Doc. Id: HMAFOT1-TN-0001-ERDAS-10
Issue: 1.3
Date: 2010-01-27

ESA-ESRIN

Technical Note on the proposed EO EP of ebRIM CSW improvements

erdas
The Earth to Business Company

Science & Technology Facilities Council
Rutherford Appleton Laboratory

gim
geographic information management

January 2010

EXECUTIVE SUMMARY

This technical note describes the evolutions that are proposed for the EO Extension Package of CSW ebRIM (OGC 06-131) to make it more extensible for encorporating additional EO product types and mission specific metadata schemas.

It is submitted by the consortium consisting of ERDAS, GIM and STFC in the frame of HMA Follow On Task 1 project.

The current version of this document is to be considered as a proposal for the work to be done and is intended to start the discussion with the different stakeholders involved in the HMA-FO project.

SIGNATURES

	Name	Date	Signature
Authors:	Frederic Houbie (ERDAS) Steven Smolders (GIM)	2010/01/18	

DOCUMENT STATUS SHEET

Issue	Rev.	Date	Reason for change
1	0	2009-11-21	First issue – draft
1	1	2009-12-02	Revision by GIM
1	2	2010-01-14	Update by ERDAS/GIM following review by ESA
1	3	2010-01-27	Update by ERDAS following second review by ESA

APPLICABLE DOCUMENTS

[AD01] OGC 06-080r4: GML Application Schema for EO Products, Version 0.9.3, 12/09/2008.

[AD02] OGC 06-131r6, EO Products Extension Package for ebRIM Profile of CSW 2.0.2, Version 0.2.5.

[AD03] OGC 07-110r4, CS-W ebRIM Registry Service Version 1.0.1, 05/02/2009.

REFERENCE DOCUMENTS

[RD01] OGC 09-163, sensorML Extension Package for ebRIM Profile of CSW 2.0, Version 0.0.3, 28/09/2009.

[RD02] OGC 06-121r8, OGC Web Services Common 1.2 Implementation Standard

[RD03] SOAP Message Transmission Optimisation Mechanism, http://www.w3.org/TR/2004/WD-soap12-mtom-20040608

[RD04] XML-binary Optimized Packaging, W3C Recommendation 25 January 2005, www.w3.org/TR/xop10

[RD05] SOAP 1.1 Binding for MTOM 1.0, W3C Member Submission 05 April 2006http://www.w3.org/Submission/soap11mtom10

[RD06] WS-I Basic Profile version 1.2; http://www.ws-i.org/

[RD07] Multipurpose Internet Mail Extensions (MIME), IETF RFC2045, RFC 2046, RFC 2047, RFC 2048 and RFC 2049

ACCRONYMS

CIM	Cataloguing of ISO Metadata
CR	Change Request
CSW	Catalogue Service For the Web
ebRIM	e-Business Registry Information Model
EO	Earth Observation
EP	Extension Package
ESA	European Space Agency
ESRIN	European Space Research Institute
GML	Geography Markup Language
HMA	Heterogeneous Missions Accessibility
HMA-I	Heterogeneous Missions Accessibility –Interoperability (Initial HMA Project)
HMA-T	Heterogeneous Missions Accessibility –Testbed
HMA-FO	Heterogeneous Missions Accessibility –Follow On
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
ISO	International Standards Organisation
MIME	Multipurpose Internet Mail Extensions
MTOM	Message Transmission Optimization Mechanism
OASIS	Organisation for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
RFC	Request for Comments
SensorML	Sensor Modelling Language
SOA	Service Oriented Architecture
SAML	Security Assertion Markup Language

SOAP	Simple Object Access Protocol
SSE	Service Support Environment
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
XML	extensible Markup Language
XOP	XML-binary Optimized Packaging

TABLE OF CONTENTS

1 ANALYSIS OF THE ISSUES ASSOCIATED WITH THE EO EP OF EBRIM CSW

Normal practice when developing an ebRIM CSW Extension Package is to only map the most important metadata elements – the ones that should either be usable as search criteria, the so-called queryables, and/or the "base" returnables, the set of metadata elements that one would like to see in response to a search-like operation (GetRecords) on to the ebRIM model within the so-called slots. Full metadata is only returned when executing the present-like GetRepositoryItem operation which will return the metadata with its full content in the native encoding.

This approach leads however to a large number of request/response messages if one wants to retrieve full metadata for multiple EO Products (one GetRecords request to retrieve the identifiers followed by n GetRepositoryItem requests where n equals the number of records returned in reply to the GetRecords query.)

The solution adopted in the HMA context to limit the number of request/response messages was to ensure that the GetRecords operation returns all metadata elements that are available. This was achieved by mapping all of the elements of the EO Profile of GML to ebRIM meaning that all EO Profile of GML elements are defined as slots within the EO extension package of ebRIM CSW. As such the number of operation calls is limited to the absolute minimum. Also by returning always all metadata elements, one avoids the discussions about which metadata element should be part of which elementSet.

The drawback of this solution is however the overall complexity of the ebRIM model and resulting from this, issues related to the extensibility and maintainability of the EO ebRIM CSW specification and associated GML Application Schema for EO Products which are further explained below.

1.1 COMPLEXITY OF THE MODEL STRUCTURE

In the EO extension package model, one product is split into several objects (EOProduct, EOAcquisitionPlatform, EOBrowseInformation, EOMaskInformation, EOArchivingStation, EOProductInformation). This allows reusing objects like EOAcquisitionPlatform for several products. This structure describes precisely the relation between the different parts of the product information but it is more complex for discovery of products. When querying the EO model, the query can become quite complex due to the number of objects implicated. Indeed, the CSW query must include several where clauses to cross the associations between the different objects. For instance, if you want to find products acquired within a period, with a given platform and archived at a specific place, you have to make a request on three different objects by crossing two associations.

