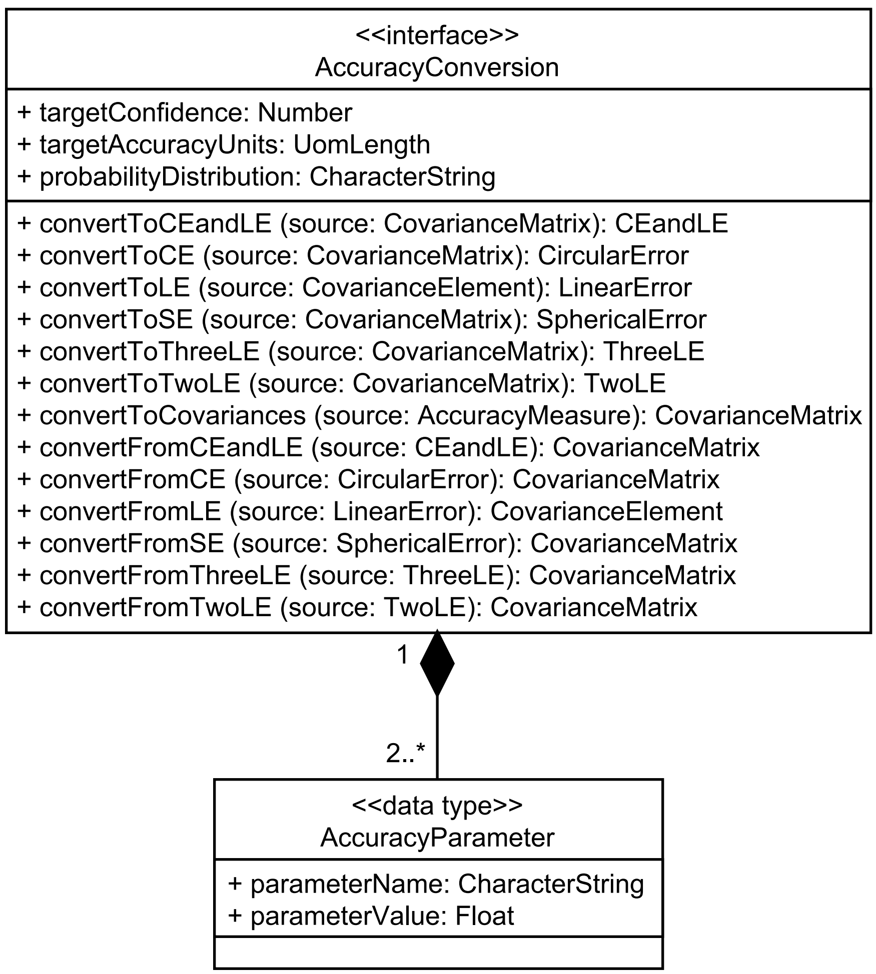

Figure 4-4. Object Model of Accuracy Conversion Package

Reference: A reference discussing the use of covariance matrices is: "Probability, Random Variables, and Stochastic Processes" by Athanasios Papoulis, 3rd Edition 1991, McGraw Hill

The following subsections more completely define the Accuracy Conversion package classes and associations.

4.4.6. Class Name: AccuracyConversion

Package Class Belongs to: Accuracy Conversion

Documentation:

This service class converts error estimates (or accuracy values) between error covariance matrix form and Circular Error (CE), Linear Error (LE), and other forms. These accuracy conversions are applicable to ground and image position coordinates and certain other accuracy estimates.

Superclass: none

Stereotype: interface

Associations: user (of 2..*) AccuracyParameter

Attributes:

targetConfidence: Number
The percentage of actual error values expected to be less that the specified error estimate, in percent units. The value of this attribute applies to all Circular Error (CE), Linear Error (LE), and Spherical Error (SE) values output by operations of this class. This percentage is often specified by an integer with legal values ranging from 0 to 100. Commonly used values are 50 (percent), 67 (percent), 90 (percent), and 95 (percent). For a Linear Error with a normal probability distribution, a confidence of 67 (percent) corresponds to the standard deviation.

targetAccuracyUnits: UomLength
The units to be used by the Accuracy Measure produced by converting from another Accuracy Measure. These units are used by all Circular Error, Linear Error, and Spherical Error estimate values output by operations of this class. The square of these units is used by each CovarianceElement value output by operations of this class.

probabilityDistribution: CharacterString
Name of the probability distribution assumed by the implementation of this class. The default and typical values are "normal distribution".

Operations:

convertToCEandLE (source: CovarianceMatrix): CEandLE
Convert the specified 3-D ground position covariance matrix to the corresponding horizontal Circular Error (CE) and vertical Linear Error (LE) accuracies. The resulting CE and LE values are both in the length units specified by the targetAccuracyUnits attribute.

convertToCE (source: CovarianceMatrix): CircularError
Convert the specified 2-D position covariance matrix to the corresponding Circular Error (CE) accuracy estimate. The resulting CE value is in the length units specified by the targetAccuracyUnits attribute. This operation can convert a 2-D ground position covariance matrix to the corresponding horizontal CE, or it can convert a 2-D image position covariance to the corresponding image CE.

convertToLE (source: CovarianceElement): LinearError
Convert the variance in the specified CovarianceElement to the corresponding Linear Error (LE) accuracy estimate. If the confidence is 67% and the errors have a normal probability distribution, this LE is the same as the Standard Deviation. The resulting LE value is in the length units specified by the targetAccuracyUnits attribute. This operation can convert a 1-D ground elevation variance to the corresponding vertical LE, or it can convert a one image axis position to the corresponding LE.

convertToSE (source: CovarianceMatrix): SphericalError
Convert the specified 3-D ground position covariance matrix to the corresponding Spherical Error (SE) accuracy estimate. The resulting SE value is in the length units specified by the targetAccuracyUnits attribute.

convertToThreeLE (source: CovarianceMatrix): ThreeLE
Convert the specified 3-D ground position covariance matrix to the corresponding three Linear Error (SE) accuracy estimates, one for each of the three coordinates. The resulting LE values are all in the length units specified by the targetAccuracyUnits attribute.

