GDI NRW
Geodateninfrastruktur Nordrhein-Westfalen

Testbed II

Web Coordinate Transformation Service

Februar – Dezember 2002

Dokumentation
Version 1.0

Teilnehmer

AED Graphics

con terra

FhG ISST

GIA

GIUB

ibR

IfGI

interactive instruments

lat/lon

Open GIS Consortium Inc.

Date: 2002-11-25

Reference number of this OpenGIS© project document: OGC 02-061r2

Version: 0.1.0

Category: Proposed OpenGIS© OGC Implementation Specification

Editor: Markus U. Müller; Andreas Poth

Web Coordinate Transformation Service Draft Implementation Specification

Copyright notice

Warning

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

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OpenGIS© Implementation Specification
Document stage: Proposed Implementation Specification
Document language: Englis h

OGC 02-061r2

Contents

i. Preface

This document describes the OGC Web Coordinate Transformation Service (WCTS) operations.

Most geo-spatial data are stored in different coordinate reference systems (CRS). To enable services to use data from different sources stored in different CRS each service has to access a transformation service. At the moment each OGC web service has to access its own transformation implementation or access a service via COM, CORBA or RMI as defined in „Coordinate Transformation Services Implementation Specification" (CTS) [5].

To offer the largest amount of interoperability and a standardized access to the transformations of CRS in a general way through the inter-/intranet this specification defines web service interfaces for performing transformations of geo-spatial data from one CRS to another. WCTS is closely linked to the Geography Markup Language (GML) Version 3 which describes an XML-encoding of geo-spatial data.

To do this a WCTS shall be able to handle the following requests:

1. Which Geometry and Coverage types of GML3 are supported?

2. Which CRSs are known by the WCTS as source and target CRS of a transformation?

3. Which transformations are known by the WCTS? This is a necessary information because in many cases there exists more than one way to transform coordinates from one CRS to another.

4. Is the WCTS capable to use user defined CRS and/or transformations?

5. Is the transformation between two sets of data with spedifict CRS possible?

6. Which 'transfomation steps' will be performed to transform a coordinate from one CRS to another?

7. Performing a transformation from one CRS to another.

ii. Submitting organizations

This proposed specification is being submitted by the following organization:

lat/lon

Meckenheimer Allee 176

53115 Bonn

Germany

iii. Submission contact points

All questions regarding this submission should be directed to the editors or the submitters:

CONTACT	COMPANY	ADDRESS	PHONE/FAX	EMAIL
Andreas Poth	lat/lon	Meckenheimer Allee 176, 53115 Bonn, Germany	++49 +228 73 9666	poth@lat-lon.de
Markus U. Müller	lat/lon	Meckenheimer Allee 176, 53115 Bonn, Germany	++49 +40 428453476	m.mueller@lat-lon.de

iv. Revision history

Date	Release	Author	Paragraph modified	Description
2002-07-26	0.0.1	Andreas Poth		initial version
2002-08-01	0.0.2	Markus Müller	7-11	added paragraphs 7-11
2002-08-14	0.0.3	Andreas Poth	6-11	chapter content slightly modified; schema definitions changed; example requests and responses added
2002-09-15	0.0.4	Markus Müller	i, 3	included changes requested by CRS SIG for public discussion paper status
2002-11-25	0.1.0	Markus Müller	All	Adapted spec to GML3 and included some changes responding to NASA

				comments on discussion paper. Dropped support of WKT.

v.   Changes to the OpenGIS Abstract Specification

No further revisions to the OGC Abstract Specification are required.

Foreword

Attention is drawn to the possibility that some of the elements of this standard may be the subject of patent rights. Open GIS Consortium Inc. shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

This version of the specification cancels and replaces all previous versions.

Normative Annexes

Annexes A and B are normative.

0   Introduction

Transformation of geo-spatial data from one CRS to another is a frequent requirement when integrating data from different sources into one application process. But not all applications – we believe most of them – are not capable to do this. The following use cases demonstrate how a web based transformation service can solve this problem.

0.1   Use Cases (Informative)

This chapter describes several use cases that will clarify the requirements to a WCTS and their realisation within this specifications.

The base of all use cases is the following scenario:

There is a map to create that uses EPSG:4326 as CRS. The borders of the Federal Republic of Germany (layer 1), the major rivers of the Federal Republic of Germany (layer 2) and the largest cities of the Federal Republic of Germany (layer 3) shall be used. The borders of the Federal Republic of Germany are available in EPSG:31467 (Gauß-Krüger band 3). The major rivers are stored in EPSG:4230 (International 1924) and the largest cities in EPSG:23032 (UTM band 32)

0.1.1   Desktop Application

The term „Desktop Application" describes a GI-application that is running stand-alone on a PC and that accesses the data it uses through a file system. It is assumed that the application is allowed to access a network (intra-/internet) where a WCTS is available. It is also assumed that all layers defined within the scenario are available through a local data store.
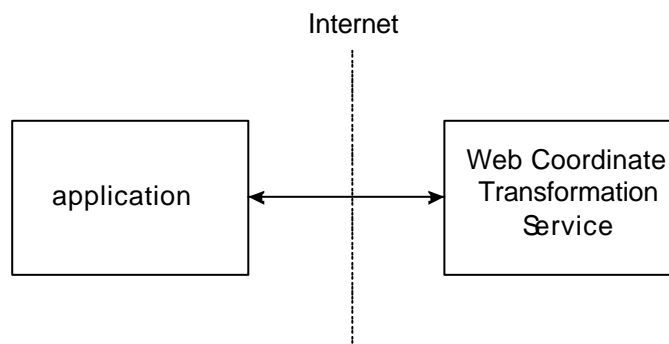
Fig. 1 use case 1: application - WCTS interaction

The following steps have to be performed at this use case:

1. The application defines a new map with the target CRS.

2. The application loads layer 1 and recognises that coordinates are not stored in EPSG:4326.

3. The application connects the WCTS through the network and requests its capabilities. Using the capabilities the application evaluates if a transformation from EPSG:31467 to EPSG:4326 can be performed by the WCTS. (This can also be done by sending an IsTransformable request to the WCTS).

4. The application sends the geo-spatial data assigned to layer 1 to the WCTS requesting their transformation to EPSG:4326.

5. The WCTS sends the transformed data back to the application.

6. Steps 2 to 5 will be repeated for layer 2 and 3.

7. The application will render the map.

0.1.2 WMS

It is assumed that the map defined above will be requested by a client through a network using a valid WMS request. Like in use case 1 it is also assumed that all required data are stored in independent sources that can be accessed by the WMS directly (integrated WMS).
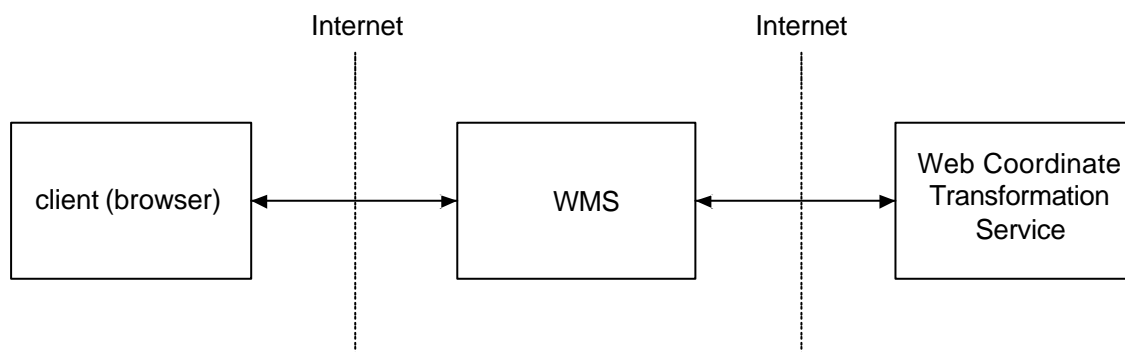
Fig. 2 use case 2: WMS - WCTS interaction

The sequence of steps to perform is similar to use case 1.

First, however, the client must inquire the WMS (Capabilities Request) whether it can render the map in the desired coordinate reference system. Afterwards the map, as described in use case 1, will be rendered and passed to the client (web browser) via the inter-/intranet.

 required steps are:

1. The client defines the parameters for a new map.

2. The client posts a GetMap request to a WMS.

3. The WMS states that the requested layer is not yet present in the correct CRS.

4. The WMS already knows that the WCTS offers the necessary transformation; it therefore sends the data of the first layer for transformation to the WCTS.