To note is that a first improvement/simplification was already done in recent versions of the HMA Catalogue specifications by adding the RegistryPackage around all the objects, so that a client application can request a RegistryPackage object and thus receive all the information in one response with a simpler query.

The current model however contains information like for instance browseURL not needed for discovering products. This information is stored in the ebRIM model which introduces a high volume of data that are indexed in the storage area. A revised model should be carefully designed to store only information useful for users when they discover and evaluate products.

1.2 MAINTAINABILITY AND EXTENSIBILITY OF THE EO EBRIM CSW

The “full mapping” approach implies that the information contained within the EO Profile of GML gets completely duplicated in the EO extension package and thus creates problems of maintainability of both standards. A simple change in cardinality of one of the elements in the EO GML Application Schemas, has got drastic effects on the EO extension package affecting all implementers (change in cardinality typically means a new association).

With the full mapping approach it also becomes difficult to add specific metadata elements for specific purposes. If one adds metadata elements to the GML Application schema for EO Products within an extended GML application schema, one would also need to create a new ebRIM CSW extension Package document. This also needs to happen in case it just consists of metadata elements that should not be queryable. If the new metadata elements however also need to be considered as queryables, the creation of an additional extension package can not be avoided.

Instead of “applying the full mapping approach” to solve the issue with the number of requests, it would be more convenient to adapt the CSW-ebRIM specification to fulfill the business need by allowing to return the full metadata of a series of repository items in their original encoding (EO GML for the EO EP ebRIM CSW) in response to a single request. This document proposes a number of possible alternative solutions.

1.3 HARMONIZATION BETWEEN ISO19115-2, SENSORML AND EO GML

Apart from the issues mentioned above there also is an opportunity to align the EO GML with SensorML The acquisition platform describes the satellite that acquires the image. The EO GML contains a section describing the acquisition platform (sensorType, sensorResolution, instrumentShortName …). Another initiative is to describe all the sensors using the sensorML specification. SensorML allows to precisely describing platform, instrument, detector, instrument modes…

Moreover, the collections of EO Products are represented using the ISO19115 standard with part 2 extensions to add the Instrument details. The link between these models should be better described in a Best Practice document.

We need some harmonization between those specifications.

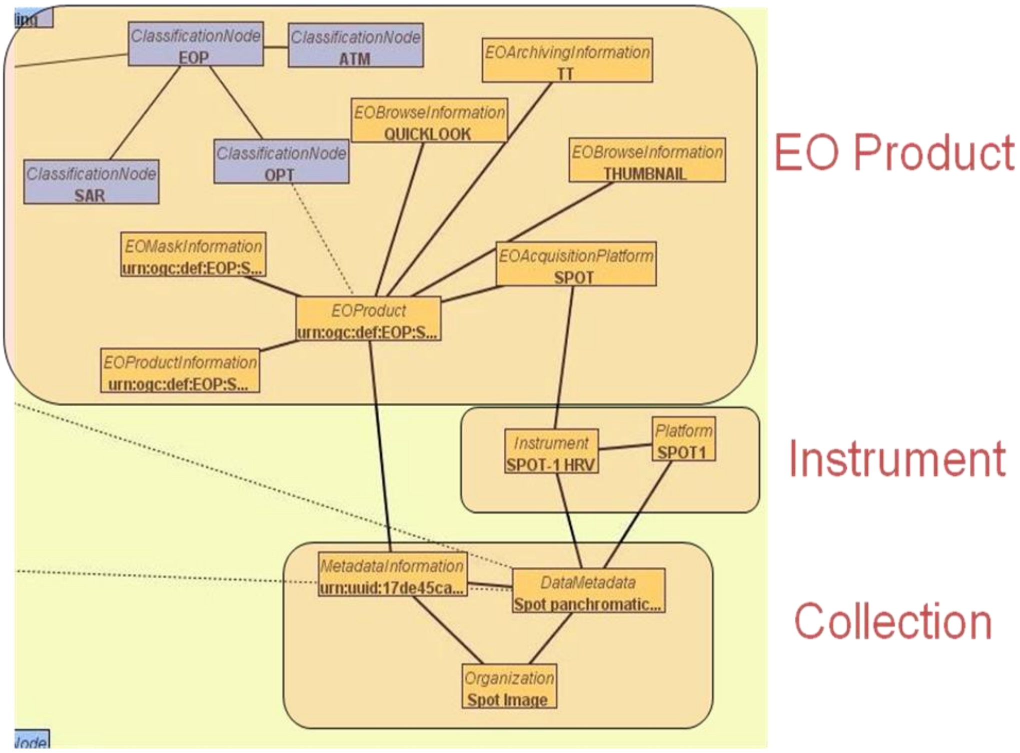

ClassificationNode EOP
ClassificationNode ATM
EOArchivingInformation TT
EOBrowseInformation QUICKLOOK
ClassificationNode SAR
ClassificationNode OPT
EOBrowseInformation THUMBNAIL
EOMaskInformation urn:ogc:def:EOP:S...
EOAcquisitionPlatform SPOT
EOProduct urn:ogc:def:EOP:S...
EOProductInformation urn:ogc:def:EOP:S...
EO Product
Instrument SPOT-1 HRV
Platform SPOT1
Instrument
MetadataInformation urn:uuid:17de45ca...
DataMetadata Spot panchromatic...
Organization Spot Image
Collection

Figure 1 : Collections, Product & Instrument linking

2 IMPROVEMENT PROPOSALS

This section contains a list of proposals to improve the usability of the EO Extension Package of ebRIM CSW.

