Open Geospatial Consortium Inc.

Date: 2006-08-18

Reference number of this document: OGC® 06-131

Version: 0.0.3

Category: OGC® Discussion Paper

Editor(s): Renato Primavera

OGC® Catalogue Services Specification 2.0.0 (with Corrigendum)

EO Products Extension Package for ebRIM (ISO/TS 15000-3) Profile of CSW 2.0

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: OGC® Application Profile
Document subtype: Discussion Paper
Document stage: Draft
Document language: English

Contents

Page

Figures

Page

Tables

Page

Examples

i. Preface

This document describes the Data Model of Earth Observation Products for the OGC® Catalogue 2.0.0 (with Corrigendum) [OGC 04-021r3] implementing the OGC® ebRIM (ISO/TS 15000-3) Profile [OGC 05-025r3].

It defines the way HMA (Heterogeneous Earth Observation Missions Accessibility) resources (Earth Observation products metadata) are organized and implemented in the Catalogue for the discovery, retrieval and management.

ii. Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this specification.

iii. Submitting organizations

The following organizations submitted the original document to the OGC® Catalogue Services Specification 2.0 Revision Working Group.

- **IONIC Software s.a.**
- **Spacebel s.a.**

iv. Document contributor contact points

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

Name	Organization
Renato Primavera (Editor)	IONIC Software s.a.
Luc Donea (Reviewer)	IONIC Software s.a.

v. Revision history

Date	Release	Editor	Sections modified	Description
28/07/2006	0.1	J. Sonnet	Creation	Template adaptation, structure definition.
18/08/2006	0.2	R. Primavera	Add content	Chapter definition, content added.

11/09/2006	0.3	R. Primavera	Complete revision	All chapters revised and content added.

vi. Changes to the OGC Abstract Specification

The OpenGIS® Abstract Specification does not require changes to accommodate the technical contents of this document.

vii. Future work

At this stage, the current document is a work in progress.

viii. Foreword

This document has been created under the impulsion of European Space Agency in the scope of the Heterogeneous Missions Accessibility project. This document is built in conformance with the [OGC 06-080] GML Application Schema for EO Products document proposed by the European Space Agency, the French Space Agency, the European Satellite Center, Spacebel s.a. and Spot Image in an early phase of the HMA Project.

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

Introduction

This specification is part of a set that describes services for managing Earth Observation (EO) data products. The services include collection level and product level catalogues, online-ordering for existing and future products, online access, etc. These services are put into context in an overall document [NR3 - Best Practices for EO Products].

The services described in this document are intended to support the identification of (EO) data products from previously identified data collections and therefore the search and presentation of metadata from catalogues of EO data products through standard compliant operations. This interface could be supported by many data providers (satellite operators, data distributors …), most of whom have existing (and relatively complex) facilities for the management of these data (including SOAP/WSDL technologies).

EO data product collections are usually structured to describe data products derived from a single sensor onboard a satellite or series of satellites. Products from different classes of sensors usually require specific product metadata. The following classes of products have been identified so far: radar, optical, atmospheric. The proposed approach is to identify a common set of elements grouped in a common (HMA) schema and extend this common schema to add sensors specific metadata.

1 Scope

This OGC™ document specifies the EO Product Data Model for ebRIM (ISO/TS 15000-3) Profile of CSW 2.0, based on the [OGC 06-080] OGC™ GML Application Schema for EO Products, in order to allow the creation of catalogues for EO Products based on the [OGC 05-025r3] OGC™ ebRIM Profile of CSW.

2 Compliance

Compliance with this specification shall be checked using all the relevant tests specified in Annex A (normative).

3 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

[ebRIM] *OASIS ebXML Registry Information Model Version 3.0*

[ISO 19105:2000] *Geographic information : Conformance and Testing*

[ISO 19115:2003] *Geographic information : Metadata*

[OGC 04-021r3] *OGC™ Catalogue Services Specification 2.0.0 (with Corrigendum)*

[OGC 05-008] *OGC™ Web Services Common Specification Corrigendum*

[OGC 05-025r3] *OGC™ ebRIM (ISO/TS 15000-3) profile of CSW*

[OGC 06-080] *OGC™ GML Application Schema for EO Products*

In addition to this document, this specification includes several normative XML Schema Document files as specified in Annex B.

4 Terms and definitions

For the purposes of this specification, the definitions specified in Clause 4 of the [OGC 05-008] OGC™ OWS Common Implementation Specification shall apply. In addition, the following terms and definitions apply.

4.1
application profile
set of one or more base standards and – where applicable – the identification of chosen clauses, classes, subsets, options and parameters of those base standards that are necessary for accomplishing a particular function [ISO 19101, ISO 19106]

4.2
client
software component that can invoke an **operation** from a **server**

4.3
data level
stratum within a set of layered levels in which data is recorded that conforms to definitions of types found at the application model level [ISO 19101]

4.4
dataset series (dataset collection[1])
collection of datasets sharing the same product specification [ISO 19113, ISO 19114, ISO 19115]. In the earth observation context, a collection typically corresponds to datasets (i.e. products) derived from data acquired by a single sensor onboard a satellite or series of satellites.

4.5
datastrip
a satellite acquisition

4.6
geographic dataset
dataset with a spatial aspect [ISO 19115]

[1] Due to historical reasons we´ll mainly use the term ‘dataset collection’ in this document although the term ‘dataset series’ is used in the ISO/TC211 Terminology Maintenance Group.

4.7
geographic information
information concerning phenomena implicitly or explicitly associated with a location relative to the Earth [ISO 19128 draft]

4.8
georesource
geographic information of a specific type (e.g. geographic dataset, geographic application, geographic service)

4.9
identifier
a character string that may be composed of numbers and characters that is exchanged between the client and the server with respect to a specific identity of a resource

4.10
interface
named set of operations that characterise the behaviour of an entity [ISO 19119]

4.11
metadata dataset (metadataset)
metadata describing a specific dataset [ISO 19101]

4.12
metadata entity
group of metadata elements and other metadata entities describing the same aspect of data

NOTE 1 A metadata entity may contain one or more metadata entities.

NOTE 2 A metadata entity is equivalent to a class in UML terminology [ISO 19115].

4.13
metadata schema
conceptual schema describing metadata

NOTE ISO 19115 describes a standard for a metadata schema. [ISO 19101]

4.14
metadata section
subset of metadata that defines a collection of related metadata entities and elements [ISO 19115]

4.15
operation
specification of a transformation or query that an object may be called to execute [ISO 19119]

4.16
parameter
variable whose name and value are included in an operation **request** or **response**

4.17
qualified name
name that is prefixed with its naming context

4.18
request
invocation of an **operation** by a **client**

4.19
response
result of an **operation,** returned from a **server** to a **client**

4.20
schema
formal description of a model [ISO 19101, ISO 19103, ISO 19109, ISO 19118]

4.21
server
service instance
a particular instance of a **service** [ISO 19119]

4.22
service
distinct part of the functionality that is provided by an entity through interfaces [ISO 19119]

capability which a service provider entity makes available to a service user entity at the interface between those entities [ISO 19104 terms repository]

4.23
service interface
shared boundary between an automated system or human being and another automated system or human being [ISO 19101]

4.24
service metadata
metadata describing the **operations** and **geographic information** available at a **server** [ISO 19128 draft]

4.25
state
condition that persists for a period

NOTE The value of a particular feature attribute describes a condition of the feature [ISO 19108].

4.26
transfer protocol
common set of rules for defining interactions between distributed systems [ISO 19118]

4.27
version
version of an Implementation Specification (document) and XML Schemas to which the requested operation conforms

NOTE An OWS Implementation Specification version may specify XML Schemas against which an XML encoded operation request or response must conform and should be validated.

5 Conventions

5.1 Abbreviated terms

Some more frequently used abbreviated terms:

API	Application Program Interface
ATM	Atmospheric
BPEL	BusinessProcess Execution Language
COTS	Commercial Off The Shelf
CQL	Common Query Language
CRS	Coordinate Reference System
CSW	Catalogue Service for Web
DCE	Distributed Computing Platform
DC	Dublin Core
DCMI	Dublin Core Metadata Initiative
DCP	Distributed Computing Platform
EBRIM	ebXML Registry Information Model
EO	Earth Observation
GML	Geography Markup Language
HMA	Heterogeneous Missions Accessibility
HTTP	HyperText Transport Protocol
ISO	International Organisation for Standardisation
OGC	Open Geospatial Consortium
OHR	Optical High Resolution
PHR	Pleiades High Resolution
SAR	Synthetic Aperture Radar
SOAP	Simple Object Access Portocol
SQL	Structured Query Language
UML	Unified Modeling Language
URI	Uniform Resource Identifier

URL	Uniform Resource Locator
URN	Uniform Resource Name
UTF-8	Unicode Transformation Format-8
WSDL	Web Service Definition Language
W3C	World Wide Web Consortium
XML	eXtensible Markup language

5.2 UML notation

Some of the diagrams in this document are presented using the Unified Modeling Language (UML) static structure diagram. The UML notations used in this document are described in Figure 1, below.