convertToTwoLE (source: CovarianceMatrix): TwoLE
Convert the specified 2-D ground position covariance matrix to the corresponding two Linear

Error (SE) accuracy estimate. The resulting LE values are all in the length units specified by the targetAccuracyUnits attribute.

convertToCovariances (source: AccuracyMeasure): CovarianceMatrix
Convert the specified Accuracy Measure to the corresponding Covariance Matrix accuracy estimate. The resulting Covariance Element values are all in the square of the length units specified by the targetAccuracyUnits attribute. In most cases, some of the covariance elements in the output matrix will be unknown, and those Covariance Element objects will be missing.

convertFromCEandLE (source: CEandLE): CovarianceMatrix
Convert the specified horizontal Circular Error (CE) and vertical Linear Error (LE) to the corresponding 3-D ground position covariance matrix. The resulting Covariance Element values are all in the square of the length units specified by the targetAccuracyUnits attribute. The off-diagonal covariance elements in the output matrix will be unknown, and those Covariance Element objects will thus be missing.

convertFromCE (source: CircularError): CovarianceMatrix
Convert the specified Circular Error (CE) accuracy estimate to the corresponding 2-D position covariance matrix. The resulting Covariance Element values are all in the square of the length units specified by the targetAccuracyUnits attribute. This operation can convert horizontal CE to the corresponding 2-D ground position covariance, or it can convert image position CE to the corresponding 2-D covariance. The off-diagonal covariance elements in the output matrix will be unknown, and those Covariance Element objects will thus be missing.

convertFromLE (source: LinearError): CovarianceElement
Convert the specified Linear Error (LE) accuracy estimate to the corresponding variance Covariance Element. The resulting Covariance Element value is in the square of the length units specified by the targetAccuracyUnits attribute. This operation can convert vertical LE accuracy to the corresponding elevation variance, or it can convert one position axis LE accuracy to the corresponding variance

convertFromSE (source: SphericalError): CovarianceMatrix
Convert the specified Spherical Error (SE) accuracy estimate to the corresponding 3-D ground position covariance matrix. The resulting Covariance Element values are all in the square of the length units specified by the targetAccuracyUnits attribute. The off-diagonal covariance elements in the output matrix will be unknown, and those Covariance Element objects will thus be missing.

convertFromThreeLE (source: ThreeLE): CovarianceMatrix
Convert the specified three Linear Error (LE) accuracy estimates to the corresponding 3-D ground position covariance matrix. The resulting Covariance Element values are all in the square of the length units specified by the targetAccuracyUnits attribute. The off-diagonal covariance elements in the output matrix will be unknown, and those Covariance Element objects will thus be missing.

convertFromTwoLE (source: TwoLE): CovarianceMatrix
Convert the specified two Linear Error (LE) accuracy estimates to the corresponding 2-D ground position covariance matrix. The resulting Covariance Element values are all in the square of the length units specified by the targetAccuracyUnits attribute. The off-diagonal covariance elements in the output matrix will be unknown, and those Covariance Element objects will thus be missing.

4.4.7. Class Name: AccuracyParameter

Package Class Belongs to: Accuracy Conversion

Documentation:

This data class contains values of the error estimate conversion parameters used by the Accuracy Conversions class. For example, these parameters can include the multipliers needed to convert Linear Error values between different confidence probabilities, and the different multipliers for converting Circular Error values.

Superclass: none

Stereotype: data type

Associations: (used by 1) AccuracyConversion

Attributes:

parameterName: CharacterString
The name of this conversion parameter.

parameterValue: Float
The numerical value of this conversion parameter.

Operations: none

4.5. Package Dependencies

The new packages defined in this document build on multiple packages defined in other Topic volumes, including the Unit Of Measure, Accuracy, Geometry, Positioning, and Coordinate Transform packages. Figure 4-5 is a package diagram showing the dependencies of the new packages on existing packages. The previously defined packages are shaded in this diagram. This diagram shows that many other packages depend on the Accuracy and Unit Of Measure packages, by showing an unnamed package that contains multiple other packages which depend on the Accuracy and Unit Of Measure packages.