5. Step 4 will be repeated for layer 2 and 3.

6. The WMS renders the map and sends it back to the client (web browser)

0.1.3 Service Chain

The requirements correspond to those of use case 2; but the renderable data are not offered directly by the WMS. Therefore the WMS must request them via the Internet from a WFS ($\rightarrow$ component WMS). This supplies the data in the CRS specified above.
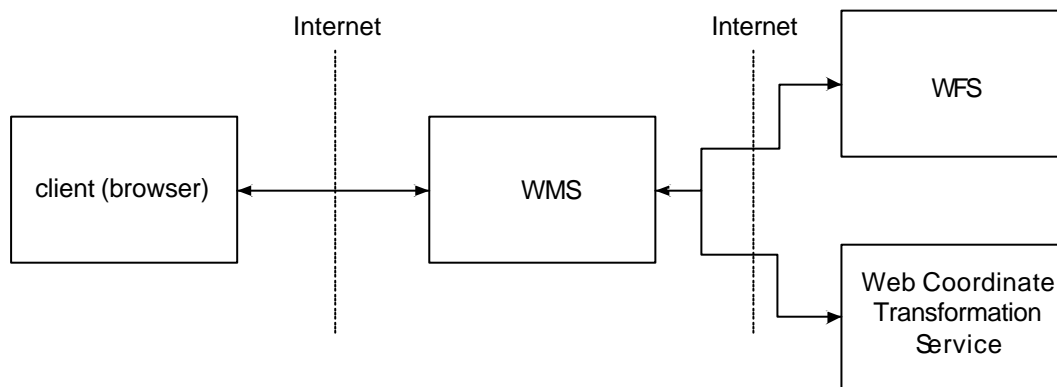
Fig. 3 use case 3: WMS - WFS - WCTS interaction

In this case the WMS sends a request to the WCTS, providing the input data as a getFeature-Request for a specific WFS. The WCTS then sends this request to the WFS, transforms the coordinates and sends those as respone to the WMS. The remaining steps will be performed as described above.

In principle the model is expandable for further levels e.g. in the case of a cascading WMS or a service chain.
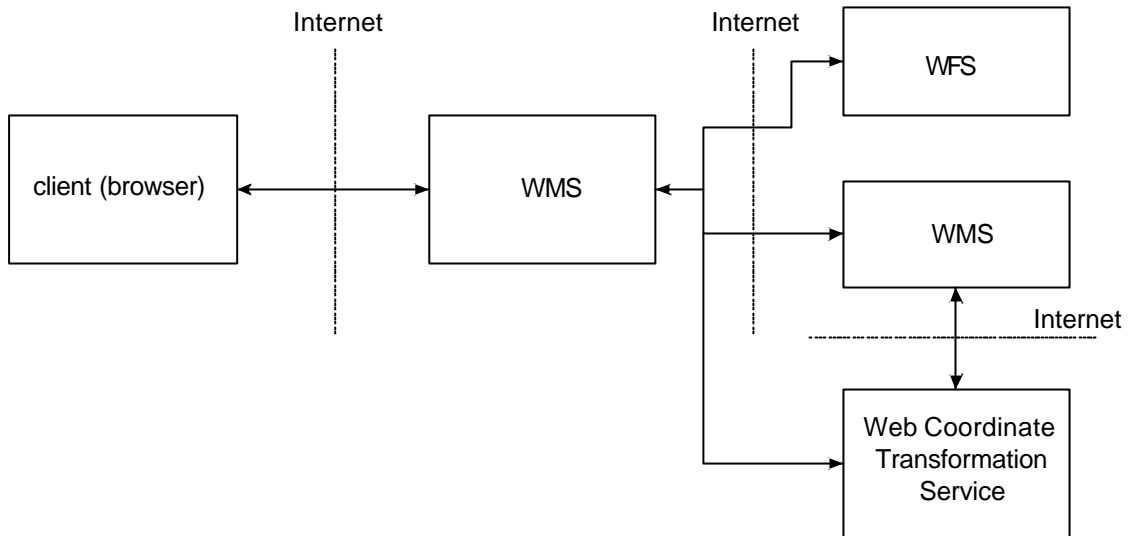
Fig. 4 use case 3: service chain

It must be considered that to the WMS addressed through the cascade already must supply the requested Layer in the target CRS, since the transformation of a raster image would be too expensive and the result possibly does not correspond to expectations (e.g. by substantial distortions of the labeling).

Web Coordinate Transformation Service Implementation Specification

1   Scope

This standard specifies the transformations of geospatial coordinates from one Coordinate Reference System (CRS) into another by means of a Web Service. Frequently geospatial data are accessible only in one coordinate references system. If geospatial data from different sources shall be processed together this is only possible if all relevant data are available in the same reference system. If this is not the case, the data must be transformed into a uniform CRS.

Not each service and/or not each client possesses the ability to transform geo-spatial coordinates from one CRS o another. In practice this makes the realisation of distributed, interoperable GI systems frequently more difficult or impossible.

This specification describes a Web Service based on the OGC CTS implementation [5] specification, ISO DIS 19111 [6] and Abstract Specification Topic 2 [7] which allows applications and other Web Services to transform geospatial coordinates from one CRS into another. Transformations shall both take place between CRS that are known by the WCTS and between CRS, which are defined by the user (client).

This standard is applicable to all geospatial data which are encoded as GML 3 which is the current version of the GML standard.

2   Conformance

Conformance with this specification shall be checked using all the relevant tests specified in Annex B (normative). The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in ISO 19105: Geographic information — Conformance and Testing.

3   Normative references

[1]   Fielding et. al., "Hypertext Transfer Protocol – HTTP/1.1," IETF RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt.

[2]   Bray, Hollander, Layman, eds., "Namespaces In XML", January 1999, W3CRecommendation, http://www.w3.org/TR/2000/REC-xml-names.

[3]   Bradner, Scott, "RFC 2119 Key words for use in RFCs to Indicate Requirement Levels," March 1997, ftp://ftp.isi.edu/in-notes/rfc2119.txt.

[4]   Percivall, George, ed., "The OpenGIS Abstract Specification, Topic 12: OpenGIS Service Architecture", 2002

[5]   Daly, Martin (ed.), "OpenGIS Implementation Specification #01-009: Coordinate Transformation Services. Revision 1.00, January 2001

[6]   ISO/DIS 19111, Geographic Information – Spatial Referencing by Coordinates, 2000

[7]   Nicolai, Roel, ed., "The OpenGIS Abstract Specification, Topic 2: Spatial Referencing by Coordinates", 2002

[8]   Whiteside, A. (ed.), "Recommended XML encoding of Coordinate Reference System Definitions, OpenGIS Project document 02-036r3", August 2002

4   Terms and definitions

4.1
operation

specification of a transformation or query that an object may be called to execute [4]

4.2
interface

a named set of operations that characterise the behaviour of an entity [4]

4.3
service

a distinct part of the functionality that is provided by an entity through interfaces [4]

4.4
service instance

an actual implementation of a service; service instance is synonymous with server

4.5
client

a software component that can invoke an operation from a server

4.6
request

an invocation by a client of an operation.

4.7
response
the result of an operation returned from a server to a client.

4.8
capabilities
service-level metadata describing the operations and content available at a service instance

4.9
coordinate reference system
as defined in [7]

5 Conventions

5.1 Normative verbs

In the sections labeled as normative, the key words **"must", "must not", "required",
"shall", "shall not", "should", "should not", "recommended", "may",** and
optional" in this document are to be interpreted as described in Internet RFC 2119 [8].

5.2 Abbreviated terms

API   Application Program Interface

CGI   Common Gateway Interface

COTS   Commercial Off The Shelf

CRS   Coordinate Reference System

CTS   Coordinate Transformation Services

DCP   Distributed Computing Platform

EPSG   European Petroleum Survey Group

GIS   Geographic Information System

GML   Geography Markup Language

HTTP   Hypertext Transfer Protocol

ISO   International Organization for Standardization

    3

KVP	key-value pair
OGC	Open GIS Consortium
RFC	Request for Comments
URL	Uniform Resource Locator
WCTS	Web Coordinate Transformation Service
WMS	Web Map Service
WFS	Web Feature Service
WCS	Web Coverage Service
XML	eXtended Markup Language

6 Basic Service Elements

This section describes aspects of Web Coordinate Transformation Service behaviour that are independent of particular operations or are common to several operations or interfaces.

6.1 Version Numbering and Negotiation

6.1.1 Version Number Form

The published specification version number contains three positive integers, separated by decimal points, in the form "x.y.z". The numbers "y" and "z" will never exceed 99.

6.1.2 Version Changes

A particular specification's version number **shall** be changed with each revision. The number **shall** increase monotonically and **shall** comprise no more than three integers separated by decimal points, with the first integer being the most significant. There may be gaps in the numerical sequence. Some numbers may denote experimental or interim versions. Service instances and their clients need not support all defined versions, but **must** obey the negotiation rules below.

6.1.3 Appearance in Requests and in Service Metadata

The version number appears in at least two places: in the Capabilities XML describing a service, and in the parameter list of client requests to that service. The version number used in a client's request of a particular service instance **must** be equal to a version number which that instance has declared it supports (except during negotiation as

described below). A service instance may support several versions, whose values clients may discover according to the negotiation rules.

6.1.4   Version Number Negotiation

A Client may negotiate with a Service Instance to determine a mutually agreeable specification version. Negotiation is performed using the GetCapabilities operation according to the following rules.

All Capabilities XML must include a protocol version number. In response to a GetCapabilities request containing a version number, an OGC Web Service **must** either respond with output that conforms to that version of the specification, **or** negotiate a mutually agreeable version if the requested version is not implemented on the server. If no version number is specified in the request, the server **must** respond with the highest version it understands and label the response accordingly.

