Open Geospatial Consortium, Inc.

Date: 2008-08-20

Reference number of this document: OGC 08-054r1

Version: 0.1.0

Category: **OGC®** Discussion Paper

Editor: Max Martinez

OGC® OWS-5 Considerations for the WCTS Extension of WPS

Warning

This document is not an OGC Standard. This document presents a discussion of technology issues considered in an initiative of the OGC Interoperability Program. This document does not represent an official position of the OGC. It is subject to change without notice and may not be referred to as an OGC Standard. However, the discussions in this document could very well lead to the definition of an OGC Standard. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Preface

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

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

Contents

Page

Figures

Page

Tables

Page

OGC® OWS-5 Considerations for the WCTS Extension of WPS

1 Introduction

1.1 Scope

This OGC® document details considerations for using the WPS specification to define a standard coordinate transformation service.

In addition to describing the standard set of supported Processes that would allow a WPS to deliver the functionality implied by the latest WCTS draft specification, this document provides specific focus on the origins, forms, and application of geolocation transformations required for the georeferenceable imagery use case in the context of a coordinate transformation service.

This OGC document does not discuss the composition of or adjustment to any specific geolocation model which could serve as the basis for coordinate transformation in a georeferenceable imagery use case.

This OGC document does not discuss the relationship between geolocation information, as found associated with imagery data, and a geolocation transformation as made available by various OGC Web Services. Rather this document focuses on suggestions for obtaining and using the already formed geolocation transformations made available by other OGC web services for purposes of interacting with a WPS implementation supporting a WCTS-like profile.

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

1.2 Document contributor contact points

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

Name	Organization
Max Martinez	Leica Geosystems Geospatial Imaging, LLC

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2008-02-26	1.0.0	Max Martinez		Initial revision

1.4 Future work

Improvements in this document are desirable to reflect the complete exploration of the use of XSLT transforms in implementing the adapter pattern in XML.

2 References

The following documents are referenced in this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

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

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

OGC 07-055r1, OpenGIS® Web Coordinate Transformation Service
OGC 05-007r7, OpenGIS® Web Processing Service
OGC 07-112, GML 3.2.1 CR, Add implementation of ISO 19123 CV_ReferenceableGrid to GML
OGC 07-022r1, Observations and Measurements – Part 1 - Observation schema

ISO 19111, *Geographic information — Spatial referencing by coordinates*

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

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] and in OpenGIS® Abstract Specification Topic 6: *Schema for coverage geometry and functions* shall apply. In addition, the following terms and definitions apply.

3.1
coordinate conversion
coordinate operation in which both **coordinate reference systems** are based on the same **datum**

EXAMPLE Conversion from an ellipsoidal coordinate reference system based on the WGS 84 datum to a Cartesian coordinate reference system also based on the WGS 84 datum, or change of units such as from radians to degrees or feet to meters.

NOTE A coordinate conversion uses parameters which have specified values that are not determined empirically.

3.2
coordinate operation
change of **coordinates**, based on a one-to-one relationship, from one **coordinate reference system** to another

NOTE Supertype of coordinate transformation and coordinate conversion.

3.3
coordinate transformation
coordinate operation in which the two **coordinate reference systems** are based on different **datums**

NOTE A coordinate transformation uses parameters which are derived empirically by a set of points with known coordinates in both coordinate reference systems.

3.4
datum
parameter or set of parameters that define the position of the origin, the scale, and the orientation of a **coordinate system**

3.5
derived coordinate reference system
a coordinate reference system which is defined by applying a coordinate conversion to another coordinate reference system (A derived CRS inherits its datum from its base CRS)

4 Conventions

4.1 Abbreviated terms

SWE	Sensor Web Enablement
JPEG	Joint Photographic Experts Group
JPIP	JPEG 2000 Interactive Protocol
WPS	Web Processing Service
WCS	Web Coverage Service
WCS-T	Transactional Web Coverage Service

WCTS Web Coordinate Transformation Service

SOS Sensor Observation Service

CRS Coordinate Reference System

4.2 Used parts of other documents

This document uses significant parts of document [OGC 07-055r1]. To reduce the need to refer to that document, this document copies some of those parts with small modifications. To indicate those parts to readers of this document, the largely copied parts are shown with a light grey background (15%).

4.3 Data dictionary tables

The UML model data dictionary is specified herein in a series of tables. The contents of the columns in these tables are described in **Error! Reference source not found.**.