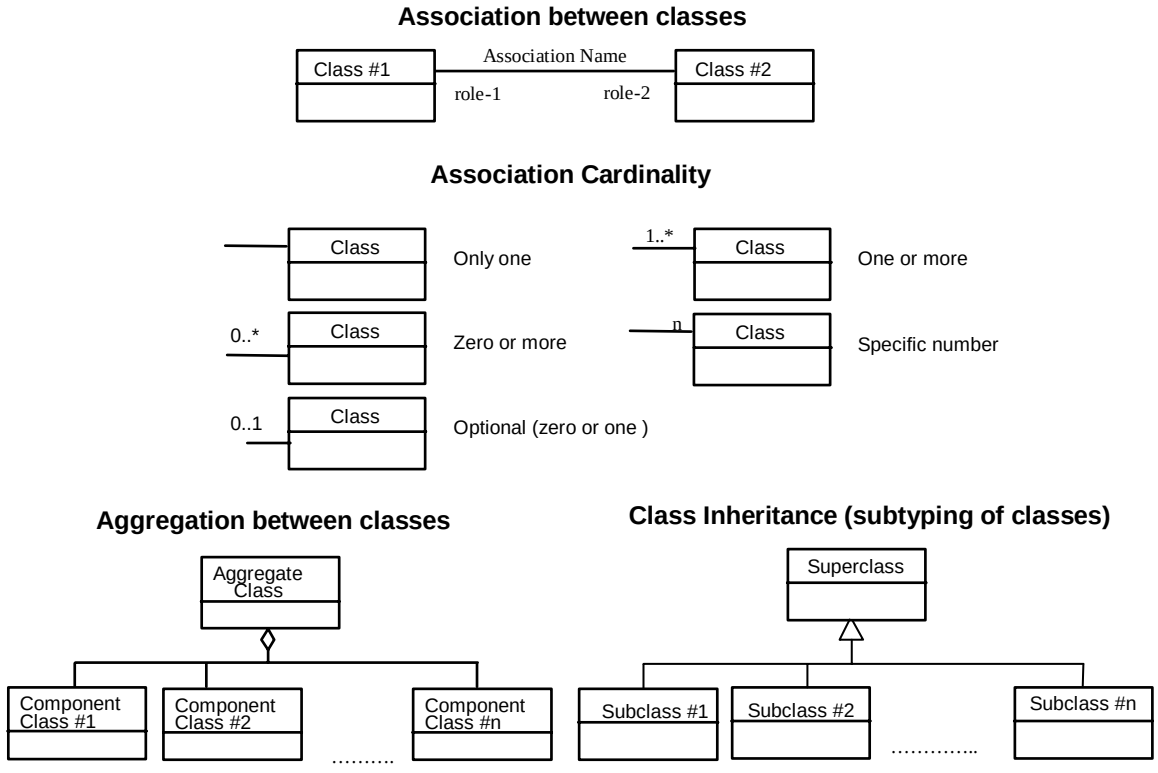

Figure 1: UML notations

In these UML class diagrams, the class boxes with a light background are the primary classes being shown in this diagram, often the classes from one UML package. The class boxes with a gray background are other classes used by these primary classes, usually classes from other packages.

In this diagram, the following stereotypes of UML classes are used:

<<Interface>> A definition of a set of operations that is supported by objects having this interface. An Interface class cannot contain any attributes.

<<Type>> A stereotyped class used for specification of a domain of instances (objects), together with the operations applicable to the objects. A Type class may have attributes and associations.

<<DataType>> A descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A DataType is a class with no operations whose primary purpose is to hold the information.

<<CodeList>> A flexible enumeration that uses string values for expressing a list of potential values. If the list alternatives are completely known, an enumeration shall be used; if the only likely alternatives are known, a code list shall be used.

<<Enumeration>> A data type whose instances form a list of alternative literal values. Enumeration means a short list of well-understood potential values within a class.

In this document, the following standard data types are used:

CharacterString – A sequence of characters

Boolean – A value specifying TRUE or FALSE

Integer – An integer number

Identifier – Unique identifier of an object

URI – An identifier of a resource that provides more information

URL – An identifier of an on-line resource that can be electronically accessed

5.3 Used parts of other documents

This document uses significant parts of document [OGC 06-080]. To reduce the need to refer to that document, this document copies some of those parts with small modifications. To indicate those parts to readers of this document, the largely copied parts are shown with a light grey background (15%).

6 Catalogue Infrastructure Overview

Catalogues are intended to store metadata describing resources published by providers and allow clients to find these resources. These resources metadata are organized in Catalogues according to specific data models, based on registry information model.

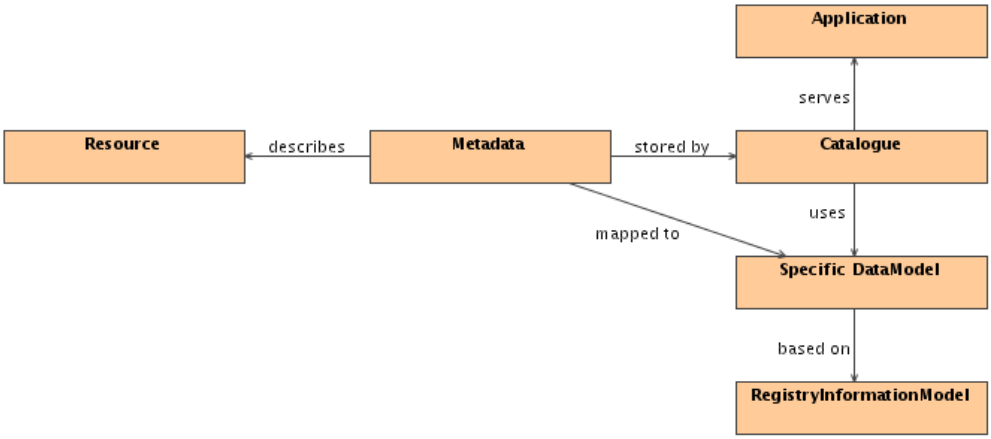

Figure 2: General Catalogue Infrastructure

This schema clearly describes the following:

- *Resources* are described using *Metadata*,
- *Metadata* are stored in *Catalogues*, according to a *DataModel* defining accurately the mapping of such type of resources (*DataModel* is resource-specific),
- *DataModel* is based on a generic model, called the *RegistryInformationModel* (aka *RIM*). The *RegistryInformationModel* is common to all resources within a catalogue,
- *Applications* use *Catalogues* to discover resources through their metadata.

In the Earth Observation context:

- *Resources* are *Earth Observation Products* (aka *EOProducts*),
- *Metadata* (describing *EOProducts*) are encoded into GML documents in conformance with [OGC 06-080],
- *Metadata* are stored in *Catalogues* according to the *EOProducts DataModel*
- The *EOProducts DataModel* is based on the *ebRIM* (a *RegistryInformationModel* used in an *Application Profile* of CSW 2.0 compliant catalogues),

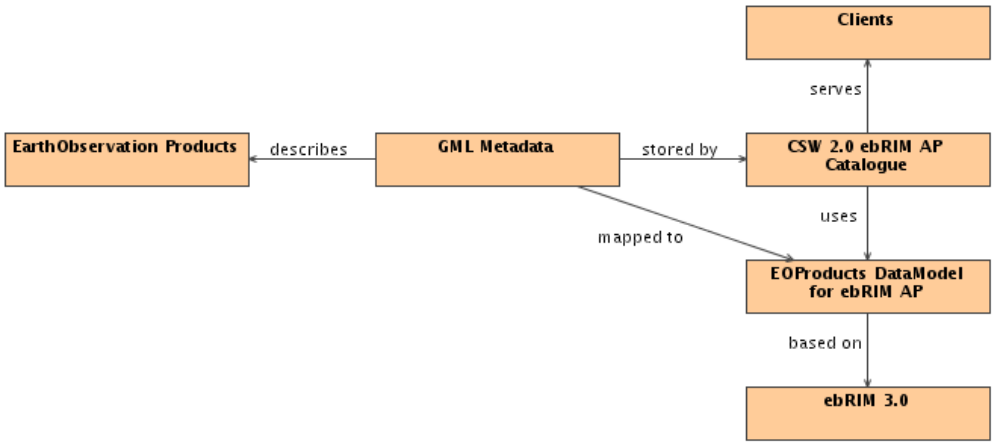

Figure 3: Earth Observation Oriented Catalogue Infrastructure

Specifications define:

- The way to interact with Catalogues through operations on the service (a Web Service in this case), to publish and query data,
- The way to use a registry information model (in this case ebRIM) to allow mapping of resources in Catalogues,
- The way to map each kind of metadata in the RIM (this way is called the data model). A specific kind of resource (e.g., an Earth Observation Product) shall be modeled using the same set of ebRIM objects in all Catalogues. Indeed, discover queries are strongly dependant of the chosen mapping. A single and common (standardized) mapping ensures interoperability between Catalogue implementations.

This document defines the data model of Earth Observation Products, without requiring any modification or extension either in the [OGC 04-021r3] OGC™ Catalogue Services Specification 2.0.0 (with Corrigendum) or in the [OGC 05-025r3] OGC™ ebRIM (ISO/TS 15000-3) profile of CSW

7 EO Products Data Model

Two levels of metadata describe the EO Products, the collection level (i.e., *dataset collection*) and the product level (i.e., *dataset*).

Collection level metadata are defined using the [ISO 19115:2003] ISO/TC211 Metadata Standards. Product level metadata are defined using the [OGC 06-080] OGC™ GML Application Schema for EO Products.

This document defines how these two kinds of metadata can be registered smoothly into CS-W Catalogues implementing the ebRIM Application Profile. It also defines the set of fields that are available for efficient discovery.

Table 1 — EOProduct Metadata Levels

Information resource	Description
EO Product Collection	Mapped to *dataset collection* in OGC Terminology. Set of metadata that describes an EO Product Collection.
EO Product	Mapped to *dataset* in OGC Terminology. Set of metadata that describes an EO Product.