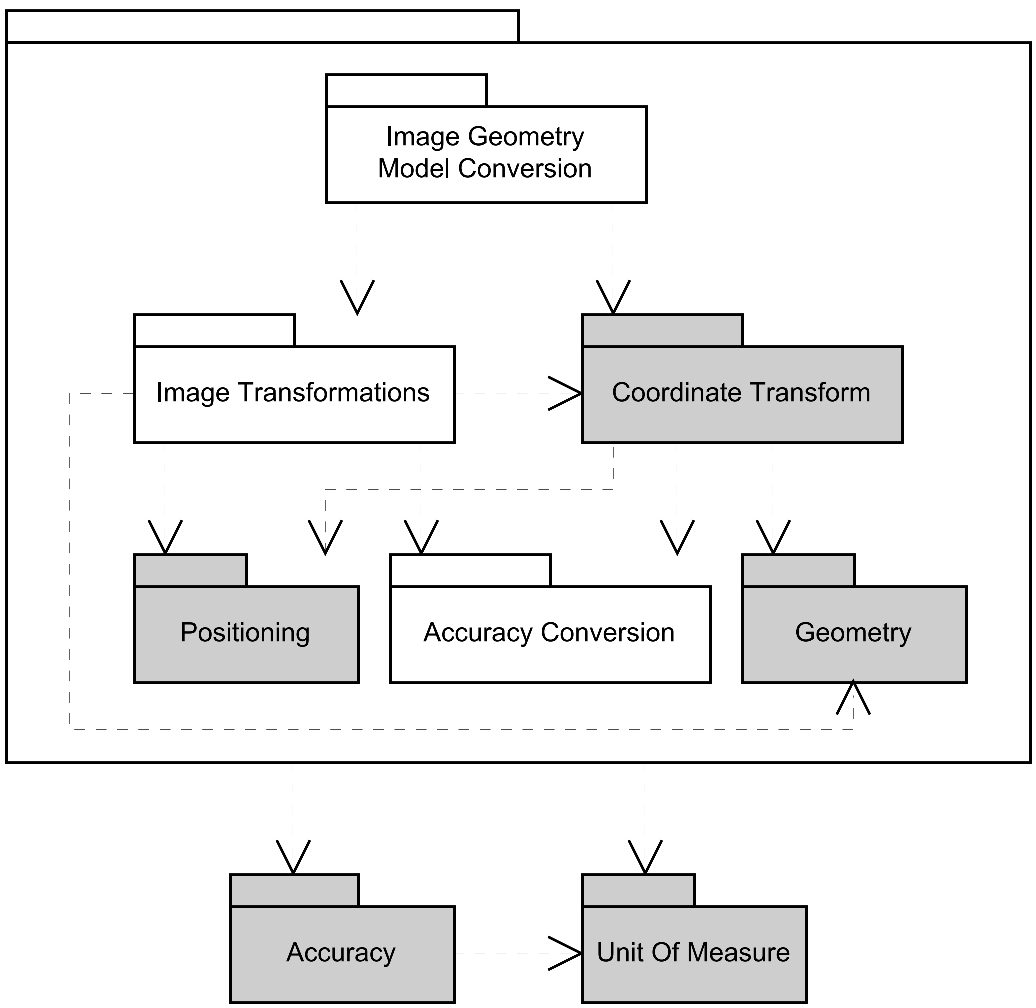

Figure 4-5. New Packages Dependencies on Other Packages

5. Well Known Structures

The object models in Section 4 refer to a set of lower level data type classes that are not defined in Section 4. Table 5-1 lists these data type classes and provides a brief summary of the information contained in each. Most of these data types are defined in shared UML packages, that are used by the abstract models in multiple Abstract Specification Topic Volumes. These shared UML packages are specified in Section 4 of Topic Volume 0. Table 5-1 also lists the shared UML package in which each data type class is defined. Many of these data type classes also correspond to the Well Known Types discussed in the remainder of Section 5.

Data Type Class	Information Contained	Package
CharacterString	A sequence of alphanumeric characters, usually human readable, often including spaces and punctuation characters	Basic Data Types
Number	A numerical value, either an integer, a floating point number, or a decimal number	Basic Data Types
Float	A single precision floating point number, a subtype of "Real"	Basic Data Types
Integer	An integer number	Basic Data Types
Matrix	An array or matrix of floating point values	TBD
UomLength	Specifies unit of measure of a length or distance	Unit of Measure
Measure	Data structure containing a number giving the value of a quantity, with information defining corresponding unit of measure used by the numerical value	Unit of Measure
CoordinatePoint	Data structure containing two or three numbers, each number representing the position of a point in one position axis	Positioning
DirectPosition	Data structure containing one CoordinatePoint with the corresponding AccuracyMeasure and an association to a SpatialReferenceByCoordinates	Positioning
GM_Object	Data structure containing a geometry	Geometry
CovarianceElement	One element of a covariance matrix, representing the expected value of the product of the simultaneous errors in two quantities. If the two quantities are the same, represents the variance of the error in that quantity.	Accuracy
CovarianceMatrix	Data structure containing a complete or partial covariance matrix, usually representing the expected errors in 2-D or 3-D position coordinates. Can be used for absolute and relative position error estimates.	Accuracy
AccuracyMeasure	Any of multiple measures of accuracy	Accuracy
CircularError	A floating point number containing a Measure for Circular Error (CE), with the corresponding confidence probability	Accuracy
LinearError	A floating point number containing a Measure for Linear Error (LE), with the corresponding confidence probability	Accuracy
SphericalError	A floating point number containing a Measure for Spherical Error (SE), with the corresponding confidence probability	Accuracy
CEandLE	Two floating point numbers, containing Measures for horizontal Circular Error (CE) and vertical Linear Error (LE)	Accuracy
ThreeLE	Three floating point numbers containing Measures for Linear Error (LE) in three position coordinates	Accuracy
TwoLE	Two floating point numbers containing Measures for Linear Error (LE) in two position coordinates	Accuracy
NameValueList	A sequence of name and value pairs, where each name is a Character String and each value is of the type appropriate to that name.	Programming Language (see second note below)

Table 5-1. Data Types or Classes Used in Object Models

Note: The Basic Data Types package defined in Section 4.2 of Topic 0 should be expanded to include "Float" and "Matrix" data types. The shared UML packages specified in Section 4 of Topic 0 should be expanded to include the "Positioning" package now specified in Topic 2.

Note: The NameValueList data type, used in Section 4.1.3 of this Topic 16, is a specialized subtype of a “Dictionary” as now defined in Section 4.1.4 of Topic 0. For the NameValueList data type, the Dictionary “KeyType” is always “string” and the “ValueType” is “any”. This discrepancy currently exists because that part was added to Topic 0 after Section 4.1.3 of Topic 16 was last updated. This discrepancy should be resolved when Section 4.1.3 of Topic 16 is next updated.

This section also discusses the contents and format of most Image Coordinate Transformation service “needed data” and “result data”. The “needed data” can alternately be called inputs, and the “result data” can alternately be called outputs. The subsections discuss various data categories, recognizing that the “needed data” and “result data” of multiple services are often identical or similar. Some of these data descriptions use the ISO (and OMG) standard Interface Definition Language (IDL).

5.1. Position Coordinates

Ground position coordinates are inputs and/or outputs of several services. Each ground position normally requires specifying the values of three (or two) coordinates. Similarly, **i**mage position coordinates are inputs and/or outputs of several services. Each image position normally requires specifying the values of two coordinates.

Well known structures for position coordinates are defined in the Positioning UML package that is specified in Section 3.1 of Abstract Specification Topic 2. This Positioning package is used by multiple Topics, including this one. This Positioning package defines CoordinatePoint and DirectPosition data type classes, both of which support specifying the values of two image coordinates or three ground coordinates.

The DirectPosition data type includes position accuracy information, while the CoordinatePoint data type does not. The accuracy attribute in the DirectPosition class allows specifying error estimates for the included coordinates. However, only the interfaces defined in Sections 4.1.10 and 4.1.23 of this Topic volume use that accuracy attribute. The interfaces defined in Sections 4.1.8 and 4.1.9 do <u>not</u> use that accuracy attribute.