Table 1 — Contents of data dictionary tables

Column title	Column contents
Names (left column)	Two names for each included parameter or association (or data structure). The first name is the UML model attribute or association role name. The second name uses the XML encoding capitalization specified in Subclause 11.6.2 of [OGC 06-121r3]. The name capitalization rules used are specified in Subclause 11.6.2 of [OGC 06-121r3]. Some names in the tables may appear to contain spaces, but no names contain spaces.
Definition (second column)	Specifies the definition of this parameter (omitting un-necessary words such as “a”, “the”, and “is”). If the parameter value is the identifier of something, not a description or definition, the definition of this parameter should read something like “Identifier of TBD”.
Data type and value (third column) or Data type (if are no second items are included in rows of table)	Normally contains two items: The mandatory first item is often the data type used for this parameter, using data types appropriate in a UML model, in which this parameter is a named attribute of a UML class. Alternately, the first item can identify the data structure (or class) referenced by this association, and references a separate table used to specify the contents of that class (or data structure). The optional second item in the third column of each table should indicate the source of values for this parameter, the alternative values, or other value information, unless the values are quite clear from other listed information.
Multiplicity and use (right or fourth column) or Multiplicity (if are no second items are included in rows of table)	Normally contains two items: The mandatory first item specifies the multiplicity and optionality of this parameter in this data structure, either “One (mandatory)”, “One or more (mandatory)”, “Zero or one (optional)”, or “Zero or more (optional)”. The second item in the right column of each table should specify how any multiplicity other than “One (mandatory)” shall be used. If that parameter is optional, under what condition(s) shall that parameter be included or not included? If that parameter can be repeated, for what is that parameter repeated?

When the data type used for this parameter, in the third column of such a table, is an enumeration or code list, all the values specified shall be listed, together with the meaning of each value. When this information is extensive, these values and meanings should be specified in a separate table that is referenced in the third column of this table row.

The data type of many parameters, in the third table column, is specified as "Character String type, not empty". In the XML Schema Documents specified herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty.

The contents of these data dictionary tables are normative, including any table footnotes.

5 A WCTS Profile for WPS overview

"A Web Coordinate Transformation Service (WCTS) transforms digital geospatial data from one Coordinate Reference System (CRS) to another. The geospatial data transformed is digital feature data, including digital coverages. Such transformations include all the types of coordinate operations, including both coordinate 'transformations' and 'conversions'. This service inputs digital features in one CRS and outputs the same features in a different CRS. The service inputs include the identifications of the input and output CRSs, and optionally the coordinate transformation between these CRSs." [1)]

"A Web Processing Service (WPS) provides client access across a network to pre-programmed calculations and/or computation models that operate on spatially referenced data. The calculation can be extremely simple or highly complex, with any number of data inputs and outputs." [2)]

The notion of a profile to WPS that provides the functionality of a WCTS arose in the context of the OWS-5 Interoperability Test bed. One of the principal use cases to be examined in the SWE thread of this test bed was the "Georeferenceable Imagery" use case. This use case involved tasking a sensor to collect imagery over some area of interest, loading the collected imagery into a JPIP server upon notification of its arrival, referencing that JPIP server in publishing a description of the collected observation to an SOS, referencing the same JPIP server via a JPIP encoded coverage published to a WCS-T, and, finally, making the published coverage available in a catalog. This workflow is depicted in Figure 1.