These two levels are interrelated according the following schema:

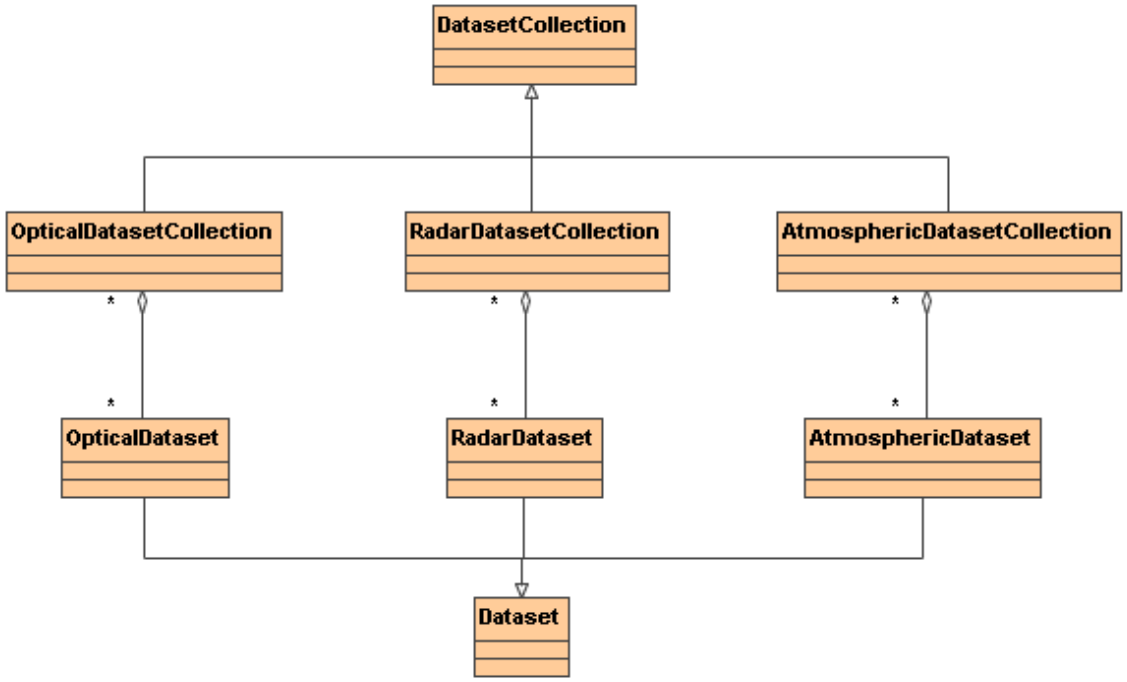

Figure 4: Relationship between EO dataset collections and datasets

7.1 EO Products Collection Mapping

[TBD]

7.2 EO Products Metadata Mapping

7.2.1 Earth Observation Product Types

The following picture describes the layered structure of the XML-Schemas used to define the different classes of product metadata. The layer structure means that the upper layer main element type is defined by extending a type from the lower level schema.

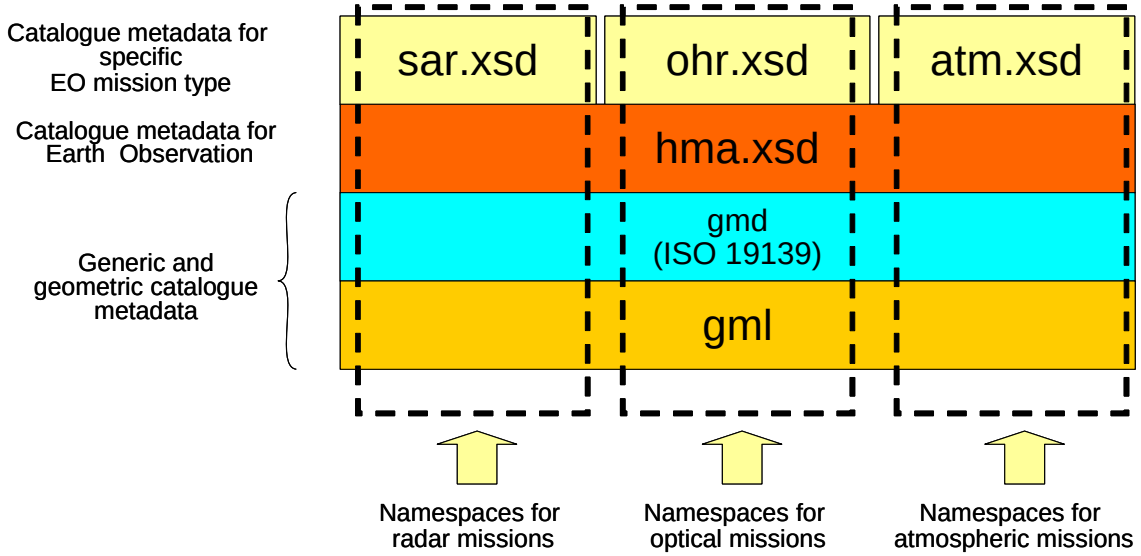

Figure 5: EO Product Level Metadata Structure

The element that describes the EO metadata is the "EarthObservationProduct" element which is defined in the SAR (Synthetic Aperture Radar), OHR (Optical High Resolution), ATM (Atmospheric) specific schemas as an extension of a common EarthObservationProduct element defined in the HMA schema.

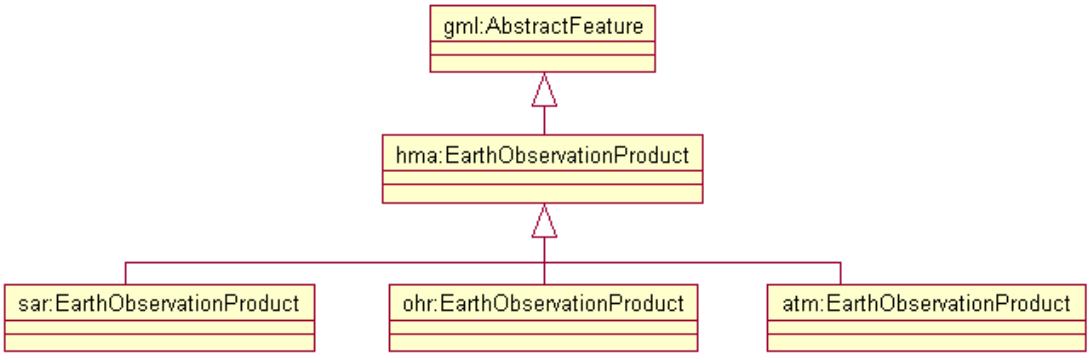

Figure 6: HMA Information Model

EO Products are described using one instance of the following types (or possibly subtypes, more specific) deriving from the hma:EarthObservationProduct,

- sar:EarthObservationProduct
- ohr:EarthObservationProduct
- atm:EarthObservationProduct

7.2.2 Mapping Schema Structure

The following sections will describe the ebRIM schema needed to map the Earth Observation metadata onto the eBusiness Registry Information Model (ebRIM). That schema defines the Catalogue discovery abilities and efficiency.

The EO schema is composed of three parts:

- the main container to store the 'acquisition' metadata,
- the EO Product Types taxonomy (hierarchical) to classify the main container
- multiple associated other metadata containers

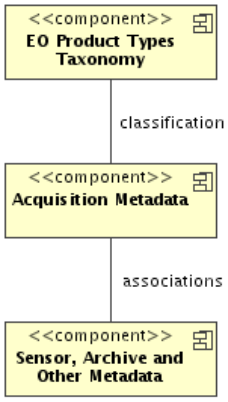

Figure 7: EO Products Mapping Schema Structure

7.2.3 Acquisition Parameters

In an ebRIM Catalogue, each EO Product instance is represented by an *ExtrinsicObject* with the '*EOProduct*' *objectType* attribute. This *ExtrinsicObject* is the main object of the EO Product mapping schema. It contains a set of attributes, matching the queryable metadata coming from the GML document. These attributes characterize directly the product acquisition.

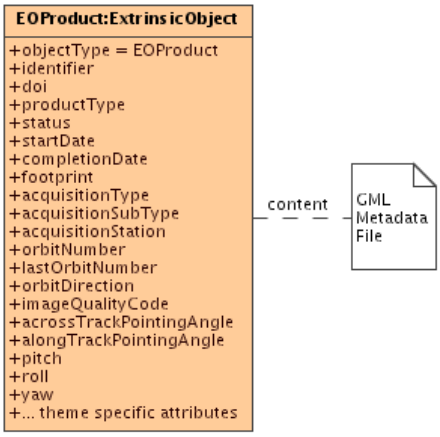

Figure 8: EO Products instances

The GML metadata file is linked to the main *ExtrinsicObject* as *content*. This can be done either internally if the Catalogue is also a repository (the GML file is stored in the Catalogue), or externally if the Catalogue is only a registry (the GML file is linked, or generated from available metadata). These implementation details are explained in depth in the Annex C.

All representative acquisition parameters are available (as slots) for extended search. It allows queries like:

- "*Give me all EO Product instances intersecting this footprint, measured after this date and shoot with this Along-Track Pointing Angle*".

More formally:

- "*Give me all ExtrinsicObjects with the objectType equals to 'EOProduct', the footprint slot value – a geometry – intersecting this box, the startDate slot value greater than this date and the alongTrackPointingAngle slot value equals to this*".

Every EO Product instance will have the general hma fields in common, plus their theme and mission-specific fields. The main *ExtrinsicObject* enables searches on the EO Products **acquisition parameters**.

Following table details the use of predefined attributes and slots, in order to map information coming from the GML metadata file to the *ExtrinsicObject* representing the EOProduct.