2.1 SIMPLIFY THE MODEL

We propose to simplify the model in two ways:

- Simplify the structure by storing only two objects: EOProduct and EOAcquisitionPlatform
- Simplify the content by storing only parameters used for the discovery

Based on implementation experience, Figure 2 sketches a proposal for a simplified model for the EOP product level. For each sub level (OPT, SAR, ATM, ...), a specific section in the revised EO EP of EbRIM CSW document will define the required elements for the thematic extensions. Moreover, the EOP level might be restructured with the analysis of limb looking product type.

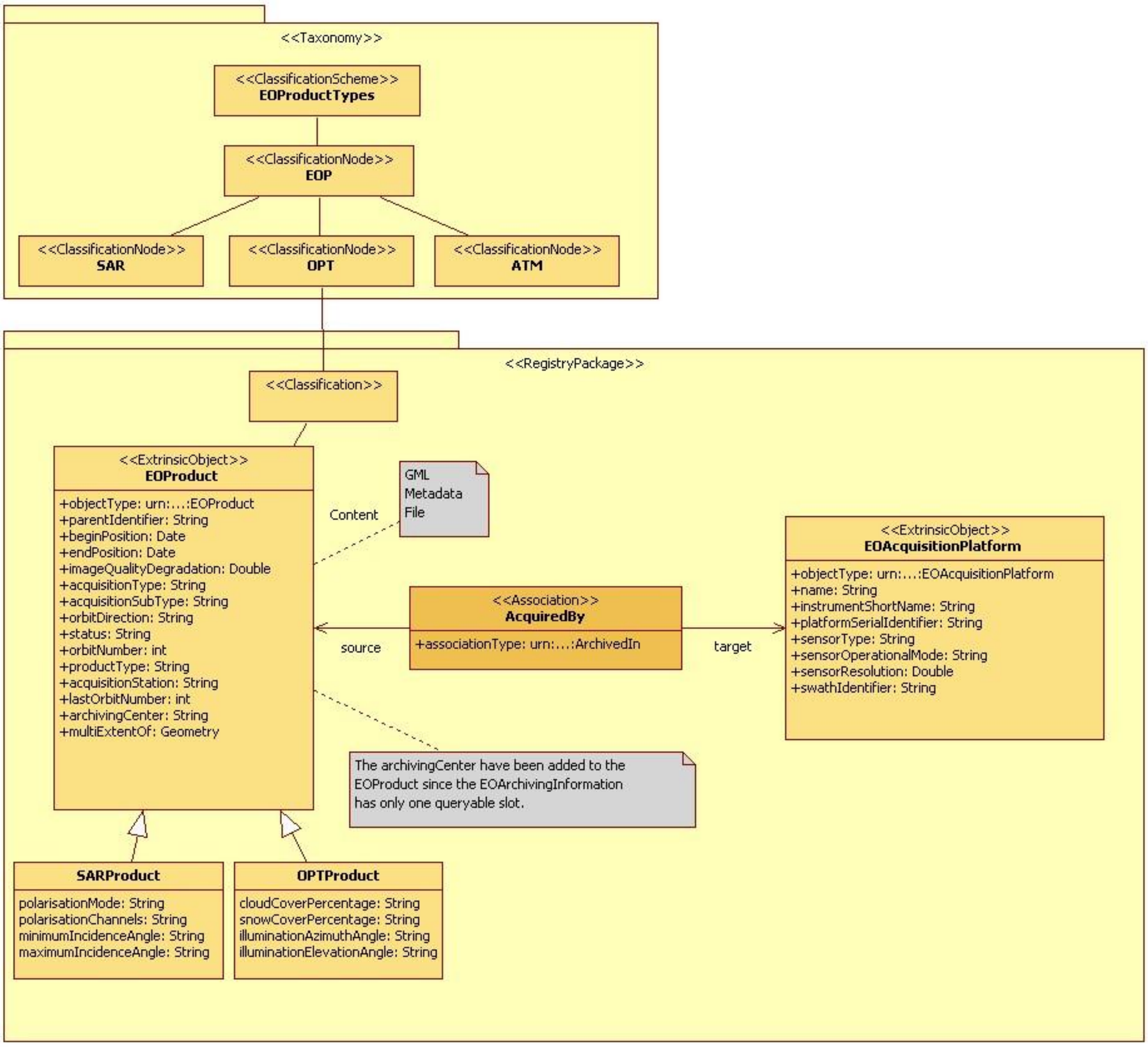

<<Taxonomy>>
<<ClassificationScheme>>
EOProductTypes
<<ClassificationNode>>
EOP
<<ClassificationNode>>
SAR
<<ClassificationNode>>
OPT
<<ClassificationNode>>
ATM
<<RegistryPackage>>
<<Classification>>
<<ExtrinsicObject>>
EOProduct
+objectType: urn:...:EOProduct
+parentIdentifier: String
+beginPosition: Date
+endPosition: Date
+imageQualityDegradation: Double
+acquisitionType: String
+acquisitionSubType: String
+orbitDirection: String
+status: String
+orbitNumber: int
+productType: String
+acquisitionStation: String
+lastOrbitNumber: int
+archivingCenter: String
+multiExtentOf: Geometry
Content
GML
Metadata
File
<<Association>>
AcquiredBy
+associationType: urn:...:ArchivedIn
source
target
<<ExtrinsicObject>>
EOAcquisitionPlatform
+objectType: urn:...:EOAcquisitionPlatform
+name: String
+instrumentShortName: String
+platformSerialIdentifier: String
+sensorType: String
+sensorOperationalMode: String
+sensorResolution: Double
+swathIdentifier: String
The archivingCenter have been added to the EOProduct since the EOArchivingInformation has only one queryable slot.
SARProduct
polarisationMode: String
polarisationChannels: String
minimumIncidenceAngle: String
maximumIncidenceAngle: String
OPTProduct
cloudCoverPercentage: String
snowCoverPercentage: String
illuminationAzimuthAngle: String
illuminationElevationAngle: String

Figure 2 Proposed revised EO EP of ebRIM CSW model structure

This model is based on the analysis of queriables supported by agencies reported in a conformance analysis provided from the HMA-AWG . The final content of this model will be decided after consultation with the different stakeholders (ESA, DLR, Eumetsat, CNES, …).