[1)] OGC 07-055r1, pg. 4

[2)] OGC 05-007r7, pg. 1

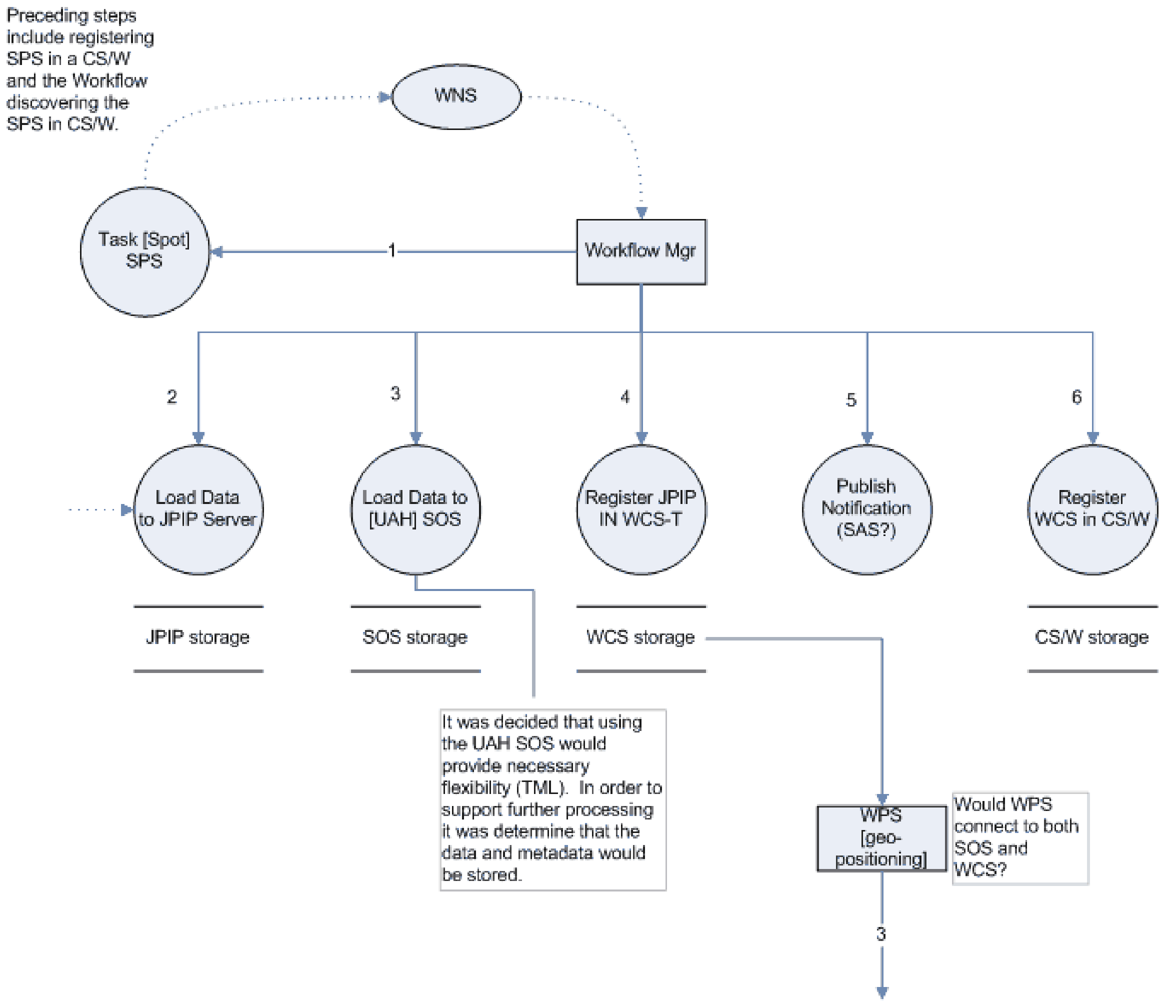

Figure 1 — OWS-5 Georeferenceable Workflow Dataflow Diagram

The publication to the catalog allowed subsequent discovery and display of the imagery by a thin client. In order to obtain ground coordinates for image locations of interest, the thin client was expected to rely on the services of a WPS.

Thus, the central functionality required of the WPS closely resembled the functionality provided by a WCTS.

The current WCTS specification shows the WCTS service inheriting one mandatory and one optional operation from its OGCWebService base:

1) GetCapabilities (mandatory)

2) GetResourceByID

and adding one mandatory and two optional operations in the WCTS specialization of an OGCWebService:

1) Transform (mandatory)

2) IsTransformable

3) GetTransformation

Of primary interest to this interoperability test bed is the central operation of the WCTS, the Transform operation. This operation is discussed in detail in the following section.

A secondary interest would be discovery of a WPS supporting a WCTS profile that had the capabilities required by the client, namely the WPS not only supported coordinate operations from/to image CRS'es or CRS'es derived from image CRS'es, but it also supported the coordinate operation method specific to the transformation(s) available for the coverage(s) with which the client was working.

Although this secondary interest was not explored in the test bed, issues relating to it seem to be implied by the discussion in section 7.3 of this document.

It is thought that the mapping of the other operations required or allowed by the WCTS specification offer no additional insight into interoperability other than what has been previously explored either here or in past interoperability test beds. Therefore, these operations are not discussed further in this report.

6 The WCTS Transform Operation

The WCTS Transform operation request requires the following parameters:

Table 2 — Parameters in Transform operation request

Names	Definition	Data type and value	Multiplicity and use
service service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely “WCTS”	One (mandatory)
request request	Operation name	Character String type, not empty Value is operation name, namely “Transform”	One (mandatory)
version version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
transformation Choice (none)	Identification or definition of desired coordinate operation	TransformationChoice data structure, see Table 3	One (mandatory)
gridCRS GridCRS	Definition of output GridCRS referenced by transformation	GridCRS data structure, see Annex G of WCS 1.1 [OGC 06-083r8]	Zero or one (optional) Include when transforming grid coverage
inputData InputData	Data to be transformed	InputData data structure, see 13.5 of [OGC 06-121r3]	One (mandatory)
interpolation Type [a] Interpolation Type	Identifier of spatial interpolation type which should be used to transform coverage	InterpolationType code list, see I.4.1 of [OGC 06-083r8] Values defined in service metadata	Zero or one (optional) Include when transforming coverage(s) using spatial interpolation
outputFormat OutputFormat	Identifier of output format to be used for the transformed features or coverage(s) [b]	MIME type, see Subclause 10.5 of [OGC 06-121r3] Allowed values defined in service metadata	Zero or one (optional) Include when output data not in input format
store store	Specifies if transformed data to be stored as remote resource(s)	Boolean Values are: true false	Zero or one (optional) Include when not default (false, return in response)

a Transforming a grid coverage usually requires resampling of coverage values to obtain the values at different grid points. For example, a coverage can be transformed from an unrectified grid to a georectified grid. When transforming a grid coverage, the SourceCRS and TargetCRS parameters shall completely specify the grids of the input and output coverages. These SourceCRS and TargetCRS parameters can be included in either the Transformation or the SourceAndTargetCRSs data structures.

b Allows WCTS servers to perform reformatting. The output formats supported by a WCTS are listed in the Contents section of the Capabilities document.