Table 2 — EOProduct ExtrinsicObject Correspondance

GML Metadata XPath	EOProduct ExtrinsicObject Attribute
"EOProduct" (fixed value)	/ExtrinsicObject/@objectType
/hma:EarthObservationProduct/hma:identifier	/ExtrinsicObject[@objectType="EOProduct"]/Name/LocalizedString/Value

GML Metadata XPath	EOProduct ExtrinsicObject Attribute
/hma:EarthObservationProduct/hma:doi	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”Doi”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:productType	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”ProductType”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:status	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”ProductStatus”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:extentOf	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”Footprint”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionType	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionType”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionSubType	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionSubType”]/ValueList/Value[*]
/hma:EarthObservationProduct/hma:startDate	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionStartDate”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:completionDate	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionCompletionDate”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionStation	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionStation”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionParameters/hma:Acquisition/hma:orbitNumber	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionOrbitNumber”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionParameters/hma:Acquisition/hma:lastOrbitNumber	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionLastOrbitNumber”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionParameters/hma:Acquisition/hma:orbitDirection	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionOrbitDirection”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionParameters/hma:Acquisition/hma:acrossTrackPointingAngle	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionAcrossTrackPointingAngle”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionParameters/hma:Acquisition/hma:alongTrackPointingAngle	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionAlongTrackPointingAngle”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionParameters/hma:Acquisition/hma:pitch	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionPitch”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionParameters/hma:Acquisition/hma:Yaw	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionYaw”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquisitionParameters/hma:Acquisition/hma:roll	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionRoll”]/ValueList/Value[1]

GML Metadata XPath	EOProduct ExtrinsicObject Attribute
/hma:EarthObservationProduct/hma:imageQualityCode	/ExtrinsicObject[@objectType="EOProduct"]/Slot[@name="ImageQualityCode"]/ValueList/Value[1]

7.2.4 Earth Observation Taxonomy

A taxonomy is needed to distinguish EOProduct types in the ebRIM Catalogue (at the thematic or mission-specific levels). That taxonomy is modeled using a *ClassificationScheme* object and a hierarchy of *ClassificationNodes*:

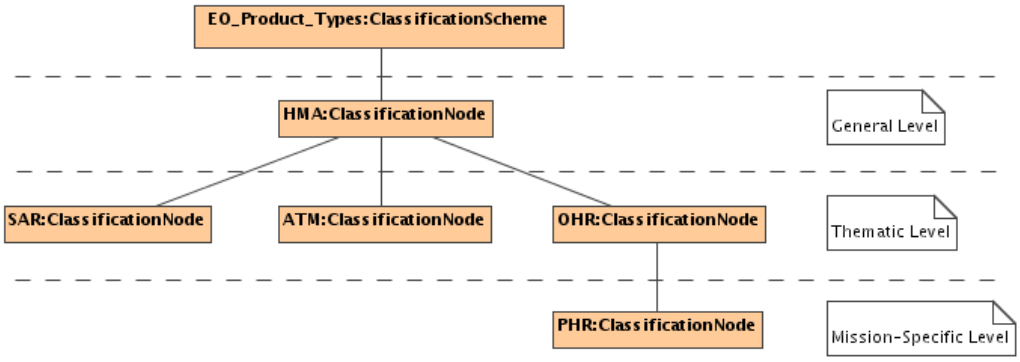

Figure 9: EO Product Types Taxonomy

Each EO Product is classified in the taxonomy. The EO Product *ExtrinsicObject* is linked to its corresponding *ClassificationNode* through a *Classification* object. This allows EO Products discovery by **acquisition type**.

Such relationship allows queries like:

- "*Give me all EO Products instances of type PHR*"

More formally:

- "*Give me all objects linked to the ClassificationNode PHR through a Classification*".

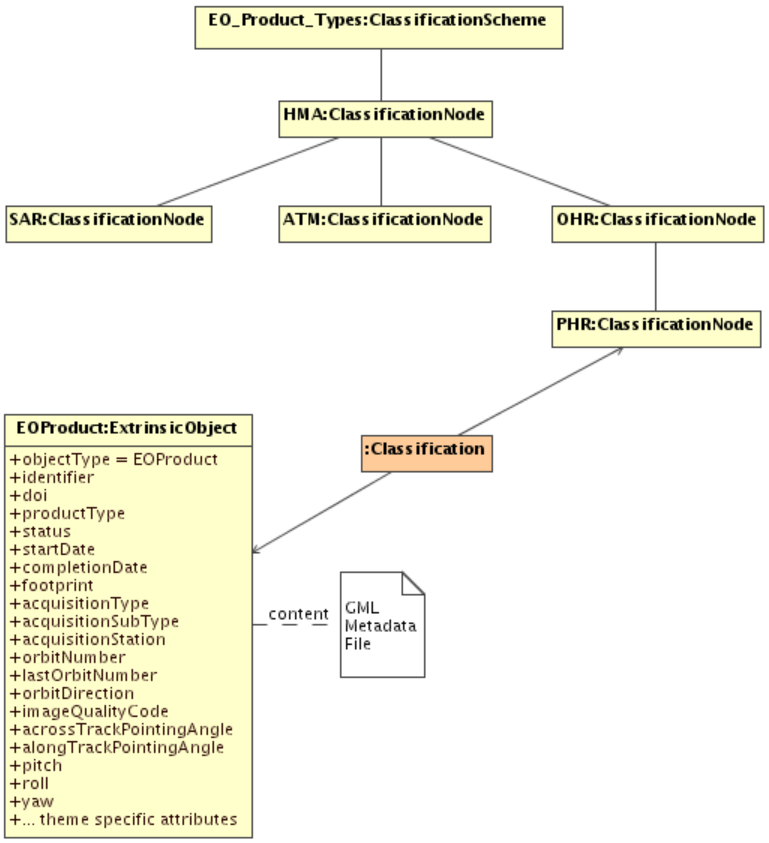

Figure 10: EO Product classified

7.2.5 Platform, Product and Archiving Metadata

Additional information is linked to the main EO Product *ExtrinsicObject*. These additional metadata are stored into specific *ExtrinsicObjects*, linked to the main one using *Associations*.

The Acquisition Platform parameters (i.e., Platform, Instrument and Sensor) are stored in an *ExtrinsicObject* having the *objectType* attribute set to '*EOAcquisitionPlatform*'. It is linked to the main *ExtrinsicObject* through an *Association* object, with the *associationType* attribute equals to '*acquiredBy*'.

An acquisition Platform metadata set will be common to multiple acquisitions, defining therefore a n:1 association from the EO Product *ExtrinsicObject* to the matching EO Acquisition Platform *ExtrinsicObject.*

Such information ensures EO Products search by **acquisition platform**, and provides a way to do queries like:

- "*Give me all EO Product acquired by this kind of Sensor*"

More formally:

- "*Give me all ExtrinsicObjects with objectType equals to 'EOProduct', which are source of an Association of type 'acquiredBy' pointing – as target – to an ExtrinsicObject having as objectType 'EOAcquisitionPlatform' and having this as sensorType slot value*".

Objects storing product information are also associated to the EOProduct. These *ExtrinsicObjects* (with *objectType* attribute equals to '*EOProductInformation*') are linked to the EO Product through *Associations* having the *associationType* attribute equals to '*hasProductInformation*'.

BrowseInformation and MaskInformation are not mapped to any ebRIM object, since they are not considered as queryable properties. Nevertheless, that information is still available through the link to the EO GML Metadata file, moreover, searches are always possible (but not optimized) using XPath queries.

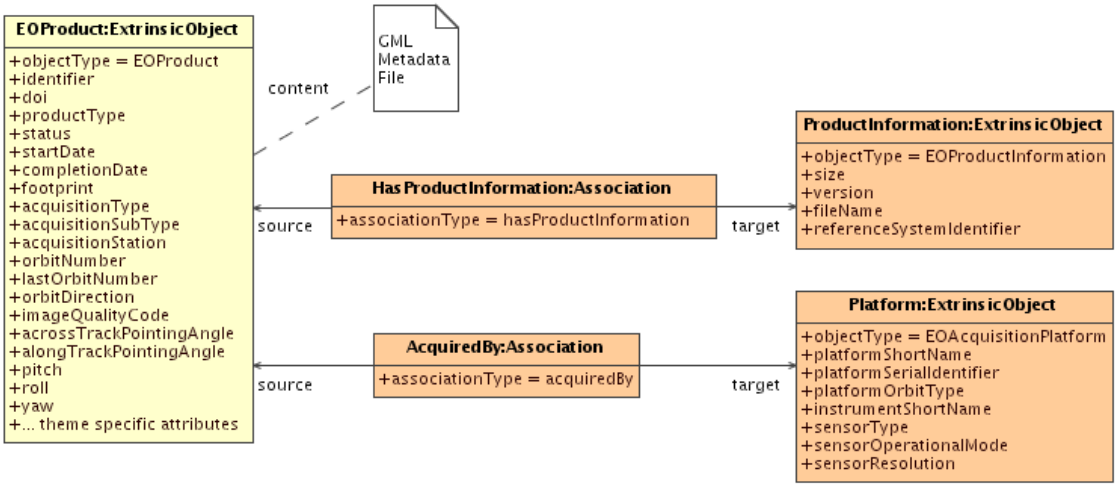

Figure 11: EO Products additional information

Following tables define the use of predefined attributes and slots, in order to map information coming from the GML metadata file to these additional *ExtrinsicObjects*.