Each object of the DirectPosition data type class currently includes an association to one SpatialReferenceByCoordinates object, that defines the Spatial Reference System (SRS) of those coordinates. However, only the interfaces defined in Sections 4.1.8 through 4.1.10 utilize that SRS association. The interfaces defined in Section 4.1.23 do <u>not</u> use that SRS association.

5.2. Ground SRS Definition

Well known structures for defining the SRS of ground coordinates are specified in Abstract Specification Topic 2.

5.3. Image SRS Definition

The SRS of image coordinates is often specified by a ground position SRS definition plus an image geometry model, that together relate image positions to ground positions (in that ground SRS). The ground position SRS can be specified as discussed in Section 5.3. The image geometry model can be specified by the values of the set of parameters used by a specified mathematical model of the image geometry. These parameters are considered metadata for the image.

Image geometry model metadata is already partially discussed or implied in Abstract Specification Topic 7: The Earth Imagery Case, and in OGC document 98-033: Alternatives for Transferring Orientation Data of Digital Aerial Images. These documents describe a number of possible forms of image geometry model metadata:

1. Values of image geometry model parameters:

   1.1. For rigorous geometry models
   (there are many existing rigorous geometry models)

   1.2. For real-time geometry models, including:

      1.2.1. Polynomial models (Section 3.2 of Topic 7)

      1.2.2. Ratios of Polynomials (Section 3.4 of Topic 7)

1.2.3. Universal Real-Time Model (Section 3.5 of Topic 7)

2. Positions of points in both ground and image coordinates:

2.1. Grid of Points with Interpolation (Section 3.3 of Topic 7)

2.2. Set of reference points, used by client for fitting parameters of image geometry model (Sections 2.3 and 3.2 of 98-033)

The OGC Technical Committee (TC) must answer several questions on the possible forms of image geometry model metadata, including:

1. Are there other possible forms of image geometry model metadata?
2. To what degree can or should the TC leave the selection of one or more forms of image geometry model metadata up to organizations that propose implementation specifications in response to an RFP?
3. For which future RFP should the form(s) of image geometry model metadata be selected (whether selected by the TC or by the RFP responders)?
4. Which one or more forms of image geometry model metadata should the TC select or prefer?

5.4. Position Error Estimates

Absolute and relative position error (or accuracy) estimates are also inputs and outputs of several services. The accuracy inputs and outputs are often optional, needed by operations only when accuracy output data is desired by a client program or user.

Well known structures for position accuracy are defined in the Accuracy Measure UML package that is specified in Section 4.4 of Abstract Specification Topic 0. This Accuracy Measure package is used by multiple Topics, including this one. This Accuracy Measure package defines several alternative forms of position error estimates that could be used, including covariance matrices.

As discussed in Section 2.8 of Topic 9, detailed accuracy information can be recorded using a covariance matrix, sometimes called a variance-covariance matrix. For the three ground coordinates of one point, a covariance matrix is a 3 by 3 matrix, with the matrix rows and columns each corresponding to the three coordinates. For just the two horizontal ground 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.

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 ground coordinates, or the squares of the standard deviations. The off-diagonal cells contain the covariances between the errors in the corresponding ground 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.

Covariance matrices can be used to record absolute and/or relative accuracies. A covariance matrix for the relative accuracy between two points uses the three (or two) coordinates of one point for matrix rows and the three (or two) coordinates of the second point for matrix columns. A complete covariance matrix for N specific points would contain 3N rows and 3N columns.

When other forms of accuracy data are desired by a user, such as Circular Error (CE), Linear Error (LE), and Spherical Error (SE), they can be converted from (or to) covariance matrices. (See Sections 2.4 through 2.7 of Topic 9 for definitions of CE, LE, and other forms of accuracy data.) Such accuracy conversion is the purpose of the Accuracy Conversion Services, as discussed in Section 4.4 of this document. CE, LE, and SE can each be transferred as a single precision floating point number, as can the confidence probability associated with each CE, LE, and SE value.

5.5. Elevation Data

When a position in a monoscopic image is used to find the corresponding ground position, elevation (or height or surface shape) data is often used. Elevation data is thus an input to certain services. Such elevation data could take one of several forms, including:

1. Single elevation value, to be used for one or more image positions
2. List of elevation values, to be used with a corresponding list of image positions
3. Elevation coverage, that defines the elevation as a function of ground position, to be used for one or more image positions

Each elevation could be transferred as one double precision floating point value. (Note: The SRSs used for elevations are listed as separate service inputs and outputs.)

The abstract model of Image Coordinate Transformation Services specified in Section 4.1 includes interfaces to surface shape or elevation data. These interfaces produce a 3-D ground position that is the intersection of an image ray with the specified surface shape model. The ElevationModel class specified in Section 4.1.18 is expected to use an OGC Coverage to contain elevation data, and to provide access to that data.

5.6. Elevation Error Estimates

When elevation is used and output accuracy is needed, elevation accuracy data is a needed input to certain services. The accuracy of a single elevation value, or of all elevations in a list, could be specified by a one single precision floating point number. This value could have one of several meanings, including variance, standard deviation, or LE (Linear Error). A LE value could use one of several confidence probabilities. However, for consistency with using a covariance matrix to specify the accuracy of two or three dimensional coordinates, a covariance matrix or variance value should be used for elevation value accuracy.

Since elevation data is used to determine a 3-D ground position that is the intersection of an image ray with the specified surface shape model, the accuracy of this position is needed when output accuracy data is required. Well known structures for position accuracy are defined in the Accuracy Measure UML package that is specified in Section 4.4 of Abstract Specification Topic 0. This Accuracy Measure package defines well known structures for variances and for covariance matrices.

When elevation is specified by an elevation coverage, the effect of horizontal position errors on the elevation value error should be represented. This can be done using a partial 3-D covariance matrix, with a special interpretation of the values in off-diagonal cells. These off-diagonal cells can contain the ratio of the covariance value of that cell to the unknown variance of the corresponding horizontal axis. The variance cell for the elevation would have the normal meaning. The other diagonal cells in the covariance matrix, for the two horizontal coordinates, would have unknown values. The off-diagonal cells for the covariances between the horizontal coordinates would also be have unknown values.