Table 3 — Parameters in TransformationChoice data structure

Names	Definition	Data type and value	Multiplicity and use
sourceAnd TargetCRSs SourceAnd TargetCRSs	References to SourceCRS and desired TargetCRS	SourceAnd TargetCRSs data structure, see Table 4	Zero or one (Mutually exclusive) [a]
transformation Transformation	Identification or definition of desired coordinate operation	Transformation data structure, see Table 5	Zero or one (Mutually exclusive) [b]
A Included when the server is permitted to select any coordinate operation which it knows about that can transform coordinates from the identified sourceCRS to the targetCRS. b Included when client is specifying a specific coordinate operation, possibly a user-defined coordinate operation. In this use, this element shall either: * Reference a well-known coordinate operation, whose definition is known to the WCTS server * Contain a URL from which that transformation definition can be retrieved, using GML encoding * Contain the coordinate operation definition object, using GML encoding, and a xlink:href value containing the URN that references this definition			

Table 4 — SourceAndTargetCRSs data structure

Names	Definition	Data type and value	Multiplicity
sourceCRS SourceCRS	Reference to input coordinate reference system	URI [a] Values defined in service metadata or in data known to client	One (mandatory)
targetCRS TargetCRS	Reference to desired output coordinate reference system	URI [a] Values defined in service metadata or in data known to client	One (mandatory)
a A URI shall reference a CRS as specified in Subclause 10.3 of [OGC 06-121r3].			

Table 5 — Transformation data structure

Names	Definition	Data type and value	Multiplicity and use
coordinate Operation (none)	Reference to or definition of coordinate operation	URI [a] or CC_CoordinateOperation [b] Values defined in service metadata or known to client	One (mandatory)
sourceCRS SourceCRS	Reference to or definition of input coordinate reference system	URI [a] or SC_CRS [c] Values defined in service metadata or known to client	Zero or one (optional) Include when operation is Conversion
targetCRS TargetCRS	Reference to or definition of output coordinate reference system	URI [a] or SC_CRS [c] Values defined in service metadata or known to client	Zero or one (optional) Include when operation is Conversion

a A URI shall reference a CRS or coordinate operation as specified in Subclause 10.3 of [OGC 06-121r3].

b The CC_CoordinateOperation UML class defines a coordinate operation and is specified in Subclause 12.4 of OGC Abstract Specification Topic 2. GML 3.1.1 [OGC 04-092r4] specifies how to XML encode the CC_CoordinateOperation class. A KVP encoding of the CC_CoordinateOperation class is not considered practical.

c The SC_CRS UML class defines a CRS and is specified in Subclause 9.4 of OGC Abstract Specification Topic 2. GML 3.1.1 [OGC 04-092r4] specify how to XML encode the SC_CRS class. A KVP encoding of the SC_CRS class is not considered practical.

The service, request, and version parameters are identical in WPS. The outputFormat and store parameters are covered by the Response Form of WPS. The concerns of our present discussion are the input parameters transformationChoice, gridCRS, inputData, and interpolationType.

When we actually apply this set of input parameters against a georeferenceable imagery use case, we find some awkwardness.

Firstly, the TransformationChoice does not cover this use case conveniently. Understanding this inconvenience requires a review of the definition of a CoordinateTransformation, a CoordinateConversion, and a DerivedCRS.

The TransformationChoice allows one to specify either both the source and target CRS or the desired coordinateOperation in its entirety. In the former case, one allows the transformation service to figure out all coordinate operation steps required to go between the source CRS and the target CRS, including choosing from among several options when available. In the latter case, the transformation service is told exactly what to do. What is lacking is the middle ground.

The reason we want to cover this middle ground in the georeferenceable imagery use case is because it alleviates the client and other servers holding geolocation transformations from doing additional work to precisely represent the coordinate operation required. This additional work, if made a requirement by the transformation service, would entail adding additional coordinate conversions to the ImageCRS end of the georeferencing transformation to reflect the modified GridCRS that a client would inevitably find themselves working with after a GetCoverage request. Since per specification, the coordinate transformation service will need to be able to figure out how to get from a sourceCRS to a targetCRS without any constraints at all, it seems that allowing it to do this with a partial constraint(s) should not be disallowed.

Another way of saying this is that the coordinate transformation service should be able to figure out that the ImageCRS that is the base CRS of the GridCRS with which the client is working matches the ImageCRS of the georeferencing transformation developed for and held with the georeferenceable coverage. It simply needs to be given the opportunity to do so.

Secondly, the ability to specify a coordinateConversion (in which case the sourceCRS and targetCRS of the transformation are required) seems to complicate the interface. It is difficult to understand under what circumstances that coordinate operation will not already be present in either the sourceCRS or targetCRS specified (because under these circumstances, one of them should be a DerivedCRS). It is also difficult to understand under what circumstances the server might have a CHOICE of coordinateConversions available for the specified sourceCRS and targetCRS (thus resulting in the desire of the client to direct the server to a particular coordinateConversion).

In our middle ground approach, coordinateConversions would never be given as constraint(s). It would only be necessary to specify coordinateTransformations (coordinateOperations that can effect a change of Datum) because it is only in the coordinateTransformation case that we truly could have multiple ways of constructing the transformation.

Note that constraint(s) rather than constraint has been specified because there may be multiple datum changes taking place in going from sourceCRS to targetCRS in the georeferenceable imagery use case (typically there will only be one transformation developed that transforms from the ImageDatum to a GeodeticDatum. If that GeodeticDatum does not match the GeodeticDatum of the desired targetCRS, an additional datum transformation will need to take place).

Thirdly, the sourceCRS may be implied by the input data under some circumstances (a coverage reference, for instance). This presents a case where specifying the sourceCRS would be redundant at best and inconsistent with the inputData in the worst case.