Version number negotiation occurs as follows:

1. If the server implements the requested version number, the server must send that version.

2. If the client request is for an unknown version greater than the lowest version that the server understands, the server must send the highest version less than the requested version.

3. If the client request is for a version lower than any of those known to the server, then the server must send the lowest version it knows.

4. If the client does not understand the new version number sent by the server, it may either cease communicating with the server or send a new request with a new version number that the client does understand but which is less than that sent by the server (if the server had responded with a lower version).

5. If the server had responded with a higher version (because the request was for a version lower than any known to the server), and the client does not understand the proposed higher version, then the client may send a new request with a version number higher than that sent by the server.

The process is repeated until a mutually understood version is reached, or until the client determines that it will not or cannot communicate with that particular server.

6.2   General HTTP Request Rules

6.2.1   Introduction

At present, the only distributed computing platform (DCP) explicitly supported by OGC Web Services is the World Wide Web itself, or more specifically Internet hosts implementing the Hypertext Transfer Protocol (HTTP)[2]. Thus the Online Resource of

each operation supported by a service instance is an HTTP Uniform Resource Locator (URL). The URL may be different for each operation, or the same, at the discretion of the service provider. Each URL **must** conform to the description in [2] but is otherwise implementation-dependent; only the parameters comprising the service request itself are mandated by the OGC Web Services specifications.

HTTP supports two request methods: GET and POST. One or both of these methods may be defined for a particular OGC Web Service type and offered by a service instance, and the use of the Online Resource URL differs in each case.

6.2.2   HTTP Get

An Online Resource URL intended for HTTP GET requests is in fact only a URL prefix to which additional parameters must be appended in order to construct a valid Operation request. A URL prefix is defined as an opaque string including the protocol, hostname, optional port number, path, a question mark '?', and, optionally, one or more server-specific

parameters ending in an ampersand '&'. The prefix uniquely identifies the particular service instance. A client appends the necessary request parameters as name/value pairs in the form "name=value&". The resulting URL must be valid according to the HTTP Common Gateway Interface (CGI) standard [7], which mandates the presence of '?' before the sequence of query parameters and the '&' between each parameter. As with all CGI applications, the query URL is encoded [8] to protect special characters.

The URL prefix must end in either a '?' (in the absence of additional server-specific parameters) or a '&'. In practice, however, Clients **should** be prepared to add a necessary trailing '?' or '&' before appending the Operation parameters defined in this specification in order to construct a valid request URL.

Table 1 summarizes the components of an operation request URL.

Table 1 A general OGC Web Service Request

URL Component	Description
http://host[:port]/path?{name[=value]&}	URL prefix of service operation. [ ] denotes 0 or 1 occurrence of an optional part; { } denotes 0 or more occurrences. The prefix is entirely at the discretion of the service provider.
name=value&	One or more standard request parameter name/value pairs defined by an OGC Web Service. The actual list of required and optional parameters is mandated for each operation by the appropriate OWS specification.

6.2.3   HTTP Post

An Online Resource URL intended for HTTP POST requests is a complete and valid URL to which Clients transmit encoded requests in the body of the POST document. A WCTS **must not** require additional parameters to be appended to the URL in order to construct a valid target for the Operation request.

6.3   General HTTP Response Rules

Upon receiving a valid request, the service **must** send a response corresponding exactly to the request as detailed in the appropriate specification. Only in the case of Version Negotiation (described above) may the server offer a differing result. Upon receiving an invalid request, the service **must** issue a Service Exception.

NOTE: As a practical matter, in the WWW environment a client should be prepared to receive either a valid result, or nothing, or any other result. This is because the client may itself have formed a non-conforming request that inadvertently triggered a reply by something other than an OGC Web Service, because the Service itself may be non conforming, etc.

Response objects **must** be accompanied by the appropriate Multipurpose Internet Mail Extensions (MIME) type [9] for that object.

Response objects **should** be accompanied by other HTTP entity headers as appropriate and to the extent possible. In particular, the Expires and Last-Modified headers provide important information for caching; Content-Length may be used by clients to know when data transmission is complete and to efficiently allocate space for results, and Content-Encoding or Content-Transfer-Encoding may be necessary for proper interpretation of the results.

6.4   Request Encoding

This document defines two methods of encoding WCTS requests. The first uses XML as the encoding language. The second method uses keyword-value pairs to encode the various parameters of a request. An example of a keyword value pair is:

```
"REQUEST=GetCapabilities"
```

where "REQUEST" is the keyword and "GETCAPABILITITES" is the va lue. In both cases, the response to a request or exception reporting must be identical. Table 2 correlates WCTS operations and their encoding semantics as defined in this specification.

Table 2 Operation Request Encoding

Operation	Request Encoding
GetCapabilities	XML & KVP
DescribeTransformation	XML & limited KVP
IsTransformable	XML & limited KVP
Transform	XML & limited KVP

The KVPs may include XML elements. This is for example the case when encoding a Transform request. The geo-spatial data that shall be transformed may be encoded as GML. If so the GML must be URL-encoded to uses it within a HTTP-GET request.

6.5   Namespaces

Namespaces [3] are used to distinguish the definitions of features and properties defined in application-specific domains from one another, and from the core constructs defined in the GML and WCTS modules. A WCTS must be able to ingest and properly discriminate XML input elements tagged with namespaces and must generate proper XML & GML, correctly tagged with namespaces, so that the output validates against the various schemas mandated by this document..

7   Common Elements

7.1   Exceptions

Upon receiving an invalid request, the service **must** issue a Service Exception XML message to describe to the client application or its human user the reason(s) that the request is invalid.

Service Exception XML **must** be valid according to the Service Exception Schema definition in Annex A.3. In an HTTP environment, the MIME type of the returned XML **must** be "`application/vnd.ogc.se_xml`".

The following Schema fragment shows the structure of an exception:

```
<!-- ======================================================= -->
<!--                    element definitions                  -->
<!-- ======================================================= -->
<xs:element name="Exception" type="ExceptionType"/>
<!-- ======================================================= -->
<!--                    type definitions                     -->
<!-- ======================================================= -->
<xs:complexType name="ExceptionType">
    <xs:sequence>
        <xs:element name="Message" type="xs:string"/>
        <xs:element name="Location" type="xs:string"/>
    </xs:sequence>
</xs:complexType>
```

7.2   XML elements and attributes

There are a number of elements and attributes that are used by several requests and responses. The full XML schema definition of these elements can be found in A.2. These schemas make use of GML 3 schemas for features, geometry, coordinate reference systems and coordinate operations.

The following elements use definitions of these external schemas and are part of the basic definitions:

```xml
<xs:element name="SourceCRS" type="wcts:CRSType"/>
<xs:element name="TargetCRS" type="wcts:CRSType"/>
<xs:element name="OnlineResource" type="wcts:OnlineResourceType"/>
<!-- ===================================================== -->
<!--                   type definitions                    -->
<!-- ===================================================== -->
    <!-- coordinate reference system of the data to be transformed -->
<xs:complexType name="CRSType">
    <xs:sequence>
        <xs:element ref="gml:_CoordinateReferenceSystem"/>
    </xs:sequence>
</xs:complexType>
<!-- ============== OnlineResourceType ================== -->
<xs:complexType name="OnlineResourceType">
    <xs:attribute name="xmlns" type="xs:string" fixed="http://www.w3.org/1999/xlink"/>
    <xs:attribute name="type" type="xs:string" fixed="simple"/>
    <xs:attribute name="href" type="xs:string"/>
</xs:complexType>
```

7.2.1   Geometry, Coverage and Interpolation typs.

The following schema fragment describes the geometric primitives, the coverage types and the interpolation methods a WCTS may support.

```xml
<!--Types of Geometric Primitives in GML 3-->
    <xs:simpleType name="GeometricPrimitiveTypeType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="PointType"/>
            <xs:enumeration value="LineStringType"/>
            <xs:enumeration value="PolygonType"/>
            <xs:enumeration value="LinearRingType"/>
            <xs:enumeration value="CurveType"/>
            <xs:enumeration value="LineStringSegmentType"/>
            <xs:enumeration value="ArcStringType"/>
            <xs:enumeration value="ArcType"/>
            <xs:enumeration value="CircleType"/>
            <xs:enumeration value="ArcStringByBulgeType"/>
            <xs:enumeration value="ArcByBulgeType"/>
            <xs:enumeration value="ArcByCenterPointType"/>
            <xs:enumeration value="CircleByCenterPointType"/>
            <xs:enumeration value="CubicSplineType"/>
            <xs:enumeration value="BSplineType"/>
            <xs:enumeration value="BezierType"/>
            <xs:enumeration value="OrientableCurveType"/>
            <xs:enumeration value="SurfaceType"/>
            <xs:enumeration value="PolygonPatchType"/>
            <xs:enumeration value="TriangleType"/>
            <xs:enumeration value="RectangleType"/>
            <xs:enumeration value="RingType"/>
```

```
                <xs:enumeration value="OrientableSurfaceType"/>
                <xs:enumeration value="SolidType"/>
                <xs:enumeration value="CompositeCurveType"/>
                <xs:enumeration value="CompositeSurfaceType"/>
                <xs:enumeration value="CompositeSolidType"/>
            </xs:restriction>
        </xs:simpleType>
        <!--Types of (referenced) Coverages in GML 3-->
        <xs:simpleType name="CoverageTypeType">
            <xs:restriction base="xs:string">
                <xs:enumeration value="MulitPointCoverageType"/>
                <xs:enumeration value="MulitSurfaceCoverageType"/>
                <xs:enumeration value="RectifiedGridCoverageType"/>
            </xs:restriction>
        </xs:simpleType>
        <!-- methods for interpolating coverages after they have been transformed-->
        <xs:simpleType name="CoverageInterpolationMethodType">
            <xs:restriction base="xs:string">
                <xs:enumeration value="nearest neighbor"/>
                <xs:enumeration value="linear"/>
                <xs:enumeration value="bilinear"/>
                <xs:enumeration value="bicubic"/>
                <xs:enumeration value="lost area"/>
                <xs:enumeration value="barycentric"/>
                <xs:enumeration value="piecewise constant"/>
                <xs:enumeration value="none"/>
            </xs:restriction>
        </xs:simpleType>
```

8 GetCapabilities Operation (required)

8.1 Introduction

The web coordinate transformation service must have the ability to describe its capabilities. This section defines an XML document that a web coordinate transformation server must generate to define its capabilities.

