TITLE: <OWS+WCS+CIS+SensorML\*>Common Object Model Container (COMC)

Author Name (s): Lucio Colaiacomo, Larry Beck, Jason Smith

Email: luciocol@mac.com, Michael.l.beck@vencore.com

DATE: 03 November 2016

CATEGORY: SWG Charter

To:  OGC members & interested parties

A new OGC Standards Working Group is being proposed. The OGC members listed below have proposed the OGC Common Object Model Container SWG.  The SWG proposal provided in this document meets the requirements of the OGC TC Policies and Procedures.

The SWG name, statement of purpose, scope, list of deliverables, audience, and language specified in the proposal will constitute the SWG's official charter. Technical discussions may occur no sooner than the SWG's first meeting.

This SWG will operate under the OGC 2007 IPR Policy. The eligibility requirements for becoming a participant in the SWG at the first meeting (see details below) are that:

* You must be an employee of an OGC member organization or an individual
member of OGC;
* The OGC member must have signed the OGC Membership agreement;
* You must notify the SWG chair of your intent to participate to the first meeting. Members may do so by logging onto the OGC Portal and navigating to the Observer page and clicking on the link for the SWG they wish to join and;
* You must attend meetings of the SWG. The first meeting of this SWG is at the time and date fixed below. Attendance may be by teleconference.

Participants also may join the SWG at any time. The OGC and the SWG welcomes all interested parties.

Non-OGC members who wish to participate may contact us about joining the OGC. In addition, the public may access some of the resources maintained for each SWG: the SWG public description, the SWG Charter, Change Requests, and public comments, which will be linked from the SWG’s page.

Please feel free to forward this announcement to any other appropriate lists. The OGC is an open standards organization; we encourage your feedback.

#  Common Object Model Container

The Name of this SWG shall be the “Common Object Model Container.”

#  Purpose of this Standards Working Group

We are reaching in OGC an important operational level for data and services interoperability...

The momentum we have enables us to think about the possibility to rid ourselves of different data encodings and specific server specifications. A user should be able to publish data and retrieve it independently from its nature and format.

We all use different data (encodings) and we need to publish these in services or have them available in many different environments. There are encoding types (imagery, video, Audio, OS, …) that oblige us to have specific software and platforms to be used, reducing the flux of information and creating bottlenecks in its usage.

There is a need for accessing the “information” no matter the service type (map, feature, coverage, etc.) or environment. We need the information derived by services to be directly re-usable. We need “services as information.”

The idea is that we have different objects (like files or messages, etc.) that normally are passive and manipulated by high level algorithms. We need active objects with their own behavior and state. Objects that can be streamed with a mechanism to react and considers the status and the object’s behavior. This should be done in either asynchronous or synchronous mode.

The basis for accessing data should be any communication technology in which symmetrical client/server communication is foreseen but we should consider future connectionless systems and eventually multicast mode. So a digital container wrapping any type of data.

For the above reasons, we need a high level of abstraction which is architecture and operative system independent.

We should define a high hierarchical model based on context containers and is partitioned into domains which are separate hierarchies built upon “objects” and containers.

Domains can be connected through “portals/servers” and all hosts should have at least one domain. These domains should be built on the real objects that belong to the hosts. When any two objects connect, they will exchange hierarchies.

We will consider having event driven systems and the interactions between objects done through “messages”, with a message management system that can update the messages if they lose scope. For example, mpeg4 messages could be sent directly to the receiving object and optionally queued for later delivery, based on an asynchronous transfer mode protocol.

#  Basic Concept

The proposed idea for the COMC is to leverage two existing OGC standards to build the container as a combination of:

* + OWS CONTEXT
	+ Coverage Implementation Schema (CIS) (formerly GMLCOV)

Considering the above assumption, the possible benefits are:

* + Encoding independency
	+ OGC W\*S independency

#  OWS Context

In order to have COMC properly implemented, it is important to have an external file with a unique UUID linking the data and the metadata file. This unique link should be done in such a way that all resources have a unique identifier following (ITU-T X.667) and with the OWS Common GetResourceByID operation properly implemented.