As we go down the road of sometimes allowing (in fact, requiring) the sourceCRS and targetCRS to be specified when one or more coordinateOperations are specified, we find that expressing the gridCRS becomes redundant to the explicitly stated or implied targetCRS, i.e., allowing transformations as partial constraint(s) requires the targetCRS to be specified explicitly in some cases which then makes the expressed gridCRS redundant.

We will additionally point out that allowing a gridCRS to be specified ONLY when transforming a coverage would seem to make it impossible to project features into a gridCRS supported by an imageCRS base via a coordinate transformation service.

Finally, we take note of the current treatment of inverse transformations in ISO 19111:

> Many but not all coordinate operations (from CRS A to CRS B) also uniquely define the inverse coordinate operation (from CRS B to CRS A). In some cases, the coordinate operation method algorithm for the inverse coordinate operation is the same as for the forward algorithm, but the signs of some coordinate operation parameter values have to be reversed. In other cases, different algorithms are required for the forward and inverse coordinate operations, but the same coordinate operation parameter values are used. If (some) entirely different parameter values are needed, a different coordinate operation shall be defined.

If this statement is implying that in some cases the same coordinateOperation will be used for both forward and inverse operations, it would be quite difficult to elicit an inverse coordinateTransformation from a coordinate transformation service by specifying a coordinateOperation (particularly via URI) but NOT specifying the sourceCRS and targetCRS (as the current WCTS specification would require). Thus, we find another instance where both the coordinateOperation and souceCRS and targetCRS need to be specified.

Thus, the input parameters to the Execute request for the WCTS Profile of WPS are as specified in Table 5.

Table 6 — Input parameters to Execute request for WCTS Profile of WPS

Names	Definition	Data type and value	Multiplicity and use
coordinate Operation (none)	Reference to or definition of coordinate transformations that must be used in constructing a full coordinateOperation from the specified or implied sourceCRS to the specified or implied targetCRS	URI [a] or CC_CoordinateOperation [b] Values defined in service metadata or known to client	Zero or more (optional) Include when the transformation service may have a choice of coordinate transformation for effecting a datum change and the client would like to pre-determine how that choice is made.
sourceCRS SourceCRS	Reference to or definition of input coordinate reference system	URI [a] or SC_CRS [c] or GridCRS data structure, see Annex G of WCS 1.1 [OGC 06-083r8] Values defined in service metadata or known to client	Zero or one (optional) Include when no CRS is associated with the inputData and (when no coordinateTransforrmation is specified or when multiple coordinateTransformations are specified or (when one coordinateTransformation is specified and the sourceCRS of the specified coordinateTransformation does not represent the sourceCRS of the inputData))
targetCRS TargetCRS	Reference to or definition of output coordinate reference system	URI [a] or SC_CRS [c] or GridCRS data structure, see Annex G of WCS 1.1 [OGC 06-083r8] Values defined in service metadata or known to client	Zero or one (optional) Include when no coordinateTransforrmation is specified or when multiple coordinateTransformations are specified or (when one coordinateTransformation is specified and the targetCRS of the specified coordinateTransformation is not the desired targetCRS for the outputData)
inputData InputData	Data to be transformed	InputData data structure, see 13.5 of [OGC 06-121r3]	One (mandatory)
interpolationType [a] Interpolation Type	Identifier of spatial interpolation type which should be used to transform coverage	InterpolationType code list, see I.4.1 of [OGC 06-083r8] Values defined in service metadata	Zero or one (optional) Include when transforming coverage(s) using spatial interpolation

Names	Definition	Data type and value	Multiplicity and use

a A URI shall reference a CRS or coordinate operation as specified in Subclause 10.3 of [OGC 06-121r3].

b The CC_CoordinateOperation UML class defines a coordinate operation and is specified in Subclause 12.4 of OGC Abstract Specification Topic 2. GML 3.1.1 [OGC 04-092r4] specifies how to XML encode the CC_CoordinateOperation class. A KVP encoding of the CC_CoordinateOperation class is not considered practical.

c The SC_CRS UML class defines a CRS and is specified in Subclause 9.4 of OGC Abstract Specification Topic 2. GML 3.1.1 [OGC 04-092r4] specify how to XML encode the SC_CRS class. A KVP encoding of the SC_CRS class is not considered practical.

7 Georeferencing Transformations

An abstraction that is central to the georeferenceable imagery use case is the georeferencing transformation. The georeferencing transformation is a coordinate transformation that relates image space to ground space. It is this abstraction that allows georeferenceable imagery to be related to other geospatial data.

The georeferencing transformation is analogous to coordinate transformations that effect a geodectic datum shift between two ground based coordinate reference systems. In the case of the georeferencing transformation, however, we are shifting between an image datum and a geodetic datum (or sometimes directly to a projected system supported by a geodetic datum).

Another difference is that unlike coordinate transformations that relate two geodetic datums, georeferencing transformations are not “well-known”. The georeferencing transformation for every image has to be developed independently, either from geolocation metadata associated with the image or by correspondence with other known geospatial data.

This means that although a WPS can be implemented to understand well-known operation methods used in constructing georeferencing transformations, the parameters to those methods vary with each image and thus must be made known to the WPS for each georeferencing transformation the WPS is expected to execute.

Whence does this georeferencing transformation come? How is it encoded? These are the questions explored in OWS-5.