Table 3 — EOAcquisitionPlatform ExtrinsicObject Correspondance

GML Metadata XPath	MaskInformation ExternalLink Attribute
"EOAcquisitionPlatform" (fixed value)	/ExtrinsicObject/@objectType
/hma:EarthObservationProduct/hma:acquiredBy/hma:Platform/hma:shortName	/ExtrinsicObject[@objectType="EOAcquisitionPlatform"]/Slot[@name="PlatformShortName"]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquiredBy/hma:Platform/hma:serialIdentifier	/ExtrinsicObject[@objectType="EOAcquisitionPlatform"]/Slot[@name="PlatformSerialIdentifier"]/ValueList/Value[1]

GML Metadata XPath	MaskInformation ExternalLink Attribute
/hma:EarthObservationProduct/hma:acquiredBy/hma:Platform/hma:orbitType	/ExtrinsicObject[@objectType="EOAcquisitionPlatform"]/Slot[@name="PlatformOrbitType"]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquiredBy/hma:Instrument/hma:shortName	/ExtrinsicObject[@objectType="EOAcquisitionPlatform"]/Slot[@name="InstrumentShortName"]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquiredBy/hma:Sensor/hma:sensorType	/ExtrinsicObject[@objectType="EOAcquisitionPlatform"]/Name/LocalizedString/Value
/hma:EarthObservationProduct/hma:acquiredBy/hma:Sensor /hma:operationalMode	/ExtrinsicObject[@objectType="EOAcquisitionPlatform"]/Slot[@name="SensorOperationalMode"]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:acquiredBy/hma:Sensor /hma:resolution	/ExtrinsicObject[@objectType="EOAcquisitionPlatform"]/Slot[@name="SensorResolution"]/ValueList/Value[1]

Table 4 — ProductInformation ExtrinsicObject Correspondance

GML Metadata XPath	MaskInformation ExternalLink Attribute
"EOProductInformation" (Fixed value)	/ExternalObject/@objectType
/hma:EarthObservationProduct/hma:product/hma:ProductInformation/hma:referenceSystemIdentifier	/ExtrinsicObject[@objectType="EOProductInformation"]/Slot[@name="ReferenceSystemIdentifier"]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:product/hma:ProductInformation/hma:size	/ExtrinsicObject[@objectType="EOProductInformation"]/Slot[@name="Size"]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:product/hma:ProductInformation/hma:version	/ExtrinsicObject[@objectType="EOProductInformation"]/Slot[@name="Version"]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:product/hma:ProductInformation/hma:fileName	/ExtrinsicObject[@objectType="EOProductInformation"]/Slot[@name="FileName"]/ValueList/Value[1]

Finally, an object modeling the archiving information is linked to the EOProduct *ExtrinsicObject* through an *Association* with *associationType* attribute equals to '*archivedIn*'. This *Association* has the EOProduct *ExtrinsicObject* as *sourceObject* and the EOArchivingInformation *ExtrinsicObject* as *targetObject* :

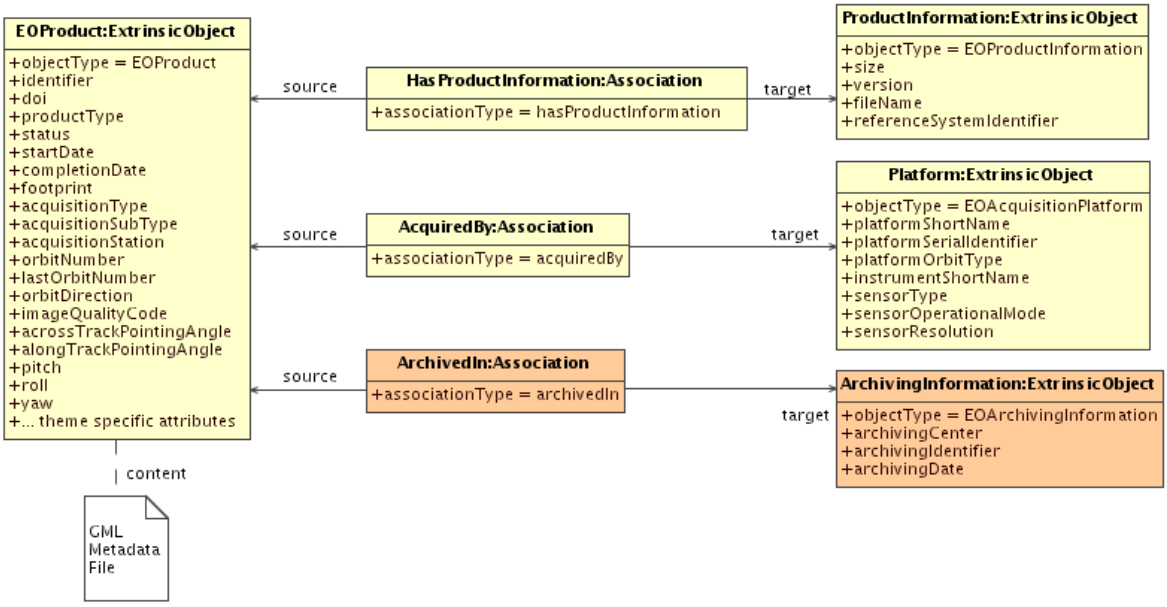

Figure 12: EO Products additional information

Table 5 — EOArchivingInformation ExtrinsicObject Correspondance

GML Metadata XPath	MaskInformation ExternalLink Attribute
“EOArchivingInformation” (fixed value)	/ExtrinsicObject/@objectType
/hma:EarthObservationProduct/hma:archivedIn/hma:ArchivingInformation/hma:archivingCenter	/ExtrinsicObject[@objectType=”EOArchivingInformation”]/Name/LocalizedString/Value
/hma:EarthObservationProduct/hma:archivedIn/hma:ArchivingInformation/hma:archivingIdentifier	/ExtrinsicObject[@objectType=”EOArchivingInformation”]/Slot[@name=”ArchivingIdentifier”]/ValueList/Value[1]
/hma:EarthObservationProduct/hma:archivedIn/hma:ArchivingInformation/hma:archivingDate	/ExtrinsicObject[@objectType=”EOArchivingInformation”]/Slot[@name=”ArchivingDate”]/ValueList/Value[1]

7.2.6 The Complete Mapping Schema

Here is what the complete schema looks like :

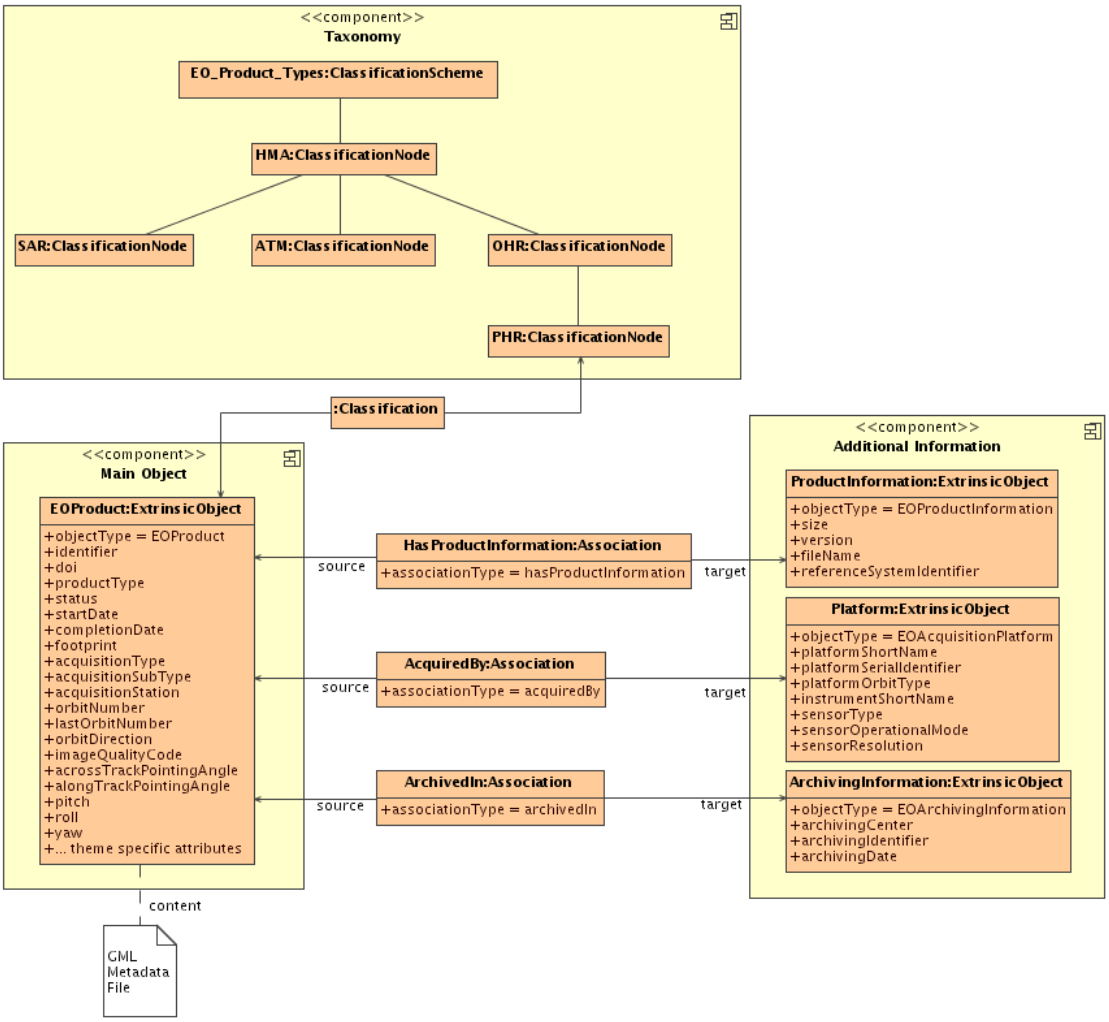

Figure 13: Complete EO Products Data Model

7.2.7 Thematic and Mission-Specific Metadata

Next sections describe the thematic-specific metadata and their mapping in the complete structure described earlier.

7.2.7.1 Synthetic Aperture Radar EarthObservationProduct Mapping

Following table defines mapping for SAR specific additional information.