Using this approach, elevation accuracy data types can be defined. Such a set defined using ISO standard IDL data types and structures is:

```
// Type: Matrix Cell, of a covariance matrix
      struct MatrixCell {
         string <2>   axes;   // Axes of covariance matrix
         float        value;  // Units: Meters squared
         };

// Type: Elevation Covariances
      typedef sequence <MatrixCell, 3>  ElevationCovariances;
   // Covariance matrix cells included only when correct
   //    value is known
   // Required values of "axes" string: ZZ
   // Optional values of "axes" string: XZ, YZ
   // Where X, Y, and Z stand for three ground coordinates
```

```
   // Matrix cell XZ contains the ratio of XZ to XX
   // Matrix cell YZ contains the ratio of YZ to YY
```

5.7. Desired Image Section

The desired image section is also an input or output of some services. The location and size of the desired image section must be specified, within a larger image (or set of image pixels). The OGC accepted Simple (or grid) Coverages Implementation Specification specifies a standard format for specifying the desired image section, as inputs to Grid Coverage access operations. This document assumes that this image section specification format could be used for image exploitation services.

Alternately, the desired image section could be specified using an area geometry, such as specified in the three Simple Features implementation specifications accepted by the OGC. We assume this area feature geometry could be in 2-D image coordinates, not 3-D or 2-D ground coordinates. Alternately, this area feature geometry could be specified in 3-D ground coordinates, to be converted into the corresponding 2-D image coordinates if needed.

If the desired image section is a rectangle in image space, such a rectangle could be specified by the pixel position of one corner, plus the pixel section width and height. Using this approach, an Image Section data type defined using ISO standard IDL data types and structures is:

```
// Type: Image Section, rectangular section of an image
      struct ImageSection {
         long corner_column; // Smallest pixel column number
         long corner_row;    // Smallest pixel row number
         long width;         // Number of image pixel columns
         long height;        // Number of image pixel rows
         };
```

5.8. Strategy Parameters

Some services require inputs containing values of strategy parameters, that are used by the service algorithms to control service operations. The values of such strategy parameters are often heuristic, being experimentally found to produce the best results for some set of primary input data. However, the most effective set of values differs for different categories of other input data.

The needed set of strategy parameters is different for different services and is very likely to be different for different implementations of the same service. However, a name value list could be used as a standard data structure for all possible sets of strategy parameters. A name value list is similar to a “Dictionary” as specified in Section 4.1.1 of Topic 0, where the KeyType is “string” and the ValueType is “any”.

Of course, each implementation of each service must specify the set of names and definitions that it uses for strategy parameters, together with the data type, units, and range (or domain) of the values for each parameter name. Each service probably should make all this information available to a client by providing an operation that retrieves this strategy parameter description information.

5.9. Selection of Service Operation

Selection of the specific service function desired is listed as an input for some services. Such selection would conventionally be done by calling a different service object or operation for each specific service. Alternately, selection can be done by using one or more “Service Selection” inputs whose data type is an enumeration of all the alternative specific services available through one service operation. These “Service Selection” inputs could be handled like, or as an extension of, the Strategy Parameters discussed above.

5.10. Accuracy Parameters

The “Accuracy parameters” specified in Section 4.4.7 are needed for conversion between covariance matrices and CE plus LE and other forms. Table 5-3 defines one set of Accuracy Parameters that might be used, for each supported error probability.

Name	Data Type	Units	Definition	Comments
Error Probability	float	(none)	Confidence probability for CE and LE error estimates	Note 1
Probability Distribution Name	string	(none)	Name of probability distribution, such as "normal distribution"	
LE Multiplier	float	(none)	Multiplier of standard deviation to obtain LE with specified probability	Note 2
CE Multiplier Function	sequence <float>	(none)	Multipliers of standard deviation to obtain CE with specified probability	Note 3

Table 5-3. Possible Accuracy Conversion Parameters

Table Notes:

1. The possible probability values include 0.50, 0.6827, 0.90, and 0.95.
2. This multiplier is used to compute LE with the specified confidence probability from the standard deviation of the elevation error (square root of the elevation variance).
3. These multipliers are used to compute CE with the specified confidence probability from the standard deviation (square root of the variance) along the major axis of the error ellipse.

   The CE is computed from the covariance matrix of the expected errors in two axes, either the two horizontal ground coordinates or the two image coordinates. When the error estimates in the two axes are correlated and/or are not equal, the variances are first computed for the major and minor axes of the error ellipse. The ratio of the variance in the minor axis to the variance in the major axis of the error ellipse is then computed. As this ratio varies from 0.0 to 1.0, the multiplier needed to compute CE varies from one number to a larger number.

   The multipliers are specified for a variable number of evenly spaced values of the minor/major axis variance ratio, from 0.0 to 1.0. (The number of ratio values may be 21, for ratio values differing by 0.05.) For a minor/major ratio between the recorded values, linear interpolation is used.

6. Future Work

Further refinement of the object models in this document appears needed. In particular, further work is needed on the abstract model for the Imaging Time Determination Service. Specification and implementation of the imaging time determination service probably depends on specification of temporal reference systems. Complete specification of temporal reference systems has been put off for future work by the OGC. Therefore, complete specification and implementation of this imaging time determination service will be delayed.

As stated in Section 2.10.3, the OGC probably should cooperate with the International Society of Photogrammetry and Remote Sensing (ISPRS) in defining standard image geometry models and submodels. The next ISPRS meeting is scheduled July 19-25, 2000, in Amsterdam, The Netherlands. The IES SIG and probably the CT WG should participate in related ISPRS sessions.

Further refinement of the object models in this document is also needed to represent and access the metadata needed to describe specific data objects and services. Work on service metadata has started in the Metadata SIG and the Architecture SIG. There are several possible approaches to abstract modeling of metadata using UML, including:

1. Metadata Operations – One approach would be to provide metadata retrieval operations in each class that needs metadata. This is the approach previously suggested in Section 4.1 of this Topic Volume.
2. Metadata Classes – Another approach would be to provide metadata classes that are associated with each data and service class which needs metadata. The CT ad hoc working group has tentatively decided to pursue this approach.
3. Metadata Attributes – Another approach would be to provide metadata attributes in each data and service class that needs metadata.