7.1 Possible Sources of Georeferencing Transformations

The ultimate source of a georeferencing transformation is its development from geolocation metadata or correspondence with other known geospatial data. This process is not our concern in OWS-5, but rather, once developed, where does this transformation persistently reside.

The answer varies by OGC service and GML encoding.

7.1.1 WCS Transformation Location

The WCS specification confronts this question directly by allowing this transformation to be made available in the SpatialDomain of the DescribeCoverage response as well as the coverage metadata returned by the GetCoverage response. In both cases, the transformation is to be encoded as a gml:Transformation or gml:ConcatenatedOperation. This encoding presents some problems when it is desired to directly use an encoding expressed somewhat differently, namely a SensorML representation of the transformation. This issue is discussed further in section 7.2.

NOTE The changes specified in OGC 07-067r5 (WCS 1.1.2) are required to properly allow the encoding of the georeferencing transformation as a gml:Transformation or gml:ConcatenatedOperation.

7.1.2 JPEG 2000 and JPIP Transformation Location

OWS-5 explored the use of JPIP encoded coverages extensively in the Georeferenceable Imagery use case. As a result, there was discussion of the transport of image metadata both in the JPEG 2000 and JPIP encoding cases.

The GML in JPEG 2000 specification employs several box types defined by both Part 1 and Part 2 of the JPEG 2000 standard for purposes of storing and referencing GML encoded data associated with a JPEG 2000 codestream.

Part 9 of the JPEG 2000 standard (JPIP) defines a mapping of these boxes to "data-bins" that is meant to enable transport efficiency "while allowing file-centric concepts to be replaced by streaming-centric concepts where appropriate".

EDITOR'S NOTE During the course of the test bed, numerous references were made to a so-called "JPIP back channel". Use of this term could not be found in any specification document and the editor assumes that what was meant was the labeled boxes defined by the standard for purposes of storing data with the codestream.

Thus, one would conclude that the approach to including SensorML in JPEG 2000 should be the same for both file based encodings as well as the stream-based encoding (JPIP).

An approach taken during the test bed was to simply place the SensorML document in the gml.root-instance box. This seems like a good start, but without any tie-in to the other information being described, it seems like this could lead to confusion

a) between SensorML process models meant to describe geolocation and SensorML process models meant to describe the collection of the range information

b) between SensorML process models meant to describe geolocation and similar information currently being introduced to the coverage description via an encoding of a ReferenceableGridCoverage in GML (OGC 07-112)

What seems more appropriate is to purse the intent of OGC 07-112 except in a more general manner:

```
<complexType name="ReferenceableGridType">
    <annotation>
```

```xml
		<documentation>An implementation of CV_ReferenceableGrid of ISO 19123. The transformation
	provides the datum shift from the ImageDatum underlying the Grid geometry (either explicitly or by
	association to an ImageCRS to which the GridCRS is also associated) to an external CRS (if this
	external CRS has an earth-based Datum, the ReferencableGrid is referred to as a
	GeoreferencableGrid).
		</documentation>
	</annotation>
	<complexContent>
		<extension base="gml:GridType">
			<choice>
				<element name="transformation" type="gml:CoordinateOperationRefType"/>
				<!-- Do these other choices really need to remain in here? Aren't they simply specific
				ways of expressing this transformation? -->
				<element name="gridPoints" type="gml:GridPointListPropertyType"/>
				<sequence>
					<element name="origin" type="gml:PointPropertyType"/>
					<element name="generalGridAxis" type="gml:GeneralGridAxisPropertyType"
						maxOccurs="unbounded"/>
				</sequence>
			</choice>
		</extension>
	</complexContent>
</complexType>
```

This approach is not without its own set of issues. These issues stem from the choice of ISO 19123 to eschew the use of the ISO 19111 model in properly describing the “internal” grid coordinate reference system of quadrilateral grid coverages. Instead, only the “external” coordinate reference system is exposed and its association to the internal system is achieved through properties of the grid itself, rather than through properties of a coordinate reference system definition associated with the grid. This makes it very messy to describe the semantics of this transformation in the same way we have chosen to do it for our WPS and the WCS from which it was retrieved.

Assuming those issues could be resolved, we would still need a way to allow a SensorML document to be treated as a gml:_CoordinateOperation. This is discussed in section 7.2.

7.1.3 SOS Transformation Location

Although the intent of the Georeferenceable Imagery use case was for both the thin-client and thick-client to interact with the WCS, a logical question to ask is “if a client had chosen to interact with the SOS, where might that client find the georeferencing transformation?”.

Informal discussion seemed to indicate that the place where this would be found would be the location element of the feature of interest of the observation. This would seem consistent with the O&M document (last paragraph of section 6.4):

> For these reasons, the generic Observation class does not have an inherent location property. Relevant location information should be provided by the feature of interest, or by the observation procedure, according to the specific scenario.

But if one looks more closely at GML, one finds that:

1) gml:location, as it relates to a FeaturePropertyType is described with the following comment which seems like a bit of a stretch for a georeferencing transformation encoded as a SensorML ProcessModel:

    The value of the gml:location property describes the extent, position or relative location of the feature.