Table 6 — EOProduct ExtrinsicObject Correspondance

GML Metadata XPath	EOProduct ExtrinsicObject Attribute
/sar:EarthObservationProduct/sar:polarisationMode	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionPolarisationMode”]/ValueList/Value[1]
/sar:EarthObservationProduct/sar:polarisationChannels	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionPolarisationChannels”]/ValueList/Value[1]
/sar:EarthObservationProduct/sar:antennaLookDirection	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionAntennaLookDirection”]/ValueList/Value[1]
/sar:EarthObservationProduct/sar:minimumIncidenceAngle	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionMinimumIncidenceAngle”]/ValueList/Value[1]
/sar:EarthObservationProduct/sar:maximumIncidenceAngle	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionMaximumIncidenceAngle”]/ValueList/Value[1]
/sar:EarthObservationProduct/sar:dopplerFrequency	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionDopplerFrequency”]/ValueList/Value[1]
/sar:EarthObservationProduct/sar:beam	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionBeam”]/ValueList/Value[1]

Table 7 — EOAcquisitionPlatform ExtrinsicObject Correspondance

GML Metadata XPath	MaskInformation ExternalLink Attribute
/hma:EarthObservationProduct/hma:acquiredBy/hma:Sensor/hma:swathIdentifier	/ExtrinsicObject[@objectType=”EOAcquisitionPlatform”]/Slot[@name=”SensorSwathIdentifier”]/ValueList/Value[1]

7.2.7.2 Optical High-Resolution EarthObservationProduct Mapping

Following table defines mapping for OHR specific metadata.

Table 8 — EOProduct ExtrinsicObject Correspondance

GML Metadata XPath	EOProduct ExtrinsicObject Attribute
/ohr:EarthObservationProduct/ohr:AcquisitionParameters/ohr:Acquisition/ohr:illuminationAzimuthAngle	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionIlluminationAzimuthAngle”]/ValueList/Value[1]
/ohr:EarthObservationProduct/ohr:AcquisitionParameters/ohr:Acquisition/ohr:illuminationElevationAngle	/ExtrinsicObject[@objectType=”EOProduct”]/Slot[@name=”AcquisitionIlluminationElevationAngle”]/ValueList/Value[1]

GML Metadata XPath	EOProduct ExtrinsicObject Attribute
/ohr:EarthObservationProduct/ohr:cloudCoverPercentage	/ExtrinsicObject[@objectType="EOProduct"]/Slot[@name="AcquisitionCloudCoverPercentage"]/ValueList/Value[1]
/ohr:EarthObservationProduct/ohr:snowCoverPercentage	/ExtrinsicObject[@objectType="EOProduct"]/Slot[@name="AcquisitionSnowCoverPercentage"]/ValueList/Value[1]

7.2.7.3 Atmospheric EarthObservationProduct Mapping

ATM defines several Data Layers by EO Product. These Data Layers are modeled in an ebRIM Catalogue using *ExtrinsicObjects* associated to the EOProduct *ExtrinsicObject* through an *Association* with *associationType* attribute equals to '*hasDataLayer*'.

Following table defines mapping for ATM specific metadata.

Table 9 — EODataLayer ExtrinsicObject Correspondance

GML Metadata Xpath	EOProduct ExtrinsicObject Attribute
"EODataLayer" (fixed value)	/ExtrinsicObject/@objectType
/atm:EarthObservationProduct/atm:dataLayers/atm:DataLayer/atm:specy	/ExtrinsicObject[@objectType="EODataLayer"]/Name/LocalizedString/Value
/atm:EarthObservationProduct/atm:dataLayers/atm:DataLayer/atm:unit	/ExtrinsicObject[@objectType="EODataLayer"]/Slot[@name="Unit"]/ValueList/Value[1]
/atm:EarthObservationProduct/atm:dataLayers/atm:DataLayer/atm:highestLocation	/ExtrinsicObject[@objectType="EODataLayer"]/Slot[@name="HighestLocation"]/ValueList/Value[1]
/atm:EarthObservationProduct/atm:dataLayers/atm:DataLayer/atm:lowestLocation	/ExtrinsicObject[@objectType="EODataLayer"]/Slot[@name="LowestLocation"]/ValueList/Value[1]
/atm:EarthObservationProduct/atm:dataLayers/atm:DataLayer/atm:algorithmName	/ExtrinsicObject[@objectType="EODataLayer"]/Slot[@name="AlgorithmName"]/ValueList/Value[1]
/atm:EarthObservationProduct/atm:dataLayers/atm:DataLayer/atm:algorithmVersion	/ExtrinsicObject[@objectType="EODataLayer"]/Slot[@name="AlgorithmVersion"]/ValueList/Value[1]

8 External Interfaces

This chapter describes the interactions with the EO ebRIM Catalogue, and the behavior of its CSW interface. It provides some examples of request and response message structures as part of the operation signatures, as specified in the [OGC 04-021r3] OGC™ Catalogue Service 2.0.0 (with Corrigendum) and [OGC 05-025r3] OGC™ ebRIM (ISO/TS 15000-3) Application Profile for CSW. It also documents supported query facilities and implementation guidances.

It should be noted that the EO Products Data Model defined in this specification is perfectly compliant with the [OGC 04-021r3] OGC™ Catalogue Service 2.0.0 specification (with Corrigendum) and with the [OGC 05-025r3] OGC™ ebRIM (ISO/TS 15000-3) Application Profile for CS-W. Chapter 7 has shown that no modifications or extensions are needed in the information model. This chapter will show that no modifications or extensions are needed in the requests and responses.

The following sections use significant parts of the [OGC 04-021r3] OGC™ Catalogue Service 2.0.0 specification (with Corrigendum) and the [OGC 05-025r3] OGC™ ebRIM (ISO/TS 15000-3) Application Profile for CS-W documents. To reduce the need to refer to those documents, this document copies some of their parts with small modifications.

8.1 Supported Protocol Binding and Available Operations

The used protocol for an ebRIM based Catalogue is HTTP (possibly with SOAP).

The following table summarizes required operations on ebRIM Catalogues:

Table 10 — Required Operations on ebRIM Catalogue Service

Operation	Description
GetCapabilities *(Description)*	Allows a client to retrieve service metadata that describe the computational and non-computational characteristics of the service.
GetRecords *(Discovery)*	The principal operation used to search Catalogue content and retrieve all or some members of the result set.
DescribeRecord *(Discovery)*	Allows a client to discover the information model(s) supported by the Catalogue and to retrieve type definitions.
GetRecordById *(Discovery)*	A simple means of retrieving one or more registry objects by their identifier.
GetRepositoryItem *(Discovery)*	Requests the repository item for some *ExtrinsicObject*.
Harvest *(Publication)*	Enables a ‘pull’ style of publication whereby a resource is retrieved from some remote location (URL) and inserted into the Catalogue

Following figure is an UML diagram summarizing the service interfaces. Optional operations are not shown on the schema.

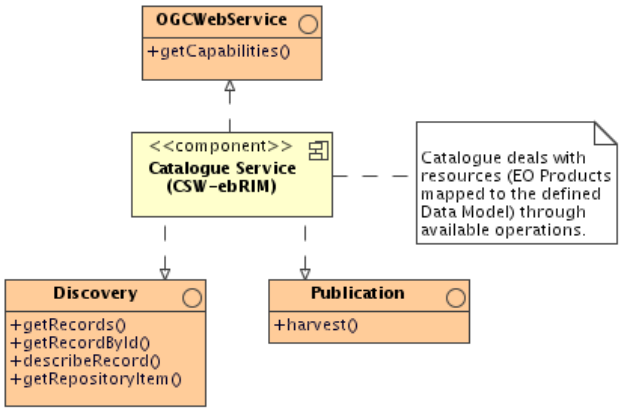

Figure 14: Service interface (CSW-ebRIM)

All operations must support the embedding of requests and responses in SOAP messages.

8.2 Interface Specifications

This chapter highlights syntax and semantic details of the interface operations specified in both [OGC 04-021r3] OGC™ Catalogue Service 2.0.0 (with Corrigendum) and [OGC 05-025r3] OGC™ ebRIM (ISO/TS 15000-3) Application Profile for CSW. It gives formal, language-independant interface (W3C WSDL) specifications that admit multiple programming language bindings and shows error conditions that can occur.

8.2.1 GetCapabilities Operation

The *GetCapabilities* operation allows clients to retrieve service metadata from a server. The response to a *GetCapabilities* request should be an XML document containing service metadata about the server (ISO 19119 document).

Here is a list of sections available in CSW ebRIM Capabilities Document:

Table 11 — Permissible Section Names

Section Names	Content
ServiceIdentification	General information about the service (type, version, etc.).
ServiceProvider	Information about the organization providing this service.
OperationsMetadata	Summarizes the operational characteristics of the service

Section Names	Content
Filter_Capabilities	Describes supported OGC filter operations
ServiceFeatures	Information about implemented features
ServiceProperties	Information about general service properties.

8.2.1.1 GetCapabilities Request

The value of the mandatory *service* parameter shall be the following service type identifier : '*urn:x-ogc:specification:csw-ebrim:Service:OGC-CSW:ebRIM*'. When included within a query component of the Request-URI, the '*:*' character (COLON) must be percent-encoded as '*%3A*', since that character is not a delimiter in this context.

The following XML-Schema fragment defines the XML encoding of the *GetCapabilities* operation request.

```
<!-- ========================================================== -->
<element name="GetCapabilities" type="ows:GetCapabilitiesType"/>
<!-- ========================================================== -->
<complexType name="GetCapabilitiesType">
  <annotation>
    <documentation>XML encoded GetCapabilities operation request. This operation allows clients to retrieve service metadata about a specific service instance. In this XML encoding, no "request" parameter is included, since the element name specifies the specific operation. This base type shall be extended by each specific OWS to include the additional required "service" attribute, with the correct value for that OWS. </documentation>
  </annotation>
  <sequence>
    <element name="AcceptVersions" type="ows:AcceptVersionsType" minOccurs="0">
      <annotation>
        <documentation>When omitted, server shall return latest supported version. </documentation>
      </annotation>
    </element>
    <element name="Sections" type="ows:SectionsType" minOccurs="0">
      <annotation>
        <documentation>When omitted or not supported by server, server shall return complete service metadata (Capabilities) document. </documentation>
      </annotation>
    </element>
    <element name="AcceptFormats" type="ows:AcceptFormatsType" minOccurs="0">
      <annotation>
        <documentation>When omitted or not supported by server, server shall return service metadata document using the MIME type "text/xml". </documentation>
      </annotation>
    </element>
  </sequence>
  <attribute name="updateSequence" type="ows:UpdateSequenceType" use="optional">
    <annotation>
      <documentation>When omitted or not supported by server, server shall return latest complete service metadata document. </documentation>
    </annotation>
  </attribute>
</complexType>
```

8.2.1.2 GetCapabilities Response

If the request is processed successfully, the body of the response message shall include an XML document where the document element has the following infoset properties:

- A [local name] of '*Capabilities*',
- A [namespace name] of '*http://www.opengis.net/cat/wrs*' (usually represented by the '*wrs*' prefix).