2.2 ALLOW TO PIGGYBACK THE EO GML IN THE CSW GETRECORD RESPONSE

In the current approach of the EO EP, the GetRepositoryItem call is not used as all EO GML elements are returned in ebRIM in response to the GetRecords request in order to avoid multiple web Service Requests.

The number of request/response messages could be reduced without performing a full EO GML to EO EP of ebRIM CSW mapping if the EO GML metadata could be added directly in the GetRecordsResponse.

It should be noted that as most EO EP of ebRIM CSW Catalogues are implemented as proxy implementations on top of legacy datastores using a proprietary database model, there are no file accesses required to return the EO GML. Instead the EO GML will be composed on the fly as is currently done with the ebRIM representation.

To allow this behavior, the CSW-ebRIM specification must be extended to support this inclusion of GML (or ISO19115 metadata in case of CIM) inside the GetRecords response.

Several ways could be envisaged to realise this:

At “content” level

- By incorporating the entire EO GML instance within a single additional slot
- By allowing elements similar to wrs:ExtrinisicObject to appear in response to a GetRecords request. wrs:ExtrinsicObject is defined in OGC07-110r4 [AD03] to allow the inclusion of RepositoryItems within CSW-ebRIM insert statements using either XOP [RD04] or multi-part MIME messages [RD07]. wrs:ExtrinsicObject therefore extends the rim:ExtrinsicObject to allow two additional elements:
    - "repositoryItemRef”: element which has an xlink:href attribute the value nof which shall be an URL according to the cid schema.
    - “repositoryItem”: that contains an xop:include element of which the href attribute of the xop:Include element must be a valid URI in accord with the ‘cid’ URL scheme.

  In order to avoid the difficulties that mulipart MIME and or XOP could cause (see further), a third option could potentially be introduced in the form of an additional element for instance named RepositoryItemXML that would allow inclusion of the RepositoryItem GML instance. The EO GML would then be included as repositoryItem associated with the main ExtrinsicObject (type EOProduct). Instead of adding a new element, one could also consider changing the definition of the repositoryItem element itself to allow the inclusion of the repositoryItem directly and/or by reference.

At “message” level:

- Return a zip or tar package containing multiple XML files: one normal ebRIM GetRecords response and several EO Profile of GML instance documents.
- Using multipart MIME messages where the first part would contain the ebXML structure and the further parts would contain the EO GML
- Using SOAP with attachments for SOAP 1.1
- Using SOAP with MTOM/XOP for SOAP 1.2
- Using Direct Internet Message Encapsulation (DIME) .

The solution at **content level** whereby a specific RepositoryItem slot would be created would be specific to the Earth Observation and/or CIM Extension Packages and hence would not entail changes to the underlying CSW and CSW-ebRIM standards.

As it does not entail any changes at the "message level" there is no risk of it

- not being supported by (Web Service) firewalls
- requiring more advanced server and client tools to implement applications
- affect already adopted security solutions within HMA

The drawback of the solution is that it – at least for native ebRIM implementations - mandates duplication of information inside the registry database and the fact that it is a workaround that may cause some aversion in the working group and associated difficulties in the adoption process.

The solution of using an extension of wrs:ExtrinisicObject to allow inclusion of GetRepositoryItem XML directly inside the GetRecords response fits more with the ebRIM-CSW filosophy. It requires a small change at ebRIM-CSW level but when this is done both CIM and EO EP can benefit from this new feature.

What concerns the alternatives **at "message level"**, support for zipped files and "normal" multipart MIME response messages in Commercial and Open source Off the Shelf Web Services oriented tools is limited. Within the HMA-T Phase 2 project, a set of tests were in the past conducted with the Oracle BPEL Controller that were not succesful. Also these techniques can be hindered by firewalls and are not compatible with WS-Security approaches.

Within the world of SOAP Web Services, three alternative standards have been defined to standardize the use of multipart messages when using SOAP:

- SOAP with attachement (SwA) defines a MIME multipart/related structure for packaging attachments with SOAP 1.1 messages. It is considered as being superseded by MTOM.
- DIME: Direct Internet Message Encapsulation is a lightweight, binary message format that was used to encapsulate SOAP messages and their attachments and was employed in the.Microsoft Web Services Enhancements. It is considered obsolete and replaced by MTOM.
- SOAP with MTOM and XOP: evolution of SwA that describes how to transfer binary data as part of a SOAP message in a more efficient manner. Defined for SOAP 1.2 but a separate document exists that specifies how MTOM can be used in conjunction with SOAP 1.1

To note is that MTOM/XOP are encorporated in the WS-I basic profile 1.2 and the forthcoming basic profile 2.0 and should hence be increasingly widely supported. One of the stated advantages of MTOM over the other attachments mechanisms is that it is compatible with the WS Security sets of standards.