2) gml:location appears to have been deprecated (in version 3.2.1)

One could make a case that this seems directly related to the GMLJP2 discussion above. The observation result in the georeferenceable imagery use case should be a coverage and the whole point of the GMLJP2 discussion is that proper definition of a ReferenceableGrid geometry for use in a ReferenceableGridCoverage provides a logical location for a georeferencing transformation in GMLJP2. If consensus were reached on that, one would expect to find the SensorML ProcessModel that describes the geolocation of an observation result in the same place, i.e, one of the choice elements that extend the GridType base of the ReferenceableGrid geometry describing the coverage domain of that observation result.

7.2 SensorML as a gml:_CoordinateOperation

We now wrestle with the issue of treating a SensorML ProcessModel that represents a geolocation transformation as a gml:_CoordinateOperation.

During the test bed, this was "accomplished" by a modification to the WCS schemas to add the indicated additional choice for describing a transformation highlighted in red below:

```
<element ref="wcs:Transformation">
  <annotation>
    <documentation>
      Georeferencing coordinate transformation for unrectified coverage, which should be included when available
      for a coverage that is georeferenced but not georectified. When included, this Transformation will specify
      the variable spatial resolution of this non-georectified image. To support use cases 4, 5, 9, and/or 10
      specified in Annex H, a WCS server needs to use a georeferencing coordinate transformation for a
      georeferenced but not georectified coverage. However, a WCS server may not support those use cases, not
      use a georeferencing transformation specified in that manner, or not make that transformation available to
      clients.
    </documentation>
  </annotation>
</element>
<element name="OtherTransformation">
  <complexType>
    <sequence>
      <any processContents="skip" />
    </sequence>
    <attributeGroup ref="xlink:simpleLink" />
    <attribute name="xmlschema" />
  </complexType>
```

</element>

This seems undesirable in the long run (in programming language terms, it is not what one would consider “type-safe” because an “Other Transformation” could be just about anything).

What this context really calls for is the XML equivalent of the adapter pattern. Although the use of XML Schema’s substitutionGroup allows using a specialization in place of a more generalized base class, what more often evolves in the real software development world are not clean logically related type hierarchies, but rather independently and simultaneously-developed, implementations which are based on similar, but not identical, abstractions. Software developers will then typically adapt these independently developed API libraries to a common abstraction in order to easily utilize the specific behaviours or approaches represented by the individual implementation from higher level applications.

The XML equivalent of the adapter pattern might involve a link to the adapted element instance, a schema for that instance, and an XSLT StyleSheet that "adapts" the adapted element instance to the element instance required. In this particular case it would map a SensorML process model to a gml:_CoordinateOperation.

The advantage of approaching harmonization in this way in certain circumstances is that it would:

1) Acknowledge real world realities that subsystems are often developed independently and there is a cost to harmonizing them with the main type hierarchy

2) Force sub-implementations to demonstrate that they are equivalent to the main type hierarchy concepts by requiring them to provide XSLT transforms to these head elements

3) Allow implementations to be built based on the main type hierarchy and guarantee reasonable handling of adapted sub-implementations because of the required XSLT transform that would allow the data to be consumed as the main type hierarchy describes

4) Allow implementations to be built based on the sub-implementations and to easily consume sub-implementation data through the main type hierarchy (because it is not a requirement to apply the XSLT transform, only to provide it)

An encoding profile of WCS is an example of this adapter pattern in OGC schemas (put aside for a moment that it is not clear that any encoding profiles actually exist). Rather than formally describing this adaptation in an XSLT transform, the encoding profile describes the adaptation in documentation. Thus, implementations cannot consume these encodings dynamically (they must be pre-programmed to consume them).

Adapter patterns could then and should be used to treat the two choices that compose the extension of gml:GridType by gml:ReferenceableGridType as a more generalized gml:_CoordinateOperation.

Finally, this adapter pattern approach could be used to treat a GridCRS and any future simplified template for specifying a complex CRS definition as an AbstractCRS (this cannot be done currently and it is awkward). In order to define general CRS inputs for the WPS, the gml:CRSPropertyType needed to be modified thusly in the ows5 namespace:

```
<complexType name="CRSPropertyType">
    <annotation>
        <documentation>CRSPropertyType is a property type for association roles to a CRS abstract
coordinate reference system or a wcs:GridCRS, either referencing or containing the definition of that
CRS.</documentation>
    </annotation>
    <choice minOccurs="0">
        <element ref="gml:AbstractCRS"/>
        <element ref="wcs:GridCRS"/>
    </choice>
    <attributeGroup ref="gml:AssociationAttributeGroup"/>
</complexType>
```

This type of modification should not be necessary.

7.3 Schema Constraints

Modifying the WCS schema to allow a SensorML document reference to be treated as the georeferencing transformation of a coverage eventually raised an additional issue when this document reference hit the WPS.

The issue had to do with schema constraints. The schema for SensorML could certainly be used to validate the SensorML document containing the georeferencing transformation. But that schema is much looser than we would like in this context. What we want in this context is a schema that describes only SensorML ProcessModels that represent georeferencing transformations. This schema is not currently available.

Additionally, if the WPS had wanted to accept only certain kinds of georeferencing transformations, for instance transformations expressed as a collection of Rational Polynomial Coefficients as happened in this use case, then the only schema available for this use was not an XML Schema Document, but rather a Schematron schema document.