These three approaches are not mutually exclusive. Different approaches could be used in different places. Alternately, two approaches could be used together. For example, metadata retrieval operations could be used to create actual metadata objects that are associated with each data and service object that needs metadata.

7. Appendix A. Acronyms and Glossary

7.1. Acronyms

1-D	One-Dimensional
2-D	Two-Dimensional
3-D	Three-Dimensional
CE	Circular Error
CT	Coordinate Transformation
IDL	Interface Definition Language
IES	Image Exploitation Services
IFSAR	Interferometric Synthetic Aperture Radar (Interferometric SAR)
IGM	Image Geometry Model
ISO	International Standards Organization
ISPRS	International Society of Photogrammetry and Remote Sensing
LE	Linear Error
OGC	Open GIS Consortium
RFP	Request for Proposals
RRDS	Reduced Resolution Data Set
SAR	Synthetic Aperture Radar
SE	Spherical Error
SIG	Special Interest Group
SRS	Spatial Reference System
TBD	To Be Determined
TBR	To Be Reviewed
TC	Technical Committee
UML	Universal Modeling Language
USIGS	United States Imagery and Geospatial System
WG	Working Group

7.2. Definitions

Editor's Note: Most of these definitions are copied from the USIGS Glossary. That glossary includes definitions extracted from many other documents, including ISO and OGC documents. The definitions not copied from the USIGS Glossary are annotated "(This definition is not in the USIGS Glossary.)".

Absolute accuracy

1. Absolute accuracy is defined as the statistic which gives the uncertainty of a point with respect to the datum required by a product specification. This definition implies that the effects of all error sources, both random and systematic, must be considered.

2. Absolute accuracy is the error estimate for a single point, relative to the specified spatial reference system (SRS, for example, WGS-84). This accuracy includes errors from all known and expected sources. The error estimate from a particular error source is usually called an error component.
(Note: This definition is not in the USIGS Glossary.)

Accuracy

The degree to which information on a map or in a digital database matches true or accepted values. Accuracy pertains to the quality of data and the number of errors contained in a dataset or map. In discussing a GIS database, it is possible to consider horizontal and vertical accuracy with respect to geographic position, as well as attribute, conceptual, and logical accuracy. The effect of inaccuracy and error on a GIS solution is the subject of sensitivity analysis. Accuracy, or error, is distinguished from precision , which concerns the level of measurement or detail of data in a database.

Altimetry profile

Atmospheric refraction

The apparent displacement of an object that results from light rays from a source outside the atmosphere being bent in passing through the atmosphere. This results in all objects appearing to be higher above the horizon than they actually are. The magnitude of this displacement is greater when the object is near the horizon and decreases to a minimum assumed to be zero when the object is at the zenith. Also called **astronomic refraction error; celestial refraction.** See also **atmospheric refraction; refraction.**

Catalog

A collection of entries, each of which describes and points to a feature collection. Catalogs include indexed listings of feature collections, their contents, their coverages, and other metadata. Registers the existence, location, and description of feature collections held by an Information Community. Catalogs provide the capability to add and delete entries. At a minimum Catalog will include the name for the feature collection and the locational handle that specifies where this data may be found. The means by which an Information Community advertises its holdings to members of the Information Community and to the rest of the world. Each catalog is unique to its Information Community.
(Note: This definition is not in the USIGS Glossary.)

Circular Error

An accuracy figure representing the stated percentage of probability that any point expressed as a function of two linear components (e.g., horizontal position) will be within the given figure. Commonly used are Circular Error Probable (CEP [50 percent]), and CE (90 percent). A horizontal measurement on the ground, in feet or meters, defining a radius of a circle, within which an object of known coordinates should be found on an image. The CE value should have some measure of probability (P) associated with it. For example, a CE of 100 meters and .9 P, means that 90 percent of the time the object will fall within a circle having a radius of 100 meters.

Concatenated transformation

Sequential application of multiple transformations.
(Note: This definition is not in the USIGS Glossary.)

Coordinate Conversion

An exact transformation of position coordinates from one Spatial Reference System (SRS) to another. This term is used only when the SRS transformation is known exactly.
(Note: This definition is not in the USIGS Glossary.)

Coordinate Transformation

An approximate transformation of position coordinates from one Spatial Reference System (SRS) to another. For example, this term is used when the transformation coefficients are determined by least squares adjustment. This term is strictly used only when the SRS transformation is known only approximately. This term is loosely used when the SRS transformation is known either approximately or exactly.
(Note: This definition is not in the USIGS Glossary.)

Covariance matrix

A detailed form of position accuracy data, sometimes called a variance-covariance matrix. For three ground coordinates, a covariance matrix is a 3 by 3 matrix, with the matrix rows and columns each corresponding to the three coordinates. For just two horizontal ground 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.

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 ground coordinates, or the squares of the standard deviations. The off-diagonal cells contain the covariances between the errors in the corresponding ground 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.

Covariance matrices can be used to record absolute and/or relative accuracies. A covariance matrix for relative accuracy uses the three (or two) coordinates of one point for matrix rows and the three (or two) coordinates of the second point for matrix columns. A complete covariance matrix for N specific points would contain 3N rows by 3N columns.
(Note: This definition is not in the USIGS Glossary.)

Coverage

1. GIS coverages (including the special case of Earth images) are two- (and sometimes higher-) dimensional metaphors for phenomena found on or near a portion of the Earth's surface. Fundamentally, coverages (and images) provide humans with an n-dimensional (where n is usually 2, and occasionally 3 or higher) "view" of some (usually more complex) space of geographic features. The power of coverages is their ability to model and make visible spatial relationships between, and the spatial distribution of, Earth phenomena.

2. A coverage is a special case of (or a subtype of) feature.

Delineate

Two dimensional collection of feature position in an image.
(Note: This definition is not in the USIGS Glossary.)