The forthcoming OWS 1.2 specification [RD2] must also be taken into account. The version that is currently out for public comments specifies

1. the use of a "Manifest" for operations that require the return of multiple document to open operation request. "A manifest is a document describing the contents of a package of documents. A manifest can be used to quickly determine the contents of a package without having to scan the package contents. The specified Manifest data

structure lists and describes each document or resource bundled in a package. How the documents are packaged is irrelevant; for example, a package may be a zip file or a multi-part mime message. "This would mean that in addition to the EO ebRIM and the EO GML, a manifest needs to be added. OWS 1.2 further specifies that

- If the package is a list of files, there shall be one file named "ows-manifest.xml", whether the files are compressed in a ZIP or GZIP file, archived in a tar, or some other technique of packaging a list of files.
- If the manifest is in a package that doesn't store file names or is difficult to find resources based on a file name, such as a SOAP message with attachments or multipart mime message, then it is best to use an identifier to find the manifest. The identifier shall be a URN, namely "urn:ogc:def:documentType:OWS:1.2:Manifest.

2. For binary data send via a SOAP message, the Message Transportation Optimization Mechanism (MTOM) shall be used in conjunction with XML-binary Optimized Packaging (XOP). MTOM/XOP could be useful if one would for instance include the browse images directly in the Catalogue responses or if the EO GML would be binary encoded.

Based on the above, if a solution at message level would developed it would likely consist of grouping the GetRecords XML together with the individual GML instances within a zip archive that would be encorporated in the SOAP Message using XOP and transmitted using MTOM.

As the proposed solution "at content level" where the GML is included using extended wrs:ExtrinsicObjects, is however less of a drastic change for both client and server implementations, this solution is preferred.

2.3 REDEFINE THE BRIEF/SUMMARY/FULL DEFINITION OF THE GETRECORDS

Currently, here is the definition of the elementSetName parameter from the CSW-ebRIM specification.

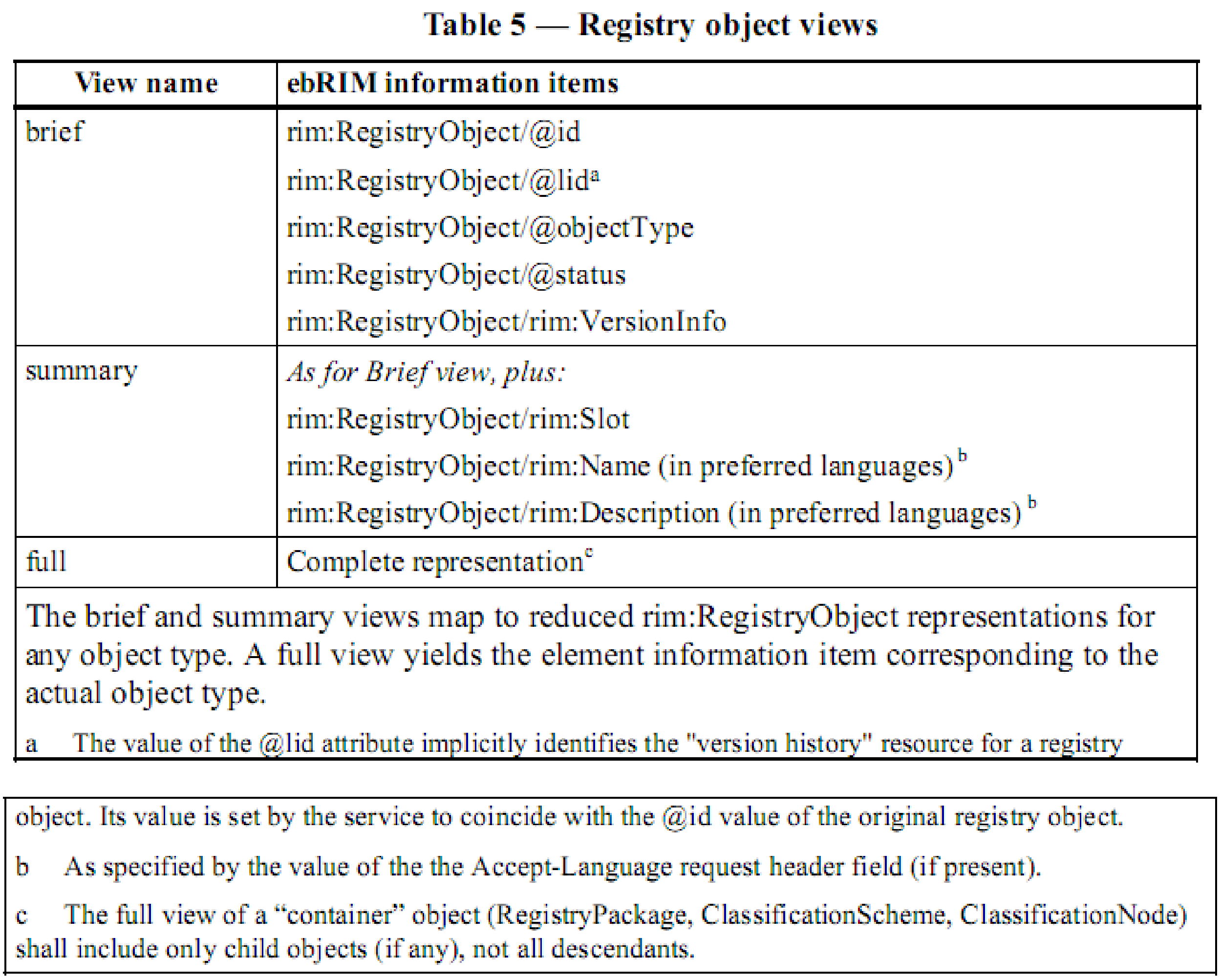


Table 5 — Registry object views

View name	ebRIM information items
brief	rim:RegistryObject/@id rim:RegistryObject/@lid[a] rim:RegistryObject/@objectType rim:RegistryObject/@status rim:RegistryObject/rim:VersionInfo
summary	*As for Brief view, plus:* rim:RegistryObject/rim:Slot rim:RegistryObject/rim:Name (in preferred languages) [b] rim:RegistryObject/rim:Description (in preferred languages) [b]
full	Complete representation[c]

The brief and summary views map to reduced rim:RegistryObject representations for any object type. A full view yields the element information item corresponding to the actual object type.