It was eventually concluded that the WPS specification did not mean to limit the types of schema documents that could be referenced to constrain complex data inputs and outputs, but there was no conclusion reached on what exactly this would mean for WPS implementers or clients.

7.4 Direction and Inversion of Coordinate Transformations

The direction and inversion of coordinate transformations continues to confound developers as they approach ISO 19111 derived GML elements for the first time. As previously pointed out, ISO 19111 seems to consider that the forward and inverse transformations are represented by the same coordinate operation unless the parameters need to be modified significantly.

It is noted, however, that the sources of georeferencing transformations discussed above typically account for the delivery of only a single coordinate operation. Therefore, if there was ever a case where the forward and inverse transformation needed to be represented by different sets of parameters, the best practice would be to still treat this as a single coordinate operation that was invertible. It would simply be the case that some of the parameters would not be used in transforming from one direction or the other.

The reason this approach must be taken is that services supplying access to this georeferencing transformation do not currently allow the client to assert whether they require the georeferencing transformation to go from ground space to image space or the other way around. In the absence of any changes, an invertible transformation should be constructed and associated with the coverage whenever possible so that clients would be likely to receive an invertible georeferencing transformation and, therefore, be able to use it in any way they please.

8 Coordinate Reference Systems

The principal coordinate reference systems of concern in the Georeferenceable Imagery use case are the ImageCRS and the GridCRS derived from it. This use case provides an opportunity for an in-depth exploration of these forms of coordinate reference system.

8.1 ImageCRS

In the georeferenceable imagery use case, an image is a gridded coverage for which the geometry of the domain is implicit in the organization of the range data. Any persistent encoding of the image must necessarily include an exact description of how image coordinates in the ImageCRS (or a GridCRS that might be derived from an ImageCRS as discussed in section 8.2) are related to the access of range data in the persistent image.

What was concluded about this ImageCRS is that it most definitely needed to be uniquely identifiable. This is required to allow georeferencing transformations to be discovered based on the ImageCRS (or GridCRS with an ImageCRS base) of spatial data.

After looking at the recommendations in OGC 05-027, it was decided that use of a URN to reference an ImageCRS is not currently advisable. The reason for this is that the method recommended for constructing the URN cannot guarantee that the URN will uniquely identify the ImageCRS. The document states:

> The URN value “urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:TBD” shall reference the definition of an (unrectified) image CRS with its origin in the centre of the first pixel in the image file. This image CRS shall be for the image, or image group, whose alphanumeric character string identifier is substituted for the “TBD” in this URN.
>
> NOTE 1 This document does not specify any format for an image alphanumeric identifier, since many different formats are used for such identifiers. One such format could concatenate a camera identifier with the image collection date and time.

Without authoritatively defining the method of forming an image alphanumeric identifier, this recommendation is not useful for guaranteeing construction of a URN that uniquely identifies the ImageCRS.

Thus, a reasonable alternative approach of using a URL that references the ImageCRS definition document at the coverage server has been adopted.

8.2 GridCRS

The wcs:GridCRS was a clever invention. It allows the affine conversion from the grid coordinate reference system to a ground coordinate reference system to be carried around inside the GridCRS definition.

It also allows georectified imagery to be distinguished from imagery that is merely associated with an affine transformation that represents a crude correspondence between image space and ground space based on a handful of ground control points. ISO 19123 misleads the reader into believing that the mere fact that the available relationship between the grid coordinate reference system and the ground is expressed as an affine transformation makes the grid rectified. In reality, it comes down to whether the grid coordinate reference system was derived from the ground system, as happens when an image is georectified, or whether the grid coordinate reference system existed prior to having any knowledge to its relationship to the ground.

In OWS-5, we were able to put the GridCRS to another good use. We found that it was logical to construct a GridCRS with a base coordinate system of an ImageCRS under conditions where subsets, rescales, etc. were taking place against georeferenceable imagery. This allows all images derived from the same original coverage to later be related to one another.

Furthermore, this also allows the original coverage’s georeferencing transformation to be utilized in coordinate transformations on images derived from that original coverage without any further concatenation of additional coordinate operations as long as the a GridCRS for the derived image is available. The consequence of this is a greatly reduced onus on both clients utilizing georeferenceable imagery and servers holding and serving georeferencing transformations for that imagery in terms of needed understanding and manipulation of georeferencing transformations.

Specifically in the OWS-5 georeferenceable imagery use case, a client could pass a georeferencing transformation to a WCS-T, which could then hold it in association with the published coverage, deliver it to another client upon a GetCoverage request, which client could then pass it on to the WPS for coordinate operations to take place. None of clients or servers handling the georeferencing transformation prior to the transformation service needs to understand the form of the georeferencing transformation.

9 Axis Order

SensorML currently allows the attributes of a Quantity (namely uom) and the order of the Vector coordinates to differ from its specified “referenceFrame”. This poses impediments to interoperability unless it can be clearly stated how the coordinate axes rearrangement can be related to the arrangement of the axes in the original referenceFrame definition. The recommendation is that the “axisID” attribute for the Quantity element of the coordinate should assume the value of the “axisAbbrev” element in the definition of the referenceFrame for the axis to which the coordinate corresponds.