Dimension

Exploitation

The evaluation of an image or multiple images to extract the information contained within the image(s) as it pertains to a specific list of questions or general categories of questions. Exploitation may result in the creation of a report or product to disseminate the information, whether it be to a requester or to a data base.

Extraction

Two or three dimensional collection of information from one or more images. In monoscopic extraction, extraction of each point is normally from one image. In stereoscopic extraction, extraction of each point is normally from one stereoscopic pair of images (or stereomates), sometimes called "conjugate feature extraction". In the case of range images, such as SAR or laser images, a one dimensional extraction of distance might be done, to determine the distance from the camera to a feature.
(Note: This definition is not in the USIGS Glossary.)

Feature

A digital representation of a real world entity or an abstraction of the real world. It has a spatial domain, a temporal domain, or a spatial/temporal domain as one of its attributes. Examples of features include almost anything that can be placed in time and space, including desks, buildings, cities, trees, forest stands, ecosystems, delivery vehicles, snow removal routes, oil wells, oil pipelines, oil spills, and so on. Features are usually managed in groups as feature collection.

Feature collection

A set of related features managed as a group.

Feature type

Class of features with common characteristics

Frame camera

A camera in which an entire frame or format is exposed through a lens that is fixed relative to the focal plane. See also **panoramic camera.**

Framework

In terms of software design, a reusable software template, or skeleton, from which key enabling and supporting services can be selected, configured and integrated with application code.
(Note: This definition is not in the USIGS Glossary.)

Image

The permanent record of the likeness of any natural or man-made features, objects, and activities reproduced on photographic materials. This image can be acquired through the sensing of visual or any other segment of the electromagnetic spectrum by sensors, such as thermal infrared, and high resolution radar.

Image column

Position of an image pixel in the horizontal direction, as that image is normally viewed. Sometimes referred to as (image) sample.

Image compression

An operation that, through various techniques, reduces the quantity of stored data needed to represent a digital image.

Image correlation

The matching of position and physical characteristics between images of the same geographic area from different types of sensors, between sensor images and a data base, or between two images from the same sensor.

Image data

All data collected by a sensor, which after processing, comprises an image.

Image enhancement

Any one of a group of operations that improve the detectability of features in an image. These operations include contrast improvement, edge enhancement, spatial filtering, noise suppression, image smoothing, and image sharpening.

Image geometry model

A mathematical model that specifies the mapping (or projection) between 3–D ground position coordinates and the corresponding 2–D image position coordinates. Such an image geometry model is alternately called an image sensor model, sensor model, imaging model, or image mathematical model. The term "sensor" is often used when the image is generated by a digital camera is thus originally digital. The word "camera" is usually used when the image is recorded in analog form, normally on film. Of course, film images can be later scanned or digitized and are then "digital".

The data used to define or establish such an image geometry model is often called image support data. However, image support data can also include non-geometry information.

An image geometry mathematical model can also be used to determine the correct ground position for an image position, if used with additional data. When a single (or monoscopic) image is used, this additional data normally defines the shape and position of the visible ground (or object) surface. For example, this additional data is often a single elevation or is grid elevation data, sometimes called a Digital Terrain Model (DTM).

Alternately, two stereoscopic images or multiple overlapping images can be used, that show the same ground point viewed from different directions. In this case, the two (or more) image geometry mathematical models can also be used, with the point coordinates in each individual image, to determine the corresponding 3–D ground position.
(Note: This definition is not in the USIGS Glossary.)

Image row

Position of an image pixel in the vertical direction, as that image is normally viewed. Sometimes referred to as (image) line.

Image perspective transformation

This product type includes video and hardcopy format showing several views of a scene with other than the original image geometry. This product type is used to simulate movement around a scene at ground or flight level.

Interface

A shared boundary between two functional entities. A standard specifies the services in terms of the functional characteristics and behavior observed at the interface. The standard is a contract in the sense that it documents a mutual obligation between the service user and provider and assures stable definition of that obligation.

Interferometric SAR

Lens distortion

Image displacement caused by lens irregularities and aberrations.

Linear Error

1. A one-dimensional error (such as an error in elevation) defined by the normal distribution function.

2. Vertical error at the target. The accuracy with which the elevation of the target can be determined. The LE at a point is a value such that the true elevation of the point can be expected to have the given value plus or minus LE with same degree of probability (usually 0.9 P).

3. In a Linear Error, we record that the value has a specified probability of having an error magnitude less than a specified distance.
(Note: This definition is not in the USIGS Glossary.)

Mensuration

1. The act, process, or art of measuring.

2. That branch of mathematics dealing with the determination of length, area, or volume.

3. The determination of length, area, volume, time, or other physical units.
(Note: This definition is not in the USIGS Glossary.)

Metadata

Data about the content, quality, condition, and other characteristics of data.

Monoscopic image

A single image taken of the target.

Orthorectification

1. The process of removing image displacement caused by tilt and terrain relief. Tilt, however, is not relevant in radar images.

2. The process of transforming one input image into an output image possessing a perpendicular parallel projection. Orthorectified images thus have a constant scale. The orthorectification process removes image tilt and displacement where applicable. Orthorectification requires using digital elevation data, usually in grid form. Orthorectification is sometimes termed "differential rectification" since the input image is separately rectified to cover each elevation grid cell.
(Note: This definition is not in the USIGS Glossary.)

Panorama

Panoramic camera

1. A camera which takes a partial or complete panorama of the terrain. Some designs utilize a lens which revolves about an axis perpendicular to the optical axis; in other designs, the camera itself is revolved by clockwork to obtain a panoramic field of view. See also frame camera.

2. A camera which takes a partial or complete panorama of the terrain. Some designs utilize a lens which revolves about an axis perpendicular to the optical axis; in other designs, the camera itself is revolved by clockwork to obtain a panoramic field of view. In still other designs, an assembly of lenses permits simultaneous, instantaneous recording of panoramic scenes. See also **frame camera.**
(Note: This definition is not in the USIGS Glossary.)

Perspective scene generation

See Image Perspective Transformation

Pixel

1. 2-dimensional picture element that is the smallest nondivisible element of a digital image.