[a] The value of the @lid attribute implicitly identifies the "version history" resource for a registry object. Its value is set by the service to coincide with the @id value of the original registry object.

[b] As specified by the value of the the Accept-Language request header field (if present).

[c] The full view of a “container” object (RegistryPackage, ClassificationScheme, ClassificationNode) shall include only child objects (if any), not all descendants.

The difference between the summary and full view is small. Moreover, the summary view returns RegistryObject and not the native ebRIM object like ExtrinsicObject, Association, etc which is not useful. From our experience, the brief and full modes are used but the summary is not relevant.

We propose to redefine this as follow:

- Brief: not changed (id, lid, objectType, status and VersionInfo). The brief view map to reduced rim:RegistryObject representation for any object type.
- Summary: complete representation with the native ebRIM object.
- Full: idem as summary + the repository item encoded using XOP-MTOP or equivalent

This new definition allows to easily retrieving all the information about EO products in one GetRecords request.

2.4 HARMONIZATION WITH SENSORML

A new extension package for ebRIM (OGC 09-163 sensorML Extension Package for ebRIM) will be proposed as a discussion paper during the TC of Mountain View (December 2009). This document describes the discovery model for sensors & processes described in sensorML. It also describes the discovery model for EO profile of sensorML. The attributes currently stored in the EOAcquisistionPlatform are stored in this model in a SensorML platform.

2.5 STORED QUERIES DEFINITION

An interesting feature of the ebRIM model is the definition of Business queries. This allows to abstract the underlying model and to give an entry level access as a discovery interface.

2.5.1 Purpose

Extension packages may define various stored queries in order to provide a simple means of searching a registry. A stored query is essentially a parameterized, named query definition known to the hosting service. The query response shall include a csw:GetRecordsResponse element in the message body.

Using the POST method with a Content-Type of “application/xml”, a <rim:AdhocQuery> element may appear in a GetRecords request instead of a <csw:Query> element. The id attribute specifies the stored query to execute; any parameters are passed in as child Slot elements.

Example Invoking a stored query in a GetRecords context (POST method).

```
<csw:GetRecords service="CSW" version="2.0.2"
  resultType="results"
  outputSchema="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">
  <rim:AdhocQuery id="urn:ogc:def:ebRIM-Query:OGC:findServices">
    <rim:Slot name="serviceType">
      <rim:ValueList><rim:Value>WFS</rim:Value></rim:ValueList>
    </rim:Slot>
  </rim:AdhocQuery>
</csw:GetRecords>
```

When invoking a stored query using the GET method, the request is submitted to the GetRecords endpoint. A query string is appended to the request URI as follows: the value of the mandatory qid parameter specifies the stored query by identifier; additional parameters are appended as shown in the following example.

Example Invoking a stored query using the GET method.

```
http://host:port/path?request=Query&service=CSW-
ebRIM&qid=urn:ogc:def:ebRIM-Query:findServices
```

If there is no matching stored query, the response message must indicate a status code of 404 (Not Found); the body may also include an exception report with code “NotFound”. If any required query parameters are missing, then an exception with code “InvalidRequest” must be returned with status code 400.

2.5.2 Search Request

2.5.2.1 Purpose

The Search Request adhoc query allows returning of products via a search condition.

It is the equivalent of several standard queries in one procedure. The adhoc query simplifies the work on building the complex filter following OGC Filter standard.

2.5.2.2 Definition

Identifier	urn:ogc:def:ebRIM-EO-Query:HMA:searchEOProduct
Name	searchEOProduct
Parameters	*collectionIdentifier : string* identifier of on or several collections *geographicElement : geometry* a geometry expressed in GML to represent the geographic extension of the products *temporalElementStart : ISO8601* start time of the temporal coverage of the products *temporalElementEnd : ISO8601* end time of the temporal coverage of the products *platformName : string* platform name *instrumentName :string* instrument (sensor) name *cloudCoverPercentage : string* cloud coverage percentage
Response	A <csw:GetRecordsResponse> element containing a <rim:RegistryPackage> element corresponding to each EO Product matching the query recognized by the parameters.

2.5.2.3 Example of request

```
<csw:GetRecords service="CSW" version="2.0.2"
  resultType="results"
  outputSchema="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0">
  <rim:AdhocQuery id="urn:ogc:def:ebRIM-EO-Query:HMA:searchEOProduct">
    <rim:Slot name="collectionIdentifier">
      <rim:ValueList>
        <rim:Value>urn:x-EOP:ING:EOLI:ESA.EECF.ENVISAT_ASA_APx_xS</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="temporalElementStart">
      <rim:ValueList>
        <rim:Value>2002-10-10T17:00:00Z</rim:Value>
      </rim:ValueList>
    </rim:Slot>
    <rim:Slot name="temporalElementEnd">
      <rim:ValueList>
        <rim:Value>2002-10-10T19:00:00Z</rim:Value>
      </rim:ValueList>
    </rim:Slot>
  </rim:AdhocQuery>
</csw:GetRecords>
```

3 ANALYSIS OF DIFFERENT APPROACHES

3.1 INTRODUCTION

Within this section five alternative approaches are compared. The first approach is the one currently used. For each approach, the number of requests and the size of the information transferred to retrieve 100 EO Products is compared.