The metadata definition should have the security classification embedded and should help in avoiding GetCapabilities long answers from the services.

OWS Context was designed for the Cloud as well as Off-Line processing:

OGC Web Services Context (OWS Context) was created to allow a set of configured information resources (service set) to be passed between applications primarily as a collection of services.

OWS Context was developed to support in-line content and external links

The goal is to support use cases such as the distribution of search results, the exchange of a set of resources such as OGC Web Feature Service (WFS), Web Map Service (WMS), Web Map Tile Service (WMTS), Web Coverage Service (WCS) and others in a ‘common operating picture’ (COP).

Additionally, OWS Context can deliver a set of configured processing services (Web Processing Service (WPS)) parameters to allow the processing to be reproduced on different nodes.

**OWS CONTEXT**

**ENVELOPE**

**IMAGERY**

**FEATURES**

**Area of Interest**

**GeoPDF**

 **Motion**

**Services**

**Figure 1. OWS Context**

#  Coverage Implementation Schema (CIS) (formerly GMLCOV)

Coverage Implementation Schema (CIS) specifies the OGC coverage model by establishing a concrete, interoperable, conformance-testable coverage structure. Based on the abstract concepts of OGC Abstract Topic 6 [1] (which is identical to ISO 19123). CIS is interoperable in the sense that coverages can be conformance tested, regardless of their data format encoding, down to the level of single “pixels” or “voxels.”

Coverages can be encoded in any suitable format (such as: NITF/NSIF, J2K, JP2, JPEG DCT, GeoTIFF, Net-CDF, XML, GML, JSON, MP3, WAV, MPEG4, H.264, H.265, Motion J2K) and can be partitioned (e.g., for a time-interleaved representation). Coverages are independent from service definitions and, therefore, can be accessed through a variety of OGC services types Such as the Web Coverage Service (WCS) Standard [6]. The coverage structure can serve a wide range of coverage application domains, thereby contributing to harmonization and interoperability between and across these domains.

* **Encodings easily adapted to new format**
	+ Including a link with a unique identifier
	+ Embed data, raw or formatted, directly in the file

 Examples:

* + - Still:
			* Formatted: NITF/NSIF, GeoTIFF, GMLJP2, GeoJP2, JP2, JPX
			* Raw: J2K or JPEG DCT
		- Motion:
			* Formatted: MP4, Motion J2K
			* Raw: H.264, H.265
		- Audio: MP3 or WAV
		- Metadata: XML, GML, JSON

**Coverage Implementation Schema (CIS)**

**COVERAGES**

**GRID**

**RECTIFIED**

**REFEREANCEABLE (SENSORS)**

**POINT CLOUD**

**ANY…FEATURE… (TIME, Etc…)**

**Figure 2. Coverage Implementation Schema (CIS)**

#  The Common Object Model Container

If we merge the capabilities of CIS with OWS Context, we obtain a descriptor that is independent (besides the link with the specific imagery) from the encoding of the imagery itself.

Consider the case of a xml box inside a jp2 file. Following gmljp2 specifications, the box can be referenced in an external file with a simple href link. We now have a xml container that is always structured the same way independently from the specific imagery format. This gives us a structure of files (imagery, vector, pdf, video, etc..) that have a unique link (UUID) with the metadata and the OWS Context.



**Figure 3. Common Object Model Container**

At this point we have a structure composed of a OWS Context xml file referencing a CIS xml file and data (which may be embedded to form only a single file). When all of the data has the same href to the same structured CIS description (raster and features, etc.) we have a descriptor enabling the transport to the servers of any data type. On the server side a user should not be aware of the type of service to discover the data needed, because the nature of such data is that it is appropriate to be directly served from an OGC web service (independent it nature) with WCS extending its purpose to handle features, in the case of this example.

We should have a W\*S in which the coverages and features are served as objects and the user application can distinguish the type and format and process or manage those objects appropriately. This should lead to the possibility of having a REQUEST to a W\*S which returns the objects in a given time/space range and the “GET” of the object activates at the server level the retrieval of the information in its original format (raster, features, doc, etc.).