The capabilities document defined in this specification is closely modeled after the capabilities document defined for web map servers and web feature servers as described in the Web Map Service Implementation Specification [9] and the Web Feature Service Implementation Specification [10].

8.2 Request

8.2.1 Key-Value pair encoding

The general form of a **GetCapabilities** request is summarized in Table 1 below.

Table 3 — The parameters of a GetCapabilities request URL

Request Parameter	Required/ Optional	Description
REQUEST=GetCapabilities	R	Request name
VERSION=version	O	Request version
SERVICE=WCTS	R	Service type

When making this request to a WCTS server, which may offer other OGC Web Services as well, it is necessary to indicate that the client seeks information about the WCTS server in particular. Thus, the SERVICE parameter of the request **must** have the value "WCTS" as shown in Table 3.

8.2.2   XML encoding

The **<GetCapabilities>** element is used to request a capabilities document from a web coordinate transformation server.

```
<!-- ======================================================= -->
<!--            element definition           -->
<!-- ======================================================= -->
<!-- root element of a transform request -->
<xs:element name="GetCapabilities" type="wcts:GetCapabilitiesType"/>
<!-- ======================================================= -->
<!--            type definitions            -->
<!-- ======================================================= -->
<xs:complexType name="GetCapabilitiesType">
    <xs:attribute name="version" type="xs:string" use="optional" fixed="0.1.0"/>
    <xs:attribute name="service" type="xs:string" use="optional" fixed="WCTS"/>
</xs:complexType>
```

The top-level XML element, **GetCapabilities**, has two attributes, **version** and **service**, denoting, respectively, the version number of the protocol and the service it addresses.

8.3   Response

8.3.1   Response Schema

The schema of the response to a **GetCapabilities** request is normatively defined using XML Schema in Annex A.5.

8.3.2   Capabilities Document

The capabilities document is composed of two sections:

1. Service Section

> The service section provides information about the service itself.

2. Capabilities Section

> The capabilities section specifies the list of requests that the WCTS can handle. Besides the mandatory operations **GetCapabilities**, **IsTransformable** and **Transform** the optional operation **DescribeTransformation** can be described in the **<Request>** section.

Any WCTS has to know at least one coordinate transformation and two coordinate reference systems so that it is able to perform one operation. The known transformations and reference systems are defined in **<KnownTransformation>** and **<KnownCoordinateReferenceSystem>**. Transformations refer to general methods for

transformation of coordinates from one reference system into another and not the precise transformations between two particular reference systems. Examples for transformtion types are: longitude rotation, abridged Molodenski, geocentric to ellipsoid etc.

As GML3 defines a vide spectrum of geometries it is not mandatory for a WCTS to support all those different kinds of geometries. Many WCTS will only need to support "simple" geometries as they were defined in GML 2. Others will not want to support all of the possible curve and surface interpolations available in GML 3 or are not interested in 3D geometries. The **<SupportedGeometricPromitiveTypes>** element allows to specify the geometric primites the WCTS is able to support. It is assumed, that a WCTS has the ability to work with the corresponding geometric complexes and aggregates. The **<SupportedCoverageTypeElement>** describes the types of GML 3 coverages the WCTS is able to handle. In the case of coverages which use specific geometric primitives, a client would have to check if the corresponding primitive types and coverage types are supported. The **<SupportedCoverageInterpolation>** elements give information about the interpolation methods the WCTS is able to handle in case an interpolation is necessary (as for transformation of rectified grids).

8.4   Exceptions

In the event that the web coordinate transformation server encounters an error servicing a **GetCapabilities** request, it shall raise an exception as described in Section 7.1 above.

8.5   Examples

This example shows what a capabilities document might look like for a basic web coordinate transformation server. To request a capabilities document, a client would issue the following request:

```
<GetCapabilities version="0.1.0" service="WCTS"/>
```

In response to such a request, the web coordinate transformation server might generate a document that looks like:

```xml
<?xml version="1.0" encoding="iso-8859-1"?>
<WCTS_Capabilities xmlns="http://schemas.opengis.net/wcts" xmlns:wcts="http://schemas.opengis.net/wcts"
xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.opengis.net/wcts
./GetCapabilitiesResponse.xsd" version="0.1.0" updateSequence="1">
    <Service>
        <Name>WCTS</Name>
        <Title>Web Coordinate Transformation Service</Title>
        <Abstract>Service for transformaing coordinates from one CRS to another through a network</Abstract>
        <KeywordList>
            <Keyword>Coordinate Reference System</Keyword>
            <Keyword>transformation</Keyword>
        </KeywordList>
        <OnlineResource/>
        <ContactInformation>
            <ContactPersonPrimary>
                <ContactPerson>Andreas Poth</ContactPerson>
                <ContactOrganization>lat/lon</ContactOrganization>
```

```xml
          </ContactPersonPrimary>
          <ContactPosition>Senior Software Engineer</ContactPosition>
          <ContactAddress>
              <AddressType>postal</AddressType>
              <Address>Meckenheimer Allee 176</Address>
              <City>Bonn</City>
              <StateOrProvince>NRW</StateOrProvince>
              <PostCode>53115</PostCode>
              <Country>Germany</Country>
          </ContactAddress>
          <ContactVoiceTelephone>++49 228 732838</ContactVoiceTelephone>
          <ContactFacsimileTelephone>++49 228 732153</ContactFacsimileTelephone>
          <ContactElectronicMailAddress>poth@lat-lon.de</ContactElectronicMailAddress>
      </ContactInformation>
      <Fees>none</Fees>
      <AccessConstraints>none</AccessConstraints>
  </Service>
  <Capability userDefinedCoordinateSystems="false" userDefinedTransformations="false">
      <Request>
          <GetCapabilities>
              <Format>String</Format>
              <DCPType>
                  <HTTP>
                      <Get>
                          <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="ww.lat-lon.de/transform?"/>
                      </Get>
                      <Post>
                          <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="ww.lat-lon.de/transform"/>
                      </Post>
                  </HTTP>
              </DCPType>
          </GetCapabilities>
          <Transform>
              <Format>String</Format>
              <DCPType>
                  <HTTP>
                      <Post>
                          <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="ww.lat-lon.de/transform"/>
                      </Post>
                  </HTTP>
              </DCPType>
          </Transform>
          <IsTransformable>
              <Format>String</Format>
              <DCPType>
                  <HTTP>
                      <Post>
                          <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="ww.lat-lon.de/transform"/>
                      </Post>
                  </HTTP>
              </DCPType>
          </IsTransformable>
          <DescribeTransformation>
              <Format>String</Format>
              <DCPType>
                  <HTTP>
                      <Post>
                          <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple"
xlink:href="ww.lat-lon.de/transform"/>
                      </Post>
                  </HTTP>
```

```xml
                    </DCPType>
                </DescribeTransformation>
            </Request>
            <SupportedTransformation>
                <coordinateTransformationID>
                    <gml:code>9601</gml:code>
                    <gml:codeSpace>EPSG</gml:codeSpace>
                </coordinateTransformationID>
            </SupportedTransformation>
            <SupportedTransformation>
                <coordinateTransformationID>
                    <gml:code>9602</gml:code>
                    <gml:codeSpace>EPSG</gml:codeSpace>
                </coordinateTransformationID>
            </SupportedTransformation>
            <SupportedTransformation>
                <coordinateTransformationID>
                    <gml:code>9603</gml:code>
                    <gml:codeSpace>EPSG</gml:codeSpace>
                </coordinateTransformationID>
            </SupportedTransformation>
            <SupportedTransformation>
                <coordinateTransformationID>
                    <gml:code>9604</gml:code>
                    <gml:codeSpace>EPSG</gml:codeSpace>
                </coordinateTransformationID>
            </SupportedTransformation>
            <SupportedTransformation>
                <coordinateTransformationID>
                    <gml:code>9807</gml:code>
                    <gml:codeSpace>EPSG</gml:codeSpace>
                </coordinateTransformationID>
            </SupportedTransformation>
            <SupportedCoordinateReferenceSystem>
                <crsID>
                    <gml:code>4326</gml:code>
                    <gml:codeSpace>EPSG</gml:codeSpace>
                </crsID>
            </SupportedCoordinateReferenceSystem>
            <SupportedCoordinateReferenceSystem>
                <crsID>
                    <gml:code>23031</gml:code>
                    <gml:codeSpace>EPSG</gml:codeSpace>
                </crsID>
            </SupportedCoordinateReferenceSystem>
            <SupportedCoordinateReferenceSystem>
                <crsID>
                    <gml:code>23032</gml:code>
                    <gml:codeSpace>EPSG</gml:codeSpace>
                </crsID>
            </SupportedCoordinateReferenceSystem>
            <SupportedCoordinateReferenceSystem>
                <crsID>
                    <gml:code>31467</gml:code>
                    <gml:codeSpace>EPSG</gml:codeSpace>
                </crsID>
            </SupportedCoordinat eReferenceSystem>
            <SupportedGeometricPrimitiveType>PointType</SupportedGeometricPrimitiveType>
            <SupportedGeometricPrimitiveType>LineStringType</SupportedGeometricPrimitiveType>
            <SupportedGeometricPrimitiveType>PolygonType</SupportedGeometricPrimitiveType>
        </Capability>
</WCTS_Capabilities>
```