2. In image processing, the smallest element of a digital image that can be assigned a gray level. Note: This term originated as a contraction for "picture element".

Photogrammetry

1. Use of aerial photographs to produce planimetric and topographic maps of the earth's surface and of features of the built environment. Effective photogrammetry makes use of ground control by which aerial photographs are carefully compared and registered to the locations and characteristics of features identified in ground-level surveys.

2. The science of mensuration and geometric adjustment of, an aerial photograph or satellite image. 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. Correction of imagery based on computational algorithms and measurement of geometrical position in an image.

3. The science of mensuration and geometric adjustment of metric or non-metric analogue or digital photographs. These photos can be taken in an industrial environment (sometimes referred to as terrestrial photogrammetry), as aerial images, or as imagery from space. Photogrammetry requires a mathematical model of the geometric and physical image formation process, computation of the internal geometry of an image, and location of the imaging position in space. The location of the imaging position in space is often done using ground control or reference points with known positions in a given SRS.
(Note: This definition is not in the USIGS Glossary.)

Profile, terrain

1. A vertical section of the surface of the ground, or of underlying strata, or both, along any fixed line.

2. Elevation of the terrain along some definite line. Elevations are measured at a sufficient number of points to enable defining the configuration of the ground surface.

Proprietary

Pushbroom camera

Rectification

1. In photogrammetry, the process of projecting a tilted or oblique photograph onto a horizontal reference plane. [Although the process is applied principally to aerial photographs, it may also be applied to the correction of map deformation.]

2. The geometric adjustment of image pixels to remove distortions caused by the imaging sensor and the geometry of the sensor to the ground. For the removal of terrain relief displacement see orthorectification.

3. The process of projecting a tilted or oblique image onto a selected plane or other surface. The plane is often horizontal, but can be tilted to achieve some desired condition, such as to better fit the local surface of the earth. Rectification of an ideal frame image is a plane-to-plane projection. Rectification of non-ideal images and of images with non-planar geometries (such as pushbroom images) includes corrections for the known image deviations from a plane, using an accurate mathematical model of the image geometry.
(Note: This definition is not in the USIGS Glossary.)

Reduced Resolution Data Set (RRDS)

Original, high-resolution imagery is useful for many applications (especially close-up displays), but for overall displays there may be too much data to process. For example, if the raw image consists of 16K X 16K pixels, it is impossible to fit this much data into the cathode ray tube refresh memory at once to build an overview display. Therefore, RRDSs are created (as a preprocessing step) and are used as input by overview displays whenever the original high-resolution data are impossible to use. An original 16K X 16K (=256 megabytes) may be reduced to 1K X 1K (=1 megabyte) for overview purposes.

Relative accuracy

1. An accuracy evaluation based on random errors in determining the positional accuracy of one point feature with respect to another feature.

2. Relative accuracy is the error estimate for the distance between two points, or the accuracy of one point with respect to the other point. This accuracy includes errors from all known and expected sources.
(Note: This definition is not in the USIGS Glossary.)

Request for Proposals (RFP)

An explicit OGC request to the industry to submit proposals to one of the Task Force Working Groups for technology satisfying a portion of the Abstract Specification. RFPs result in Implementation Specifications.

Spatial Reference System (SRS)

Description of how geographic objects are positioned in space.

Spherical Error

A spherical error records that a 3-D position has a specified probability of having a vector error magnitude less than a specified distance.

Stereoscopic images

1. Two images taken of a single target on one imaging pass to allow three-dimensional viewing of a target.

2. Two photographs [or images] with sufficient overlap of detail to make possible stereoscopic examination of an object or an area common to both.

3. Two or more images of the same object (or target) taken from different imaging positions in space, and thus different object viewing directions. These multiple images can be used to determine object position or dimensions in three-dimensions, or to allow three-dimensional viewing of the object. Two stereoscopic images are often taken from different points along one flight path, but there are also stereoscopic images from different flight paths.
(Note: This definition is not in the USIGS Glossary.)

Synthetic aperture radar (SAR)

An effective antenna is produced by storing and comparing the doppler signals received while the aircraft travels along its flight path. This synthetic antenna (or array) is many times longer than the physical antenna, thus sharpening the effective beam width and improving azimuth resolution. A synthetic aperture radar achieves azimuth resolution through computer operations on a set of coherently recorded signals such that the processor is able to function like a large antenna aperture in computer memory, thus realizing azimuth resolution improvement in proportion to aperture size. The SAR concept was introduced by C. Wiley (USA) in 1951.

Submodel

A part of a larger model, especially of an image geometry model.
(Note: This definition is not in the USIGS Glossary.)

Tile

1. Partition of a dataset based on the definition of a geographic area.

2. Part of an image based on rectangular or square image areas.
(Note: This definition is not in the USIGS Glossary.)

Tiling scheme

The scheme used to define tile shape and size, and unique tile identification number.

Transformation

1. (photogrammetry) The process of projecting a photograph (mathematically, graphically, or photographically) from its plane onto another plane by translation, rotation, and/or scale change. The projection is made onto a plane determined by the angular relations of the camera axes and not necessarily onto a horizontal plane. See also rectification.

2. (surveying) The computational process of converting a position from UTM or other grid coordinates to geodetic, and vice versa; from one datum and ellipsoid to another using datum shift constants and ellipsoid parameters. The survey position of a point is frequently given in several different grids or ellipsoids; local datum and Doppler-derived WGS 84 are common requirements.

3. The computational process of converting a position from a position given in one SRS into the corresponding position in another SRS. This transformation can require and use datum and ellipsoid parameters . The position of a point is frequently given in several different geodetic SRSs.
(Note: This definition is not in the USIGS Glossary.)

Whiskbroom camera

8. References

[1] OpenGIS Abstract Specification, OpenGIS™ Project Documents 99-100 through 99-116, available through www as< http://www.opengis.org/techno/specs.htm >

[2] Cook, Steve, and John Daniels, Designing Objects Systems: Object-Oriented Modeling with Syntropy, Prentice Hall, New York, 1994, xx + 389 pp.