#  Additional Possibilities

#  Encapsulation

COMC could be encapsulated in an EXI, or any other transport mechanism, as a message body. This also allow the user to save views of the operational scenario in the form of a “message” for storing in a Web Streaming Service (WSS).

#  Streaming

The streaming of messages and objects could be made through any transport mechanism and could be managed by WSS / CIS combinations implementing a “message” management system. The streaming protocol can be any protocol (including real time or near real time) running on a variety of transport protocols.

Object/data

Messages/ COMC

The data streaming would move in parallel with COMC streaming, the two linked via UUID identification in each stream. Alternately, the COMC could be embedded into the data stream. The bulkiness of the XML can be greatly reduced with EXI with or without compression or replaced with JSON, GeoJSON, or another data-interchange format. In this way, it is possible to transmit quickly a Common Operating Picture (COP) using the COMC with just the description of the various objects composing the scenario.

See [**Appendix A for an example**](#_Appendix_A_–)

#  Business Value Proposition

The work of this proposed SWG would simplify the effort software companies have to expend each time a new format/encoding is available and speed the adoption process of the new encoding by improving the user experience in quickly and efficiently accessing the data through OGC services.

#  Scope of Work

This SWG should develop a new Common Object Model Container standard integrating the existing standards OWS Context, WCS, GMLJP2, and perhaps others. By relying on existing standards, COMC should be more readily adopted by the market.

The work will include:

Identify portions of OWS Context and CIS necessary to build COMC

Develop draft standard for interim public review

Develop candidate standard for formal OGC review and approval.

Identify use cases and define tests

#  Statement of relationship of planned work to the current OGC standards baseline

GMLJP2

OWS Common

OWS Context

SensorML

WCS

#  What is Out of Scope?

This SWG will not make any edits to the underlying standards used to develop this new standard.

#  Specific Contribution of Existing Work as a Starting Point

The SWG is based on previously developed Standards and Testbeds:

Coverage Implementation Schema (CIS) (formerly GMLCOV)

GMLJP2

OWS context

WCS

#  Determination of SWG Completion

The COMC SWG will become inactive after the following three milestones have been achieved:

1. The SWG has completed evaluation and incorporation into the candidate standard of all comments received during the public comment period.
2. Approval by the SWG membership of a recommendation to submit the document to the TC for consideration as an OGC Adopted Standard.
3. The candidate standard has been approved by the OGC Technical and Planning Committees as an Adopted OGC standard.

#  Is this a persistent SWG?

## Yes

#  When can SWG be inactivated?

At the time the criteria in 4.4 are completed the SWG membership will determine if additional work is needed. If not, then the SWG will be inactivated.

#  Description of Deliverables

Deliverables of this project is a new candidate standard COMC and eventually associated Best Practice documents. An ad hoc meeting for this proposed SWG was held at the Orlando TC Meeting in September 2016. The submitters forecast release of a draft candidate standard 8 months after creation of the SWG.

#  IPR Policy for this SWG

RAND-Royalty Free.

#  Anticipated Participants

This SWG is intended to address the services (CIS) and the data (geopackage, gmljp2, geotiff, etc.) in order to harmonize the information flow and maintain the coherence of the metadata. Moreover, this should facilitate the seamless data discovery from a generic user accessing OGC services (no matter if the data is vector, raster, etc.).

#  Other Informative Remarks about this SWG

#  Similar or Applicable Standards Work (OGC and Elsewhere).

#  Details of the First Meeting

The first meeting of the SWG will be held by web conference within four weeks of approval of the SWG. Call-in information will be provided to the SWG's e-mail list and on the portal calendar in advance of the meeting.

#  Projected On-going Meeting Schedule

The work of the SWG will be carried out primarily by email and web conferences, possibly every two weeks, with face-to-face meetings perhaps at each of the OGC TC meetings.