The document element MUST be valid against the following element declaration:

http://schemas.opengeospatial.net/csw-ebrim/1.0.0/wrs-capabilities.xsd#Capabilities

8.2.1.3 Web Service Description Language (WSDL)

The Web Services Description Language (WSDL) is an XML language to describe the computational characteristics of web services in terms of interfaces, protocol bindings, and service endpoints. WSDL 2.0 is currently a W3C Working Draft that defines a component model in terms of an abstract XML infoset.

A WSDL description may be used to complement the metadata provided in an OGC service capabilities document. The <wrs:WSDL-services> element is a simple link element that may be used to include a reference to a WSDL description containing service and binding elements. The value of the *xlink:href* attribute must be a resolvable URI that produces the WSDL document when it is the target of a GET request, the *xlink:role* attribute must indicate the relevant version of the WSDL specification (by namespace URI).

8.2.1.4 Exceptions

If an error condition arises while performing a *GetCapabilities* request, the service shall return an exception report as specified in [OGC 05-025r3] OGC™ ebRIM (ISO/TS 15000-3) Application Profile for CSW.

8.2.2 GetRecords Operation

The mandatory *GetRecords* operation is the principal operation used to search the catalogue content. Some or all the registry objects in the result set that satisfy the search criteria may be piggy-backed in the response message.

8.2.2.1 GetRecords Request

If the *Content-Type* of the request entity body is an XML content type (*application/xml*), the document element must be the '*csw:GetRecords*' element, as defined in the following schema:

http://schemas.opengeospatial.net/csw/2.0.1/CSW-discovery.xsd

Following table specifies attributes of the *GetRecords* operation message.

Table 12 — GetRecords Operation Parameters

Parameter	Data type and value	Optionality
Service	Character String. Fixed value of '*urn:x-ogc:specification:csw-ebrim:Service:OGC-CSW:ebRIM*'	Mandatory
Version	Character String. Fixed value of '*2.0.1*'	Mandatory

Parameter	Data type and value	Optionality
RequestId	CharacterString.	Not Supported[a]
ResultType	CodeList. One of '*hits*' (default value), '*results*' or '*validate*'.	Optional
OutputFormat	CharacterString. The only supported value is '*application/xml*' (default value)	Optional
OutputSchema	CodeList. One of '*urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0*' (default value) or '*http://www.opengis.net/cat/csw*'.	Optional
StartPosition	PositiveInteger. Default Value is *1*	Optional
MaxRecords	PositiveInteger. Default Value is *10*	Optional
TypeNames	List of Character String, comma separated. Unordered List of object types implicated in the query.	Mandatory
ElementSetName *or* ElementName	CodeList. One of '*brief*', '*summary*' (default value) or '*full*'.	Optional
ConstraintLanguage	CodeList. The only supported value is '*FILTER*' (default value)	Optional
Constraint	String. The predicate expression specified in the language indicated by the ConstraintLanguage parameter. Default action is to execute an unconstrained query.	Optional.
SortBy	List of Character String, comma separated Ordered list of names of metadata elements to use for sorting the response. Default action is to present the records in the order in which they are received.	Optional
DistributedSearch	Boolean.	Not Supported[a]
HopCount	Integer.	Not Supported[a]
ResponseHandler	URL.	Not Supported[a]
[a] Not supported yet in the ebRIM Application Profile		

The following XML-Schema fragments define the XML encoding of the *GetRecords* operation request:

```
<xsd:element name="GetRecords" type="csw:GetRecordsType" id="GetRecords"/>
<xsd:complexType name="GetRecordsType" id="GetRecordsType">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
    The principal means of searching the catalogue. The matching
    catalogue entries may be included with the response. The client
    may assign a requestId (absolute URI). A distributed search is
    performed if the DistributedSearch element is present and the
    catalogue is a member of a federation. Profiles may allow
    alternative query expressions.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="csw:RequestBaseType">
      <xsd:sequence>
        <xsd:element name="DistributedSearch" type="csw:DistributedSearchType" minOccurs="0"/>
        <xsd:element name="ResponseHandler" type="xsd:anyURI" minOccurs="0" maxOccurs="unbounded"/>
        <xsd:choice>
          <xsd:element ref="csw:AbstractQuery"/>
          <xsd:any namespace="##other" processContents="strict"/>
        </xsd:choice>
      </xsd:sequence>
      <xsd:attribute name="requestId" type="xsd:anyURI" use="optional"/>
      <xsd:attribute name="resultType" type="csw:ResultType" use="optional" default="hits"/>
      <xsd:attributeGroup ref="csw:BasicRetrievalOptions"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:complexType name="RequestBaseType" abstract="true" id="RequestBaseType">
  <xsd:annotation>
    <xsd:documentation>
    Base type for all request messages except GetCapabilities. The
    attributes identify the relevant service type and version.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="service" type="ows:ServiceType" use="optional" default="http://www.opengis.net/cat/csw"/>
  <xsd:attribute name="version" type="ows:VersionType" use="optional" default="2.0.1"/>
</xsd:complexType>
```

```
<xsd:element name="Query" type="csw:QueryType" substitutionGroup="csw:AbstractQuery" id="Query"/>
<xsd:complexType name="QueryType" id="QueryType">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
    Specifies a query to execute against instances of one or
    more object types. A set of ElementName elements may be included
    to specify an adhoc view of the csw:Record instances in the result
    set. Otherwise, use ElementSetName to specify a predefined view.
    The Constraint element contains a query filter expressed in a
    supported query language. A sorting criterion that specifies a
    property to sort by may be included.

    typeNames - a list of object types to query.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="csw:AbstractQueryType">
      <xsd:sequence>
        <xsd:choice>
          <xsd:element ref="csw:ElementSetName"/>
          <xsd:element name="ElementName" type="xsd:anyURI" maxOccurs="unbounded"/>
        </xsd:choice>
        <xsd:element ref="csw:Constraint" minOccurs="0"/>
        <xsd:element ref="ogc:SortBy" minOccurs="0"/>
      </xsd:sequence>
      <xsd:attribute name="typeNames" type="csw:TypeNameListType" use="required"/>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

```
<xsd:element name="ElementSetName" type="csw:ElementSetNameType" default="summary" id="ElementSetName"/>
<xsd:complexType name="ElementSetNameType" id="ElementSetNameType">
  <xsd:simpleContent>
    <xsd:extension base="csw:ElementSetType">
      <xsd:attribute name="typeNames" type="csw:TypeNameListType" use="optional"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

8.2.2.2 GetRecords Response

If the request is processed successfully, the body of the response message shall include an XML document where the document element has the following infoset properties:

- A [local name] of '*GetRecordsResponse*',
- A [namespace name] of '*http://www.opengis.net/cat/csw*' (usually represented by the '*csw*' prefix).

The search results may include a sequence of either *<csw:Record>* or *<rim:RegistryObject>* elements. In any case valid substitution elements may also be included, where these typically correspond to different views or instances of record subtypes.

The record representation must conform to the requested output schema. The value of the *outputSchema* attribute in the request restricts which elements may appear in the response. If not specified, ebRIM representations are returned.

Table 13 — Allowable catalogue record representation

OutputSchema	Record representations
http://www.opengis.net/cat/csw	csw:Record csw:SummaryRecord csw:BriefRecord
urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0	rim:RegistryObject *Any subtype of rim:RegistryObject*

The following XML-Schema fragments define the XML format response to a *GetRecords* operation:

```
<xsd:element name="GetRecordsResponse" type="csw:GetRecordsResponseType" id="GetRecordsResponse"/>
<xsd:complexType name="GetRecordsResponseType">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
    The response message for a GetRecords request. Some or all of the
    matching records may be included as children of the SearchResults
    element. The RequestId is only included if the client specified it.
    </xsd:documentation>
  </xsd:annotation>
  <xsd:sequence>
    <xsd:element name="RequestId" type="xsd:anyURI" minOccurs="0"/>
    <xsd:element name="SearchStatus" type="csw:RequestStatusType"/>
    <xsd:element name="SearchResults" type="csw:SearchResultsType"/>
  </xsd:sequence>
  <xsd:attribute name="version" type="xsd:string" use="optional"/>
</xsd:complexType>
```

```
<xsd:complexType name="RequestStatusType" id="RequestStatusType">
  <xsd:annotation>
    <xsd:documentation>
    This element provides information about the status of the
    search request.

    status    - status of the search
    timestamp - the date and time when the result set was modified
                (ISO 8601 format: YYYY-MM-DDThh:mm:ss[+|-]hh:mm).
    </xsd:documentation>
  </xsd:annotation>
  <xsd:attribute name="status" type="csw:StatusType" use="required"/>
  <xsd:attribute name="timestamp" type="xsd:dateTime" use="optional"/>
</xsd:complexType>
```

```
<xsd:simpleType name="StatusType" id="StatusType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="complete">
      <xsd:annotation>
        <xsd:documentation>
      The request was successfully completed and valid results
      are available.
      </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="subset">
      <xsd:annotation>
        <xsd:documentation>
      Partial, valid results are available.
      </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="interim">
      <xsd:annotation>
        <xsd:documentation>
      Partial results available, not necessarily valid.
      </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="none">
      <xsd:annotation>
        <xsd:documentation>No results are available</xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
    <xsd:enumeration value="processing">
      <xsd:annotation>
        <xsd:documentation>
      Request is still being processed.  Its status will be set
      to complete when request is done.
      </xsd:documentation>
      </xsd:annotation>
    </xsd:enumeration>
  </xsd:restriction>
</xsd:simpleType>
```

8.2.2.3 Query Samples

The <*csw:Query*> element is documented in Subclause 9.1.4 of [OGC 05-025r3]. It explains the use of binding variables – or aliases – to avoid ambiguity when specifying complex queries that navigate associations by traversing multiple links between related registry objects.