9   IsTransformable Operation

9.1   Introduction

The **IsTransformable** request allows clients of the WCTS to check if a transformation of as specific set of geomtries between two coordinate references systems is possible. It only will be checked if the provided geometries are supported and if there is a valid way (sequence of transformation steps) to transform the coordinates of one CRS to those of another one. It will not be checked if this transformation makes any sense nor if it is possible to be performed for each specific coordinate expression that may be submitted by a Transform request.

In principle this information could be extracted from the capabilities of the WCTS as the elements **<SupportedGeometricPromitiveTypes>, <SupportedCoverageType>, <SupportedCoverageInteroplationType>, <KnownTransformation>** and **<KnownCoordinateReferenceSystem>** provide the necessary information. The analysis of these elements in regard to the possibility of a transformation between two reference systems for some specific geometries or coverages is in some cases rather complicated and would need powerful client logic. It seems to make more sense to provide this information directly via a distinct operation, allowing simple clients to use the service.

9.2   Request

9.2.1   Key-Value pair encoding

Table 4 — The parameters of an IsTransformable request URL

Request Parameter	Required/ Optional	Description
REQUEST=IsTransformable	R	Request name
VERSION=version	O	Request version
SOURCECRS	R	unique Identifier of the source spatial reference system
DESTINATIONCRS	R	unique Identifier of the destination spatial reference system
GEOMETRICPRIMITIVETYPE	O	list of types of geometric primitives that will be requested to be transformed
COVERAGETYPE	O	list of coverage types that will be requested to be transformed
COVERAGEINTERPOLATIONMETHOD	O	interpolation Method which should be used

9.2.2   XML encoding

```
<!-- ======================================================= -->
<!--                element definitions                      -->
```

```
<!-- ====================================================== -->
<!-- root element of an IsTtransformable request -->
<xs:element name="IsTransformable" type="wcts:IsTransformableType"/>
<!-- sub elements -->
<xs:element name="GeometricPrimitiveType" type="wcts:GeometricPrimitiveTypeType"/>
<xs:element name="CoverageInterpolationMethod" type="wcts:CoverageInterpolationMethodType"/>
<xs:element name="CoverageType" type="wcts:CoverageTypeType"/>
<!-- ====================================================== -->
<!--                    type definitions                    -->
<!-- ====================================================== -->
<xs:complexType name="IsTransformableType">
    <xs:sequence>
        <xs:element ref="wcts:SourceCRS"/>
        <xs:element ref="wcts:TargetCRS"/>
        <xs:element ref="wcts:GeometricPrimitiveType" minOccurs="0" maxOccurs="unbounded "/>
        <xs:element ref="wcts:CoverageType" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element ref="wcts:CoverageInterpolationMethod" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" fixed="0.1.0"/>
</xs:complexType>
```

9.3   Response

The response to an **IsTransformable** request is a simple 'true' or 'false' depending on whether the WCTS can perform the requested transformation.

```
<xs:element name="TransformableResponse" type="wcts:TransformableResponseType"/>
<xs:complexType name="TransformableResponseType">
    <xs:attribute name="transformable" type="xs:boolean" use="required"/>
</xs:complexType>
```

9.4   Exceptions

In the event that the web coordinate transformation server encounters an error servicing a **IsTransformable** request, it shall raise an exception as described in Section 7.1 above.

9.5   Examples

9.5.1   IsTransformable Request

An IsTransformable request for evaluating if a transformation between EPSG:4326 and EPSG:23032 is possible will look like this:

```
<IsTransformable xmlns="http://schemas.opengis.net/wcts" xmlns:xsi="http://www.w3.org/2001/XMLSchema -
instance" xmlns:gml="http://www.opengis.net/gml" xsi:schemaLocation="http://schemas.opengis.net/wcts
./IsTransformableRequest.xsd">
    <SourceCRS>
        <crsID>
            <gml:code>4326</gml:code>
            <gml:codeSpace>EPSG</gml:codeSpace>
        </crsID>
    </SourceCRS>
    <TargetCRS>
        <crsID>
            <gml:code>23032</gml:code>
            <gml:codeSpace>EPSG</gml:codeSpace>
        </crsID>
```

```
    </TargetCRS>
</IsTransformable>
```

9.5.2   IsTransformable Response

The servers response for the case that the transformation is possible looks like this:

```
<IsTransformableResponse isTransformable="true"/>
```

10   Transform Operation (required)

10.1   Introduction

The **Transform** operation is central to the web coordinate transformation server. It allows the transformation of coordinates from one coordinate reference system into another. Input data has to be provided as GML 3 Features, either in-line in the request, as a link to a GML file or as a Request to a service providing the data. This can either be a WFS or a WCS. Optionally the WCTS can support user-defined transformations which make it necessary for the client to provide the parameters for the transformation sequence.

10.2   Request

10.2.1   Key-Value pair encoding

Table 5 — The parameters of a Transform request URL

Request Parameter	Required/ Optional	Description
REQUEST=Transform	R	Request name
VERSION=version	O	Request version
SOURCECRS	R	unique Identifier of the source spatial reference system
TARGETCRS	R	unique Identifier of the target spatial reference system
TRANSFORMATIONSEQUENCE	O	Sequence of transformation steps. Each step has to be described in XML
DATA	R	The data that has to be transformed in GML 3 or as reference / WFS-request

Using a Key-Value pair encoding of a **Transform** request the number of the geometries submitted with one request is limited to one if supplied in-line.

10.2.2 XML encoding

The following schema fragment shows the structure of an XML encoded **Transform** request:

```xml
<!-- ======================================================= -->
<!--                        element definitions                        -->
<!-- ======================================================= -->
<!-- root element of an IsTransformable request -->
<xs:element name="Transform" type="wcts:TransformType"/>
<!-- sub elements -->
<xs:element name="Data" type="wcts:DataType"/>
<xs:element name="RemoteOWS" type="wcts:RemoteOWSType "/>
<xs:element name="OWSRequest" type="wcts:OWSRequestType"/>
<xs:element name="Service" type ="wcts:ServiceType"/>
<xs:element name="KVPRequest" type ="xs:string"/>
<xs:element name="XMLRequest" type ="xs:string"/>
<!-- ======================================================= -->
<!--                        type definitions                        -->
<!-- ======================================================= -->
<xs:complexType name="TransformType">
    <xs:sequence>
        <xs:element ref ="wcts:SourceCRS"/>
        <xs:element ref ="wcts:TargetCRS"/>
        <xs:element ref ="gml:_CoordinateOperation" minOccurs="0"/>
        <xs:element ref ="wcts:Data" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="version" type ="xs:string" use ="required" fixed="0.1.0"/>
</xs:complexType>
<xs:complexType name="DataType">
    <xs:choice>
        <xs:element name ="FileURL"/>
        <xs:element ref ="gml:_Feature"/>
        <xs:element ref ="wcts:RemoteOWS"/>
    </xs:choice>
</xs:complexType>
<xs:complexType name="RemoteOWSType">
    <xs:annotation>
        <xs:documentation >
        A RemoteOWS gives a reference to a remote WFS/WCS/other-OWS server.
        </xs:documentation>
    </xs:annotation>
    <xs:sequence>
        <xs:element ref ="wcts:OnlineResource"/>
        <xs:element ref ="wcts:OWSRequest"/>
    </xs:sequence>
</xs:complexType>
<xs:complexType name="OWSRequestType">
    <xs:choice>
        <xs:element ref ="wcts:KVPRequest"/>
        <xs:element ref ="wcts:XMLRequest"/>
    </xs:choice>
</xs:complexType>
<xs:simpleType name="ServiceType">
    <xs:restriction base="xs:string">
        <xs:enumeration value="WFS"/>
        <xs:enumeration value="WCS"/>
    </xs:restriction>
</xs:simpleType>
```