#  Supporters of the Proposal (Charter Members)

The following people support this proposal and are committed to the Charter and projected meeting schedule. These members are known as SWG Founding or Charter members. The charter members agree to the SoW and IPR terms as defined in this charter. The charter members have voting rights beginning the day the SWG is officially formed. Charter Members are shown on the public SWG page. Extend the table as necessary.

|  |  |
| --- | --- |
| Name | Organization |
| Lucio Colaiacomo | EU Satcen |
|  | NGA |
| Larry Beck | Vencore |
| Peter Baumann | Jacob University Bremen |
| Jason Smith | Harris Corp. |

#  Convener(s)

Name of individual(s) who started the SWG process. Could be the lead for an RFC submission, an OGC staff person, or an individual who believes it is time for a revision to an adopted standard.

Lucio Colaiacomo

Larry Beck

# Appendix A – Example 1

The following is an example of the OWS context description currently in use that could be associated to the CIS structure

Today we can deliver data with an OWS Context description as follows:

<?xml version="1.0" encoding="utf-8"?>

<feed xmlns:xsi="<http://www.w3.org/2001/XMLSchema-instance>" xmlns:dc="<http://purl.org/dc/elements/1.1/>" xmlns:georss="<http://www.georss.org/georss/10>" xmlns:gml="<http://www.opengis.net/gml>" xsi:schemaLocation="<http://www.opengis.net/owc/1.0> <https://raw.githubusercontent.com/joanma747/owscontext-xsd/master/owc.xsd>   <http://purl.org/dc/elements/1.1/> <http://dublincore.org/schemas/xmls/qdc/dc.xsd>  <http://www.georss.org/georss/10> <http://www.georss.org/xml/1.1/georss.xsd>" xmlns="<http://www.w3.org/2005/Atom>">

            <link rel="profile" href="<http://www.opengis.net/owc-atom/1.0/req/core>"/>

            <id>urn:uuid:1da405e3-97cc-4e4f-9ba7-8fab8ec2fb1c</id>

            <title> test product</title>

            <subtitle>, Product Type:DGI, Product Code:XXXX14001\_D001\_I001</subtitle>

            <updated>2014-10-10T14:03:17Z</updated>

            <dc:publisher>European Union Satellite Centre</dc:publisher>

            <generator uri="<http://www.satcen.europa.eu/Applications/CoderTool>" version="1.0">SatCen Coder Tools (ATOM OWS Context Tool)</generator>

            <rights>Copyright (c) 2014. All rights reserved. EU SatCen (European Union Satellite Centre)</rights>

            <dc:date>2014-02-27T00:00:00Z</dc:date>

            <category term="D" scheme="<http://www.satcen.europa.eu/ProductTypesCodes.xsd>" label="DGI"/>

            <category term=“NOT CLASSIFIED" scheme="<http://www.satcen.europa.eu/Classification.xsd>" label=“NOT CLASSIFIED"/>

            <georss:where>

                        <gml:Polygon srsName="EPSG:4326">

                                    <gml:exterior>

                                                <gml:LinearRing>

                                                            <gml:posList srsDimension="2">nn.877820544 nn.1541684600001 nn..9264677580001 nn.1541684600001 nn.9264677580001 nn.2303528520001 nn.877820544 nn.2303528520001</gml:posList>

                                                </gml:LinearRing>

                                    </gml:exterior>

                        </gml:Polygon>

            </georss:where>

            <entry>

                        <id>urn:uuid:XXXXXXXXXX</id>

                        <title>imagery.jp2</title>

                        <content type="text">Platform:QuickBird-1,Instrument:BGIS,sensorType:OPTICAL,operationalMode:MS,resolution:9.8310238960306E-06,</content>

                        <updated>2014-03-20T00:00:00Z</updated>

                        <georss:where>

                                    <gml:Polygon srsName="EPSG:4326">

                                                <gml:exterior>

                                                            <gml:LinearRing>

                                                                        gml:posList srsDimension="2">xx.8611316749895 xx.1506294012071 xx.934933171377 xx.1506294012071 xx.934933171377 13.326417939492 xx.8611316749895 xx.326417939492</gml:posList>