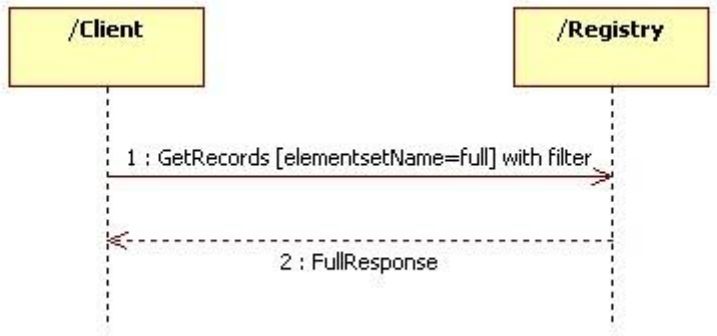
Estimations are based on the following values:

- EO GML : 6kb
- Full ebRIM representation - current model: 13kb
- Full ebRIM representation - simplified model: 6kb
- Single record brief ebRIM representation (ids) – simplified model: 1.5kb
- 100 records brief ebRIM representation (ids) –simplified model : 30Kb

3.2 GETRECORDS WITH REGISTRYPACKAGE

3.2.1 Workflow

This workflow is the one that is currently used in the EO EP of ebRIM CSW. The ExtrinsicObjects are grouped in a RegistryPackage to allow easier queries. This workflow does not use the GetRepositoryItem and thus, needs to store all the properties of the EO GML structure into the ebRIM.



3.2.2 Analysis

Number of request: 1 GetRecords

Size of response: 100 x 13kb = 1300 kb

No change required to OGC standards.

3.3 GETRECORDS AND GETREPOSITORYITEM

3.3.1 Workflow

This solution is based on the current status of the OGC ebRIM Application Profile of CS-W standard [AD03]. It uses the operations defined by this profile.to retrieve the ebRIM model (GetRecords) and the EO GML (GetRepositoryItem)

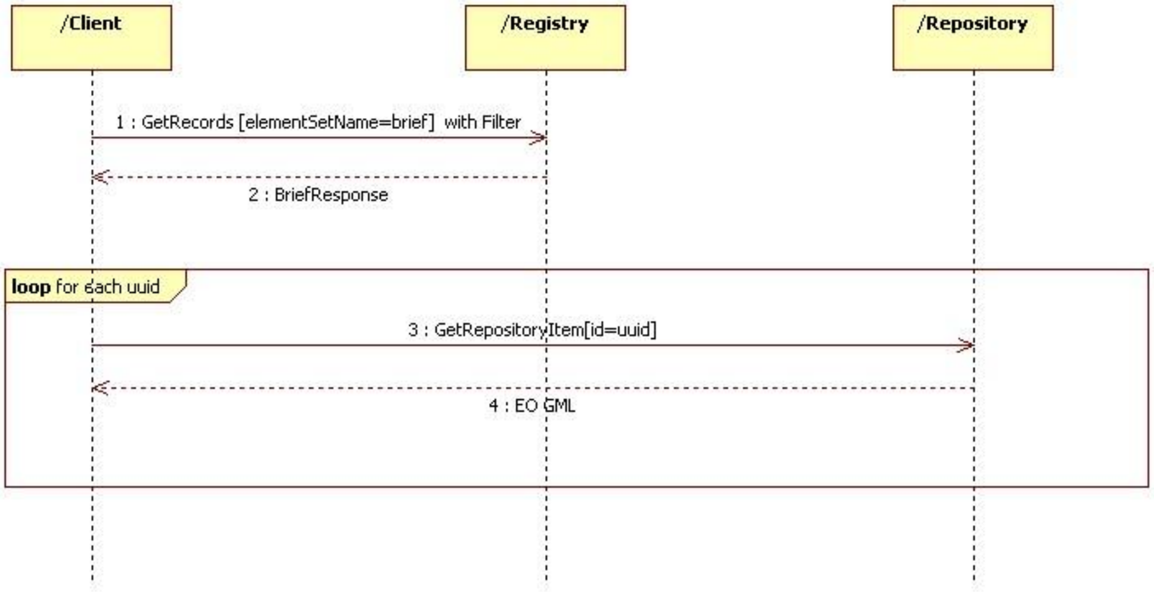

/Client
/Registry
/Repository
1 : GetRecords [elementSetName=brief] with Filter
2 : BriefResponse
loop for each uuid
3 : GetRepositoryItem[id=uuid]
4 : EO GML

Figure 3 : Sequence diagram for brief GetRecords and GetRepository Item (Solution 1)

The first request of this workflow is to query the ids of the EOProducts using the GetRecords operation using the elementSetName attributes to specify a *'brief'* encoding of the result (only *id* and *objectType* are returned). With these ids, a set of GetRepositoryItem operations are sent to retrieve the original EO GML file containing the entire metadata for each EO Product.

3.3.2 Analysis

Number of requests to display 100 results:

1 GetRecords

100 GetRepositoryItem

Total : 101 requests

1 GetRecords : 30kb

100 GetRepositoryItem :100 x 6kb = 600kb

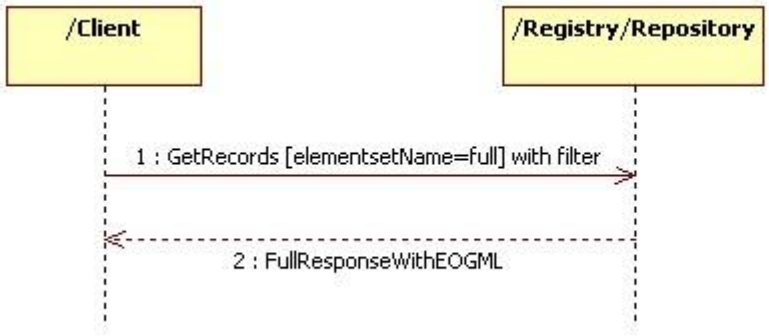
Total : 630Kb

No change required to OGC standards.