10.3 Response

The response to a **Transform** request is a GML 3 Feature.

```
<!-- ======================================================= -->
<!--                  element definitions                    -->
<!-- ======================================================= -->
<!-- root element of a transform response -->
<xs:element name="TransformResponse" type="wcts:TransformResponseType"/>
<!-- ======================================================= == -->
<!--                   type definitions                      -->
<!-- ======================================================= -->
<xs:complexType name="TransformResponseType">
    <xs:sequence>
        <xs:element ref="gml:_Feature" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

10.4 Exceptions

In the event that the web coordinate transformation server encounters an error servicing a **Transform** request, it shall raise an exception as described in Section 7.1 above.

10.5 Examples

10.5.1 Transform request

If a client wants to transform a GML FeatureCollection from EPSG:2303 to EPSG:31467 using the default transformation steps it has to construct the following request:

```
<Transform xmlns ="http://schemas.opengis.net/wcts" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:gml="http://www.opengis.net/gml" version="0.1.0">
    <SourceCRS>
        <crsID>
            <gml:code>23032</gml:code>
            <gml:codeSpace>EPSG</gml:codeSpace>
        </crsID>
    </SourceCRS>
    <TargetCRS>
        <crsID>
            <gml:code>31467</gml:code>
            <gml:codeSpace>EPSG</gml:codeSpace>
        </crsID>
    </TargetCRS>
    <Data>
        <!-- here would be the description of a GML 3 FeatureCollection in EPSG: 23032 ...-->
    </Data>
</Transform>
```

10.5.2 Transform response

As response to a request for transforming a LineString from EPSG:2303 to EPSG:31467 the server will response – if no exception occurs – with:

```
<?xml version="1.0" encoding="UTF-8"?>
<TransformResponse xmlns:gml="http://www.opengis.net/gml" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" >
    <Data>
        <!-- ...and here would be the transformed FeatureCollection in EPSG:31467-->
    </Data>
</TransformResponse>
```

11   DescribeTransformation Operation (optional)

11.1   Introduction

Support of the **DescribeTransformation** request is optional for a web coordinate transformation server. It offers the description of a transformation from one coordinate reference system into another. The WCTS DescribeTransformation request supplies the steps which are necessary for the transformation. The response will be encoded as XML.

11.2   Request

11.2.1   Key-Value pair encoding

Table 6 — The parameters of a DescribeTransformation request URL

Request Parameter	Required/ Optional	Description
REQUEST= DescribeTransformation	R	Request name
VERSION=version	O	Request version
SOURCECRS	R	unique Identifier of the source spatial reference system
TARGETCRS	R	unique Identifier of the destination spatial reference system

11.2.2   XML encoding

```
<!-- ======================================================= -->
<!--                    element definitions                  -->
<!-- ======================================================= -->
<!-- root element of a transform request -->
<xs:element name="DescribeTransformation" type="wcts:DescribeTransformationType"/>
<!-- sub elements -->
<!-- ======================================================= -->
<!--                      type definitions                   -->
<!-- === =================================================== -->
<xs:complexType name="DescribeTransformationType">
    <xs:sequence>
        <xs:element ref="wcts:SourceCRS"/>
        <xs:element ref="wcts:TargetCRS"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" fixed="0.1.0"/>
</xs:complexType>
```

11.3   Response

As the transformation from one coordinate reference system to another cannnot usually be done by a simple transformation, the transformation is described by its transformation steps. One transformation step corresponds to a **Transformation** element as defined in [8]. The number of transformation steps is provided by the **numberOfTransformations** attribute.

It is possible that a transformation is not feasable. Possible reasons for this are that the web coordinat transformation service does not know the necessary transformation steps or that a transformation is not possible at all (for example the transformation of a 3D to a 2D system). If a transformation is not feasable for a service WCTS returns an Exception.

A schema fragment describing the response to a **DescribeTransformation** request follows:

```
<!-- ========================================================= -->
<!--                     element definitions                   -->
<!-- ========================================================= -->
<!-- root element of a transform request -->
<xs:element name="DescribeTransformationResponse" type="wcts:DescribeTransformationResponseType"/>
<xs:element name="CoordinateOperationID" type="gml:ExtendedIdentifierType"/>
<!-- ========================================================= -->
<!--                      type definitions                     -->
<!-- ========================================================= -->
<!-- response type -->
<xs:complexType name="DescribeTransformationResponseType">
    <xs:sequence>
        <xs:element ref="wcts:CoordinateOperationID" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

11.4 Exceptions

In the event that the web coordinate transformation server encounters an error servicing a **DescribeTransformation** request, it shall raise an exception as described in Section 7.1 above.

11.5 Examples

11.5.1 DescribeTransformation request

A request for the description of a transformation might be encoded like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<DescribeTransformation xmlns="http://schemas.opengis.net/wcts "
xmlns:wcts ="http://schemas.opengis.net/wcts ">
    <SourceCRS>
        <crsID>
            <gml:code>4326</gml:code>
            <gml:codeSpace>EPSG</gml:codeSpace>
        </crsID>
    </SourceCRS>
    <TargetCRS>
        <crsID>
            <gml:code>31467</gml:code>
            <gml:codeSpace>EPSG</gml:codeSpace>
        </crsID>
    </TargetCRS>
</DescribeTransformation>
```

11.5.2 DescribeTransformation response

If no exception occurs the server will reply to a DescribeTransformation request with a list of transformations. To transform EPSG:4326 to EPSG:31467 just one transformation step is required.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<DescribeTransformationResponse xmlns="http://schemas.opengis.net/wcts"
xmlns:wcts ="http://schemas.opengis.net/wcts " xmlns:gml="http://www.opengis.net/gml" >
    <CoordinateOperationID>
        <gml:code>9807</gml:code>
        <gml:codeSpace>EPSG</gml:codeSpace>
    </CoordinateOperationID>
</DescribeTransformationResponse>
```

Annex A
(normative)

XML Schema definitions

A.1 Introduction

A.2 Basic Definitions

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/wcts "
xmlns:xs="http://www.w3.org/2001/XMLSchema " xmlns:gml="http://www.opengis.net/gml"
xmlns:wcts="http://schemas.opengis.net/wcts " xmlns:xlink="http://www.w3.org/1999/xlink"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:import schemaLocation="../gml3.0/base/coordinateReferenceSystems.xsd"/>
    <xs:import schemaLocation="../gml3.0/base/geometry.xsd"/>
    <xs:import schemaLocation="../gml3.0/xlink/xlinks.xsd"/>
    <!-- ========================================================= -->
    <!--                        element definitions                     -->
    <!-- ========================================================= -->
    <xs:element name="SourceCRS" type="wcts:CRSType"/>
    <xs:element name="TargetCRS" type="wcts:CRSType"/>
    <xs:element name="OnlineResource" type="wcts:OnlineResourceType"/>
    <!-- ========================================================= -->
    <!--                         type definitions                       -->
    <!-- ========================================================= -->
    <!-- defines the type for input and output format of a transform request. -->
    <xs:complexType name="FormatType">
        <xs:attribute name="name">
            <xs:simpleType>
                <xs:restriction base="xs:NMTOKEN">
                    <xs:enumeration value="GML"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
    <!-- coordinate reference system of the data to be transformed -->
    <xs:complexType name="CRSType">
        <xs:sequence>
            <xs:element name="crsID" type="gml:ExtendedIdentifierType"/>
        </xs:sequence>
    </xs:complexType>
    <!-- ============= OnlineResourceType ================== -->
    <xs:complexType name="OnlineResourceType">
        <xs:attributeGroup ref="xlink:simpleLink"/>
    </xs:complexType>
    <!--Types of Geometric Primitives in GML 3-->
    <xs:simpleType name="GeometricPrimitiveTypeType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="PointType"/>
            <xs:enumeration value="LineStringType"/>
            <xs:enumeration value="PolygonType"/>
            <xs:enumeration value="LinearRingType"/>
            <xs:enumeration value="CurveType"/>
            <xs:enumeration value="LineStringSegmentType"/>
            <xs:enumeration value="ArcStringType"/>
```