                                                            </gml:LinearRing>

                                                </gml:exterior>

                                    </gml:Polygon>

                        </georss:where>

            </entry>

            <entry>

                        <id>dfdd:LAL260</id>

                        <title>Wall</title>

                        <content type="text">A solid man-made barrier of generally heavy material used as an enclosure, boundary, or for protection.</content>

                        <updated>2014-10-10T00:00:00Z</updated>

                        <georss:where>

                                    <gml:Polygon srsName="EPSG:4326">

                                                <gml:exterior>

                                                            <gml:LinearRing>

                                                                        <gml:posList srsDimension="2">-90 -180 90 -180 90 180 -90 180</gml:posList>

                                                            </gml:LinearRing>

                                                </gml:exterior>

                                    </gml:Polygon>

                        </georss:where>

            </entry>

            <entry>

                        <id>dfdd:LAL070</id>

                        <title>Fence</title>

                        <content type="text">A man-made barrier of relatively light structure used as an enclosure or boundary. (Similar structures that are constructed of heavy materials (for example: stone, rock or masonry) are classified as walls.)</content>

                        <updated>2014-10-10T00:00:00Z</updated>

                        <georss:where>

                                    <gml:Polygon srsName="EPSG:4326">

                                                <gml:exterior>

                                                            <gml:LinearRing>

                                                                        <gml:posList srsDimension="2">-90 -180 90 -180 90 180 -90 180</gml:posList>

                                                            </gml:LinearRing>

                                                </gml:exterior>

                                    </gml:Polygon>

                        </georss:where>

            </entry>

            <entry>

                        <id>dfdd:PAL015</id>

                        <title>Building</title>

                        <content type="text">A relatively permanent structure, roofed and usually walled and designed for some particular use.</content>

                        <updated>2014-10-10T14:03:17Z</updated>

                        <georss:where>

                                    <gml:Polygon srsName="EPSG:4326">

                                                <gml:exterior>

                                                            <gml:LinearRing>

                                                                        <gml:posList srsDimension="2">-90 -180 90 -180 90 180 -90 180</gml:posList>

                                                            </gml:LinearRing>

                                                </gml:exterior>

                                    </gml:Polygon>

                        </georss:where>

            </entry>

                                                </feed>

1) All resources should have a unique identifier (ITU-T X.667)

2) The OWS Common GetResourceByID operation should be implemented as specified.

If we consider the imagery part above

<id>urn:uuid:XXXXXXXXXX</id>

                        <title>imagery.jp2</title>

                        <content type="text">Platform:QuickBird-1,Instrument:BGIS,sensorType:OPTICAL,operationalMode:MS,resolution:9.8310238960306E-06,</content>

                        <updated>2014-03-20T00:00:00Z</updated>

                        <georss:where>

                                    <gml:Polygon srsName="EPSG:4326">

                                                <gml:exterior>

                                                            <gml:LinearRing>

                                                                        <gml:posList srsDimension="2">xx.8611316749895 xx.1506294012071 xx.934933171377 xx.1506294012071 xx.934933171377 nn.326417939492 xx.8611316749895 xx.326417939492</gml:posList>

we can see that the imagery have some metadata coming from the gmlcov description like:

 <gmljp2:featureMember>

 <gmljp2:GMLJP2RectifiedGridCoverage gml:id="ID\_1">

 <gml:domainSet>

 <gml:RectifiedGrid gml:id="rg0001\_C0002" dimension="2"