3.4 GETRECORDS WITH LIGHT MODEL & EO GML INCLUDED IN RESPONSE

3.4.1 Workflow

This workflow extends the current GetRecords operation to include the repositoryItem in the response when elementSetName is set to full. This requires an update to the OGC ebRIM Application Profile of CS-W standard [AD03].


/Client
/Registry/Repository
1 : GetRecords [elementsetName=full] with filter
2 : FullResponseWithEOGML

This solution would include the repositoryItem in the response of the GetRecords operation using the method proposed above using a wrs:RepositoryItemXML in the ebXML response, This would return in a pure XML manner the queryables attributes and the original metadata file in XML. As this is defined at the ebRIM AP level, this would benefit to other extension packages like CIM & sensorML.

3.4.2 Analysis

Number of requests to display 100 results: 1 GetRecords

1 GetRecords : 30kb + 100 x 6kb

Total : 630Kb

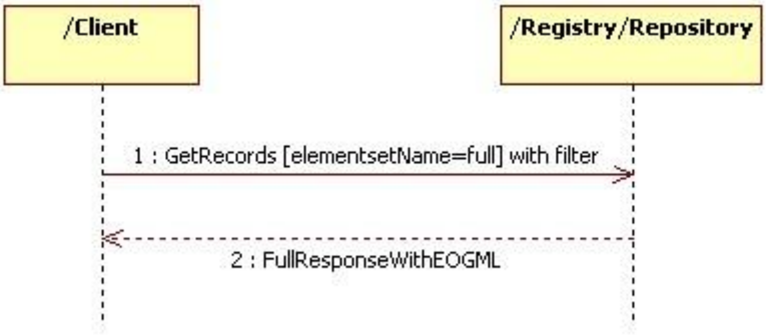
Change required to 07-110 ebRIM Application Profile of CS-W to change the structure of the GetRecords response

Change required to 06-131 EO Extension Package to simplify the model.

3.5 GETRECORDS WITH LIGHT MODEL & EO GML INCLUDED IN A SLOT

3.5.1 Workflow

To allow retrieval of the metadata in a single call, another solution would be to add a slot that would contain the xml content. Following the current version of the CS-W ebRIM Application Profile, all the slots are returned when you request elementSetName attribute to summary or full.


/Client
/Registry/Repository
1 : GetRecords [elementsetName=full] with filter
2 : FullResponseWithEOGML

3.5.2 Analysis

Number of requests to display 100 results: 1 GetRecords

1 GetRecords : 30kb + 100 x 6kb = 600kb

Total = 630Kb

Change required to 06-131 EO Extension Package to add a slot

3.5.3 Variant

A variant of this method could be done by defining in each extension package additional values for elementSetNameAttribute to define different views than the generic ones defined in the CS-W ebRIM Application Profile. This would keep an homogeneous result representation among different extension package, and allow a adequate level of detail defined for a specific model.

4 OGC SWG

4.1 CS-W 3.0 ROAD MAP

The CS-W SWG has been restarted beginning 2010 to update the OGC Catalog standard. The main updates are :

- Remove Corba Protocol Binding
- Update dependencies to OGC Common & Filter Encoding
- Include OpenSearch-Geo with feed (Atom/RSS) result set

CS-W will keep Dublin Core as metadata model.

4.2 EBRIM APPLICATION PROFILE OF CS-W UPDATE

For more than two years now, the CS-W ebRIM AP has been used by several initiatives, HMA being the most important one. With this experience, several CR have been submitted to OGC. To take these comments into account, a new SWG will be created at the next OGC Technical Commitee in March 2010. This group will analyze the CR posted and open a 30 days public comment period to allow other modification requests.

4.3 OGC CHANGE REQUESTS

4.3.1 09-133 CSW ebRIM Change GetCapabilities service parameter value

The value of the “service” parameter is not clearly defined. By default, the value is “CSW” inherited from the OGC 07-006r1. But the GetCapabilities is overridden with some specific CSW-ebRIM information so the “service” parameter value must be equal to “CSW-ebRIM”.

4.3.2 09-135 CSW ebRIM CR Encode metadata In GetRecordsResponse

Current specification does not allow retrieving the metadata in the GetRecords response. When a client wants to retrieve all the metadata matching some criteria, he has to perform a GetRecords and then a GetRepositoryItem for each result. This leads to a large number of request/responses that could be avoided by piggybacking the metadata in XML to the GetRecords response.

4.3.3 09-136 CSW ebRIM CR GetRepositoryItem Response In SOAP

Current specification explains how to support the SOAP 1.2 messaging framework but does not provide precise information for the GetRepositoryItem operation. This operation is different from the other since it returns the metadata that could be any text or binary format.

4.3.4 09-137 CSW ebRIM CR Improve ElementSetName Definition

Current specification does not allow retrieving the metadata in GetRecords response. When a client wants to retrieve all the metadata matching some criteria, he has to perform a GetRecords and then a GetRepositoryItem for each result. This leads to a large number of request/responses that could be avoided by piggybacking the metadata to the GetRecords response. In order to be able to return the repositoryItem, the typenames should be redefined.

5 SUMMARY

This document describes several problems that were raised during the HMA projects. These problems are in various Catalog standards. We have analyzed them and proposed different approaches to solve them, while keeping the specifications acceptable for people outside of the HMA project.

The most interesting approach is the one defined in section 3.4. This solution would simplify the ebRIM model by promoting as slots only the properties used for the discovery process and permit the transfer of the XML in the GetRecords Response without using binary transfer. These improvements require only smaller modifications in the different OGC specifications.

On the softwares side, the change would not be straightforward to implement and would not create any additional performance constraints or issues with firewalls or the adopted security standards.