```
                    <xs:enumeration value="ArcType"/>
                    <xs:enumeration value="CircleType"/>
                    <xs:enumeration value="ArcStringByBulgeType"/>
                    <xs:enumeration value="ArcByBulgeType"/>
                    <xs:enumeration value="ArcByCenterPointType"/>
                    <xs:enumeration value="CircleByCenterPointType"/>
                    <xs:enumeration value="CubicSplineType"/>
                    <xs:enumeration value="BSplineType"/>
                    <xs:enumeration value="BezierType"/>
                    <xs:enumeration value="OrientableCurveType"/>
                    <xs:enumeration value="SurfaceType"/>
                    <xs:enumeration value="PolygonPatchType"/>
                    <xs:enumeration value="TriangleType"/>
                    <xs:enumeration value="RectangleType"/>
                    <xs:enumeration value="RingType"/>
                    <xs:enumeration value="OrientableSurfaceType"/>
                    <xs:enumeration value="SolidType"/>
                    <xs:enumeration value="CompositeCurveType"/>
                    <xs:enumeration value="CompositeSurfaceType"/>
                    <xs:enumeration value="CompositeSolidType"/>
                </xs:restriction>
            </xs:simpleType>
        <!--Types of (referenced) Coverages in GML 3-->
        <xs:simpleType name="CoverageTypeType">
            <xs:restriction base="xs:string">
                <xs:enumeration value="MulitPointCoverageType"/>
                <xs:enumeration value="MulitSurfaceCoverageType"/>
                <xs:enumeration value="RectifiedGridCoverageType"/>
            </xs:restriction>
        </xs:simpleType>
        <!-- methods for interpolating coverages after they have been transformed-->
        <xs:simpleType name="CoverageInterpolationMethodType">
            <xs:restriction base="xs:string">
                <xs:enumeration value="nearest neighbor"/>
                <xs:enumeration value="linear"/>
                <xs:enumeration value="bilinear"/>
                <xs:enumeration value="bicubic"/>
                <xs:enumeration value="lost area"/>
                <xs:enumeration value="barycentric"/>
                <xs:enumeration value="piecewise constant"/>
                <xs:enumeration value="none"/>
            </xs:restriction>
        </xs:simpleType>
</xs:schema>
```

A.3 Exception (Response)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <!-- ======================================================== -->
    <!--                     element definitions                  -->
    <!-- ======================================================== -->
    <xs:element name="Exception" type="ExceptionType"/>
    <!-- ======================================================== -->
    <!--                      type definitions                    -->
    <!-- ======================================================== -->
    <xs:complexType name="ExceptionType">
        <xs:sequence>
            <xs:element name="Message" type="xs:string"/>
            <xs:element name="Location" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

OGC 02-061r2

A.4 GetCapabilities Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/wcts "
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wcts="http://schemas.opengis.net/wcts"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <!-- ===================================================== -->
    <!--              element definition              -->
    <!-- ===================================================== -->
    <!-- root element of a transform request -->
    <xs:element name="GetCapabilities" type="wcts:GetCapabilitiesType"/>
    <!-- ===================================================== -->
    <!--              type definitions               -->
    <!-- ===================================================== -->
    <xs:complexType name="GetCapabilitiesType">
        <xs:attribute name="version" type="xs:string" use="optional" fixed="0.1.0"/>
        <xs:attribute name="service" type="xs:string" use="optional" fixed="WCTS"/>
    </xs:complexType>
</xs:schema >
```

A.5 Capabilities (Response to GetCapabilities request)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/wcts "
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
xmlns:wcts ="http://schemas.opengis.net/wcts " elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:import schemaLocation ="WCTSCommon.xsd"/>
    <xs:import schemaLocation ="../gml3.0/base/geometry.xsd"/>
    <!-- ======================================================== -->
    <!--                    element definitions                   -->
    <!-- ======================================================== -->
    <!-- root element of a Capabilities response -->
    <xs:element name="WCTS_Capabilities " type="wcts:WCTS_CapabilitiesType"/>
    <!-- sub elements -->
    <xs:element name="Service" type="wcts:ServiceType"/>
    <xs:element name="Capability" type="wcts:CapabilityType "/>
    <xs:element name="Name" type="xs:string"/>
    <xs:element name="Title" type="xs:string"/>
    <xs:element name="Abstract" type="xs:string"/>
    <xs:element name="Format" type="xs:string"/>
    <xs:element name="DCPType" type="wcts:DCPTypeType"/>
    <xs:element name="HTTP" type="wcts:HTTPType"/>
    <xs:element name="KeywordList" type="wcts:KeywordListType"/>
    <xs:element name="Keyword" type="xs:string"/>
    <xs:element name="ContactInformation" type="wcts:ContactInformationType"/>
    <xs:element name="ContactPersonPrimary" type="wcts:ContactPersonPrimaryType "/>
    <xs:element name="ContactPerson" type="xs:string"/>
    <xs:element name="ContactOrganization" type="xs:string"/>
    <xs:element name="ContactPosition" type="xs:string "/>
    <xs:element name="ContactAddress" type="wcts:ContactAddressType"/>
    <xs:element name="AddressType" type="xs:string"/>
    <xs:element name="Address" type="xs:string"/>
    <xs:element name="City" type="xs:string"/>
    <xs:element name="StateOrProvince" type="xs:string"/>
    <xs:element name="PostCode" type="xs:string"/>
    <xs:element name="Country" type="xs:string"/>
    <xs:element name="ContactVoiceTelephone" type="xs:string"/>
    <xs:element name="ContactFacsimileTelephone" type="xs:string"/>
    <xs:element name="ContactElectronicMailAddress" type="xs:string"/>
    <xs:element name="Fees" type="xs:string"/>
    <xs:element name="AccessConstraints " type="xs:string"/>
    <xs:element name="Request" type="wcts:RequestType"/>
    <xs:element name="SupportedTransformation " type="wcts:SupportedTransformationType"/>
    <xs:element name="SupportedCoordinateReferenceSystem" type ="wcts:CRSType"/>
    <xs:element name="SupportedGeometricPrimitiveType" type="wcts:GeometricPrimitiveTypeType "/>
    <xs:element name="SupportedCoverageType " type="wcts:CoverageTypeType"/>
    <xs:element name="SupportedCoverageInterpolationMethod"
type="wcts:CoverageInterpolationMethodType"/>
    <!-- ======================================================== -->
    <!--                    type definitions                      -->
    <!-- ======================================================== -->
    <xs:complexType name="WCTS_CapabilitiesType">
        <xs:sequence>
            <xs:element ref="wcts:Service"/>
            <xs:element ref="wcts:Capability "/>
        </xs:sequence>
        <xs:attribute name="version" type="xs:string"/>
        <xs:attribute name="updateSequence" type="xs:string"/>
    </xs:complexType>
    <!-- =============== Service Type ================== -->
    <xs:complexType name="ServiceType">
```

```
            <xs:sequence>
                <xs:element ref="wcts:Name"/>
                <xs:element ref="wcts:Title"/>
                <xs:element ref="wcts:Abstract" minOccurs="0"/>
                <xs:element ref="wcts:KeywordList" minOccurs="0"/>
                <xs:element ref="wcts:OnlineResource"/>
                <xs:element ref="wcts:ContactInformation"/>
                <xs:element ref="wcts:Fees" minOccurs="0"/>
                <xs:element ref="wcts:AccessConstraints" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
        <!-- ============== KeywordListType ================= -->
        <xs:complexType name="KeywordListType">
            <xs:sequence>
                <xs:element ref="wcts:Keyword" minOccurs="0" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
        <!-- ============== ContactInformationType ================= -->
        <xs:complexType name="ContactInformationType">
            <xs:sequence>
                <xs:element ref="wcts:ContactPersonPrimary" minOccurs="0"/>
                <xs:element ref="wcts:ContactPosition" minOccurs="0"/>
                <xs:element ref="wcts:ContactAddress" minOccurs="0"/>
                <xs:element ref="wcts:ContactVoiceTelephone" minOccurs="0"/>
                <xs:element ref="wcts:ContactFacsimileTelephone" minOccurs="0"/>
                <xs:element ref="wcts:ContactElectronicMailAddress" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
        <!-- ============== ContactPersonPrimaryType ================= -->
        <xs:complexType name="ContactPersonPrimaryType">
            <xs:sequence>
                <xs:element ref="wcts:ContactPerson"/>
                <xs:element ref="wcts:ContactOrganization"/>
            </xs:sequence>
        </xs:complexType>
        <!-- ============== ContactAddressType ================= -->
        <xs:complexType name="ContactAddressType">
            <xs:sequence>
                <xs:element ref="wcts:AddressType"/>
                <xs:element ref="wcts:Address"/>
                <xs:element ref="wcts:City"/>
                <xs:element ref="wcts:StateOrProvince"/>
                <xs:element ref="wcts:PostCode"/>
                <xs:element ref="wcts:Country"/>
            </xs:sequence>
        </xs:complexType>
        <!-- ============== Capability Type ================= -->
        <xs:complexType name="CapabilityType">
            <xs:sequence>
                <xs:element ref="wcts:Request"/>
                <xs:element ref="wcts:SupportedTransformation" maxOccurs="unbounded"/>
                <xs:element ref="wcts:SupportedCoordinateReferenceSystem" minOccurs="2"
maxOccurs="unbounded"/>
                <xs:element ref="wcts:SupportedGeometricPrimitiveType" maxOccurs="unbounded"/>
                <xs:element ref="wcts:SupportedCoverageType" minOccurs="0" maxOccurs="unbounded"/>
                <xs:element ref="wcts:SupportedCoverageInterpolationMethod" minOccurs="0"
maxOccurs="unbounded"/>
                <xs:element name="VendorSpecificCapabilities" type="xs:string" minOccurs="0"
maxOccurs="unbounded"/>
            </xs:sequence>
            <xs:attribute name="userDefinedCoordinateSystems" type="xs:boolean" default="false"/>
            <xs:attribute name="userDefinedTransformations" type="xs:boolean" default="false"/>
        </xs:complexType>
        <!-- ============== RequestType ================= -->
        <xs:complexType name="RequestType">
```