srsName="http://www.opengis.net/def/crs/EPSG/0/4326">

 <gml:limits>

 <gml:GridEnvelope>

 <gml:low>0 0</gml:low>

 <gml:high>6208 3103</gml:high>

 </gml:GridEnvelope>

 </gml:limits>

 <gml:axisName>x</gml:axisName>

 <gml:axisName>y</gml:axisName>

 <gml:origin>

 <gml:Point gml:id="P0001">

 <gml:pos>nn.4756576954751 n.04345415526392</gml:pos>

 </gml:Point>

 </gml:origin>

 <gml:offsetVector>0 3.7163372079825e-005</gml:offsetVector>

 <gml:offsetVector>-3.71633720798261e-005 0</gml:offsetVector>

 </gml:RectifiedGrid>

 </gml:domainSet>

 <gml:rangeSet>

 <gml:File>

 <gml:rangeParameters/>

 <gml:fileName>gmljp2://codestream/0</gml:fileName>

 <gml:fileStructure>inapplicable</gml:fileStructure>

 </gml:File>

 </gml:rangeSet>

 <gmlcov:rangeType>

 <swe:DataRecord>

 <swe:field name="Panchromatic">

 <swe:Quantity definition="http://www.opengis.net/def/ogc-eo/opt/SpectralMode/PANCHROMATIC">

 <swe:description>Panchromatic Channel</swe:description>

 <swe:uom code="unity"/>

 <!-- Unity for value without unit of measures -->

 </swe:Quantity>

 </swe:field>

 </swe:DataRecord>

 </gmlcov:rangeType> <gmlcov:metadata>

 <gmljp2:Metadata>

 <gmljp2:eopMetadata>

 <eop:EarthObservation

 xmlns:xlink="http://www.w3.org/1999/xlink"

 xmlns:eop="http://www.opengis.net/eop/2.0"

 xmlns:opt="http://www.opengis.net/opt/2.0"

 xmlns:ows="http://www.opengis.net/ows/2.0"

 xmlns:swe="http://www.opengis.net/swe/1.0"

 xmlns:om="http://www.opengis.net/om/2.0"

gml:id="opt\_example\_1">

 <om:phenomenonTime>

 <gml:TimePeriod gml:id="tp\_2">

 <gml:beginPosition>2001-08-22T11:02:47.000</gml:beginPosition>

 <gml:endPosition>2001-08-22T11:02:47.999</gml:endPosition>

 </gml:TimePeriod>

 </om:phenomenonTime>

 <om:resultTime>

 <!-- repeat of archiving date? -->

 <gml:TimeInstant gml:id="archivingdate\_1">

 <gml:timePosition>2001-08-22T11:02:47.999</gml:timePosition>

 </gml:TimeInstant>

 </om:resultTime>

 <om:procedure>

 <eop:EarthObservationEquipment gml:id="eop\_2">

 <eop:platform>

 <eop:Platform>

 <eop:shortName>PHR</eop:shortName>

 <eop:serialIdentifier>1A</eop:serialIdentifier>

 </eop:Platform>

 </eop:platform>

 <eop:instrument>

 <eop:Instrument>

 <eop:shortName>PHR</eop:shortName>

 </eop:Instrument>

 </eop:instrument>

 <eop:sensor>

 <eop:Sensor>

 <eop:sensorType>OPTICAL</eop:sensorType>

 <eop:operationalMode

codeSpace="urn:eop:PHR:sensorMode">PX</eop:operationalMode>

 <eop:resolution uom="m">0.7</eop:resolution>

 </eop:Sensor>

 </eop:sensor>

 <eop:acquisitionParameters>

 <eop:Acquisition>

 <eop:orbitNumber>12</eop:orbitNumber>

 <eop:lastOrbitNumber>12</eop:lastOrbitNumber>

 <eop:orbitDirection>ASCENDING</eop:orbitDirection>

 <eop:illuminationAzimuthAngle

uom="deg">10</eop:illuminationAzimuthAngle>

 <eop:acrossTrackIncidenceAngle

uom="deg">14.0</eop:acrossTrackIncidenceAngle>

 <eop:alongTrackIncidenceAngle

uom="deg">13.9</eop:alongTrackIncidenceAngle>

 <eop:pitch uom="deg">0</eop:pitch>

 <eop:roll uom="deg">0</eop:roll>

 <eop:yaw uom="deg">0</eop:yaw>

 </eop:Acquisition>

 </eop:acquisitionParameters>

 </eop:EarthObservationEquipment>