```xml
            <xs:sequence>
                <xs:element name="GetCapabilities" type="wcts:ActionType"/>
                <xs:element name="Transform" type="wcts:ActionType"/>
                <xs:element name="IsTransformable" type="wcts:ActionType"/>
                <xs:element name="DescribeTransformation" type="wcts:ActionType" minOccurs="0"/>
            </xs:sequence>
        </xs:complexType>
        <!-- ============== ActionType ================= -->
        <xs:complexType name="ActionType">
            <xs:sequence>
                <xs:element ref="wcts:Format" maxOccurs="unbounded"/>
                <xs:element ref="wcts:DCPType" maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
        <!-- ============== DCPTypeType ================= -->
        <xs:complexType name="DCPTypeType">
            <xs:choice>
                <xs:element ref="wcts:HTTP"/>
            </xs:choice>
        </xs:complexType>
        <!-- ============== HTTPType ================= -->
        <xs:complexType name="HTTPType">
            <xs:choice maxOccurs="unbounded">
                <xs:element name="Post" type="wcts:AccessMethodType"/>
                <xs:element name="Get" type="wcts:AccessMethodType"/>
            </xs:choice>
        </xs:complexType>
        <!-- ============== AccessMethodType ================= -->
        <xs:complexType name="AccessMethodType">
            <xs:sequence>
                <xs:element ref="wcts:OnlineResource"/>
            </xs:sequence>
        </xs:complexType>
        <!-- ============== KnownTransformationType ================= -->
        <xs:complexType name="SupportedTransformationType">
            <xs:sequence>
                <xs:element name="coordinateTransformationID" type="gml:ExtendedIdentifierType"/>
            </xs:sequence>
        </xs:complexType>
</xs:schema>
```

A.6 IsTransformable Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/wcts "
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wcts="http://schemas.opengis.net/wcts"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:import schemaLocation="WCTSCommon.xsd"/>
    <!-- ======================================================= -->
    <!--                    element definitions                  -->
    <!-- ======================================================= -->
    <!-- root element of an IsTtransformable request -->
    <xs:element name="IsTransformable" type="wcts:IsTransformableType"/>
    <!-- sub elements -->
    <xs:element name="GeometricPrimitiveType" type="wcts:GeometricPrimitiveTypeType"/>
    <xs:element name="CoverageInterpolationMethod" type="wcts:CoverageInterpolationMethodType"/>
    <xs:element name="CoverageType" type="wcts:CoverageTypeType"/>
    <!-- ======================================================= -->
    <!--                    type definitions                     -->
    <!-- ======================================================= -->
    <xs:complexType name="IsTransformableType">
        <xs:sequence>
            <xs:element ref="wcts:SourceCRS"/>
            <xs:element ref="wcts:TargetCRS"/>
            <xs:element ref="wcts:GeometricPrimitiveType" minOccurs="0" maxOccurs="unbounded "/>
            <xs:element ref="wcts:CoverageType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element ref="wcts:CoverageInterpolationMethod" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="version" type="xs:string" fixed="0.1.0"/>
    </xs:complexType>
</xs:schema >
```

A.7 IsTransformable Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/wcts "
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wcts="http://schemas.opengis.net/wcts"
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <!-- ======================================================== -->
    <!--                        element definitions               -->
    <!-- ======================================================== -->
    <!-- root element of a transformable response -->
    <xs:element name="IsTransformableResponse" type="wcts:IsTransformableResponseType"/>
    <!-- ======================================================== -->
    <!--                        type definitions                  -->
    <!-- ======================================================== -->
    <!-- response type -->
    <xs:complexType name="IsTransformableResponseType">
        <xs:attribute name="isTransformable" type="xs:boolean" use="required"/>
    </xs:complexType>
</xs:schema >
```

A.8 Transform Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/wcts "
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:gml="http://www.opengis.net/gml"
xmlns:wcts ="http://schemas.opengis.net/wcts " elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:import schemaLocation ="../gml3.0/base/coordinateOperations.xsd"/>
    <xs:import schemaLocation ="WCTSCommon.xsd"/>
    <!-- ========================================================= -->
    <!--                       element definitions                            -->
    <!-- ========================================================= -->
    <!-- root element of a transform request -->
    <xs:element name="Transform" type="wcts:TransformType"/>
    <!-- sub elements -->
    <xs:element name="Data" type="wcts:DataType"/>
    <xs:element name="RemoteOWS" type="wcts:RemoteOWSType "/>
    <xs:element name="OWSRequest" type="wcts:OWSRequestType"/>
    <xs:element name="Service" type="wcts:ServiceType"/>
    <xs:element name="KVPRequest" type="xs:string"/>
    <xs:element name="XMLRequest" type="xs:string"/>
    <!-- ========================================================= -->
    <!--                       type definitions                                -->
    <!-- ========================================================= -->
    <xs:complexType name="TransformType">
        <xs:sequence>
            <xs:element ref="wcts:SourceCRS"/>
            <xs:element ref="wcts:TargetCRS"/>
            <xs:element ref="gml:_CoordinateOperation" minOccurs="0"/>
            <xs:element ref="wcts:Data" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="version" type="xs:string" use="required" fixed="0.1.0"/>
    </xs:complexType>
    <xs:complexType name="DataType">
        <xs:choice>
            <xs:element name="FileURL"/>
            <xs:element ref="gml:FeatureCollection"/>
            <xs:element ref="wcts:RemoteOWS"/>
        </xs:choice>
    </xs:complexType>
    <xs:complexType name="RemoteOWSType">
        <xs:annotation>
            <xs:documentation >
            A RemoteOWS gives a reference to a remote WFS/WCS/other-OWS server.
            </xs:documentation>
        </xs:annotation>
        <xs:sequence>
            <xs:element ref="wcts:OnlineResource"/>
            <xs:element ref="wcts:OWSRequest"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="OWSRequestType">
        <xs:choice>
            <xs:element ref="wcts:KVPRequest"/>
            <xs:element ref="wcts:XMLRequest"/>
        </xs:choice>
    </xs:complexType>
    <xs:simpleType name="ServiceType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="WFS"/>
            <xs:enumeration value="WCS"/>
        </xs:restriction>
```

```
      </xs:simpleType>
</xs:schema >
```

A.9 Transform Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/wcts " xmlns:wcts="http://schemas.opengis.net/ wcts "
xmlns:gml="http://www.opengis.net/gml" xmlns:xs="http://www.w3.org/2001/XMLSchema "
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:import schemaLocation ="WCTSCommon.xsd"/>
    <xs:import schemaLocation ="../gml3.0/base/feature.xsd"/>
    <!-- ======================================================= -->
    <!--                    element definitions                  -->
    <!-- ======================================================= -->
    <!-- root element of a transform response -->
    <xs:element name="TransformResponse" type="wcts:TransformResponseType"/>
    <!-- ======================================================= -->
    <!--                      type definitions                   -->
    <!-- ======================================================= -->
    <xs:complexType name="TransformResponseType">
        <xs:sequence>
            <xs:element ref="gml:_Feature" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

A.10 DescribeTransformation Request

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/wcts " xmlns:wcts="http://schemas.opengis.net/wcts "
xmlns:xs="http://www.w3.org/2001/XMLSchema " elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xs:import schemaLocation ="WCTSCommon.xsd"/>
    <!-- ======================================================= -->
    <!--                      element definitions                           -->
    <!-- ======================================================= -->
    <!-- root element of a transform request -->
    <xs:element name="DescribeTransformation" type="wcts:DescribeTransformationType"/>
    <!-- sub elements -->
    <!-- ======================================================= -->
    <!--                       type definitions                             -->
    <!-- ======================================================= -->
    <xs:complexType name="DescribeTransformationType">
        <xs:sequence>
            <xs:element ref="wcts:SourceCRS"/>
            <xs:element ref="wcts:TargetCRS"/>
        </xs:sequence>
        <xs:attribute name="version" type="xs:string" fixed="0.1.0"/>
    </xs:complexType>
</xs:schema>
```

OGC 02-061r2

A.11 DescribeTransformation Response

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://schemas.opengis.net/wcts " xmlns:wcts="http://schemas.opengis.net/wcts"
xmlns:gml="http://www.opengis.net/gml" xmlns:xs="http://www.w3.org/2001/XMLSchema "
elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:import schemaLocation="../gml3.0/base/coordinateOperations.xsd"/>
    <!-- ======================================================= -->
    <!--                  element definitions                    -->
    <!-- ======================================================= -->
    <!-- root element of a transform request -->
    <xs:element name="DescribeTransformationResponse" type="wcts:DescribeTransformationResponseType"/>
    <xs:element name="CoordinateOperationID" type="gml:ExtendedIdentifierType"/>
    <!-- ======================================================= -->
    <!--                    type definitions                     -->
    <!-- ======================================================= -->
    <!-- response type -->
    <xs:complexType name="DescribeTransformationResponseType">
        <xs:sequence>
            <xs:element ref="wcts:CoordinateOperationID" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```

Annex B
(normative)

Conformance Test

B.1 Introduction

Specific conformance tests for Web Coordinate Transformation Service have not yet been determined and will be added in a future revision of this specification.

Bibliography

[9]   de La Beaujardière, Jeff (ed.), "OpenGIS Implementation Specification #02-068r3: Web Map Service Implementation Specification", January 2002

[10]   Vretanos, Panagiotis A. (ed.), "OpenGIS Implementation Specification #01-065: Web Feature Service Implementation Specification", May 2002

39