The value of the *Query/@typeNames* attribute is a whitespace-separated list of object types that constitute the scope of the query. Each value in the list MUST be a qualified type name. One or more variables may be bound to a type name.

For example, the following query allows to fetch the EOProducts types *ClassificationScheme* in order to retrieve all children or descendants *ClassificationNodes*. Classified *ExtrinsicObject* representing EOProducts can be discovered through *Classifications* linked to such *ClassificationNodes*.

Example 1 – GetRecords query, to fetch ClassificationScheme.

```
<?xml version="1.0" encoding="UTF-8"?>
<csw:GetRecords
  xmlns:csw="http://www.opengis.net/cat/csw"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
  outputSchema="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
  version="2.0.1"
  startPosition="1"
  maxRecords="20">
  <csw:Query typeNames="rim:ClassificationScheme">
    <csw:ElementSetName typeNames="rim:ClassificationScheme">full</csw:ElementSetName>
    <csw:Constraint version="1.1.0">
      <ogc:Filter>
        <!-- This clause means “Get the EO_Product_Types ClassificationScheme” -->
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>/rim:ClassificationScheme/@name</ogc:PropertyName>
          <ogc:Literal>EO_Product_Types</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Filter>
    </csw:Constraint>
  </csw:Query>
</csw:GetRecords>
```

The above query allows to retrieve the Earth Observation taxonomy. This enables queries per theme or mission type, as illustrated in the following sample query:

- *“Give me all EO Products instances of type PHR”*

More formally:

- *“Give me all objects linked to the ClassificationNode PHR through a Classification”*

Example 2 – GetRecords query based on acquisition type.

```
<?xml version="1.0" encoding="UTF-8"?>
<csw:GetRecords
  xmlns:csw="http://www.opengis.net/cat/csw"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
  outputSchema="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
  version="2.0.1"
  startPosition="1"
  maxRecords="20">
  <csw:Query typeNames="rim:ExtrinsicObject rim:Classification rim:ClassificationNode">
    <csw:ElementSetName typeNames="rim:ExtrinsicObject">full</csw:ElementSetName>
    <csw:Constraint version="1.1.0">
      <ogc:Filter>
        <ogc:And>
          <!-- This clause means “Select the PHR ClassificationNode” -->
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>/rim:ClassificationNode/@value</ogc:PropertyName>
            <ogc:Literal>PHR</ogc:Literal>
          </ogc:PropertyIsEqualTo>
          <!-- This clause means “Get Classification linked to the PHR ClassificationNode” -->
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>/rim:Classification/@classificationNode</ogc:PropertyName>
            <ogc:PropertyName>/rim:ClassificationNode/@id</ogc:PropertyName>
          </ogc:PropertyIsEqualTo>
          <!-- This clause means “Such Classifications must classify ExtrinsicObject” -->
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>/rim:Classification/@classifiedObject</ogc:PropertyName>
            <ogc:PropertyName>/rim:ExtrinsicObject/@id</ogc:PropertyName>
          </ogc:PropertyIsEqualTo>
          <!-- This clause means “Classified ExtrinsicObjects must be of type EOProduct” -->
          <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>/rim:ExtrinsicObject/@objectType</ogc:PropertyName>
            <ogc:Literal>EOProduct</ogc:Literal>
          </ogc:PropertyIsEqualTo>
        </ogc:And>
      </ogc:Filter>
    </csw:Constraint>
  </csw:Query>
</csw:GetRecords>
```

All representative acquisition parameters are available (as slot) for extended search like the following query:

- "*Give me all EO Product instances intersecting this footprint, measured after this date and shoot with this Along-Track Pointing Angle*".

More formally:

- "*Give me all ExtrinsicObjects with the objectType equals to 'EOProduct', the footprint slot value – a geometry – intersecting this box, the startDate slot value greater than this date and the alongTrackPointingAngle slot value equals to this*".

Example 3 – GetRecords query based on acquisition parameters.

```
<?xml version="1.0" encoding="UTF-8"?>
<csw:GetRecords
 xmlns:csw="http://www.opengis.net/cat/csw"
 xmlns:ogc="http://www.opengis.net/ogc"
 xmlns:gml="http://www.opengis.net/gml"
 xmlns:rim="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
 outputSchema="urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0"
 version="2.0.1"
 startPosition="1"
 maxRecords="20">
 <csw:Query typeNames="rim:ExtrinsicObject">
  <csw:ElementSetName typeNames="rim:ExtrinsicObject">full</csw:ElementSetName>
  <csw:Constraint version="1.1.0">
   <ogc:Filter>
    <ogc:And>
     <!-- This clause means "returned ExtrinsicObjects must represent EOProduct" -->
     <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>/rim:ExtrinsicObject/@objectType</ogc:PropertyName>
      <ogc:Literal>EOProduct</ogc:Literal>
     </ogc:PropertyIsEqualTo>
     <!-- This clause mean "ExtrinsicObject's FootPrint slot value must intersect with this box" -->
     <ogc:BBOX>
      <ogc:PropertyName>/rim:ExtrinsicObject/rim:Slot[@name='FootPrint']/rim:ValueList/rim:Value</ogc:PropertyName>
      <gml:Envelope>
       <gml:lowerCorner>48.86 –124.18</gml:lowerCorner>
       <gml:upperCorner>51.72 –111.64</gml:upperCorner>
      </gml:Envelope>
     </ogc:BBOX>
     <!-- This clause means "ExtrinsicObject's acquisitionStartDate slot value must be greater than that date" -->
     <ogc:PropertyIsGreaterThan>
      <ogc:PropertyName>/rim:ExtrinsicObject/rim:Slot[@name='acquisitionStartDate']/rim:ValueList/rim:Value</ogc:PropertyName>
      <ogc:Literal>2006-07-15T00:00:00Z</ogc:Literal>
     </ogc:PropertyIsGreaterThan>
     <!-- This clause means "ExtrinsicObject's acquisitionAlongTrackPointingAngle must be equals to that value" -->
     <ogc:PropertyIsEqualTo>
     <ogc:PropertyName>/rim:ExtrinsicObject/rim:Slot[@name='acquisitionAlongTrackPointingAngle']/rim:ValueList/rim:Value
     </ogc:PropertyName>
      <ogc:Literal>15.3</ogc:Literal>
     </ogc:PropertyIsEqualTo>
    </ogc:And>
   </ogc:Filter>
  </csw:Constraint>
 </csw:Query>
</csw:GetRecords>
```

OGC filter expression may contain spatial or temporal operators that specify a query against some characteristics of a registry object. Please refer to subclause 9.1 to get an overview of such mechanisms.

8.2.2.4 Exceptions

If the request is deemed invalid for any reason (e.g. missing a required element), the the service must return an *ows:ExceptionReport* containing a service exception with the code *wrs:InvalidRequest*.

8.2.3 DescribeRecord Operation

The *DescribeRecord* operation allows a client to discover the information model(s) supported by the catalogue and to retrieve record type definitions.

8.2.3.1 DescribeRecord Request

The *DescribeRecord* operation is described in Subclause 9.3 of [OGC 05.025r3]. The XML representation of the entity body, if present, must conform to the *csw:DescribeRecord* element declaration. The *TypeName* elements, if present, identify the model elements for which type definitions are requested.

The only schema language currently supported by the ebRIM Profile is W3C XML Schema. The corresponding value of the schemaLanguage attribute is given by the following URI: “http://www.w3c.org/2001/XMLSchema”

8.2.3.2 DescribeRecord Response

If the request is processed successfully, the body of the response message shall include an XML document where the document element has the following infoset properties:

- A [local name] of ‘*DescribeRecordResponse*’,
- A [namespace name] of ‘*http://www.opengis.net/cat/csw*’ (usually represented by the ‘*csw*’ prefix).

If no *TypeName* elements were provided in the request, whole schemas defining the information model must be included within *csw:SchemaComponent* elements. If there are no matching schema components, the document element must be empty.

The content of a *csw:SchemaComponent* element may be a complete schema or a fragment of one. If it is a fragment, the *parentSchema* attribute must reference the source schema (by identifier).

8.2.3.3 Exceptions

If the request is deemed invalid for any reason (e.g. missing a required element), the the service must return an *ows:ExceptionReport* containing a service exception with the code *wrs:InvalidRequest*.

8.2.4 GetRecordById Operation

The *GetRecordById* operation provides a simple mean of retrieving one or more records by identifier; the identifier may be that of some registry object (*rim:RegistryObject/@id*) or an external identifier (*rim:ExternalIdentifier/@value*) assigned to a registry object.

8.2.4.1 GetRecordById Request

The *GetRecordById* operation is described in Subclause 9.2 of [OGC 05-025r3]. The XML representation of the entity body, if present, must conform to the *csw:GetRecordById* element declaration. All reserved characters (e.g., general delimeters) appearing in identifier values must be suitably percent-encoded in the KVP representation when using the GET method.

The value of an *Id* (message parameter) item identifies a registry objet either directly or by an external identifier that corresponds to a child *rim:ExternalIdentifier* element.

```
<xsd:element name="GetRecordById" type="csw:GetRecordByIdType" id="GetRecordById"/>
<xsd:complexType name="GetRecordByIdType" id="GetRecordByIdType">
  <xsd:annotation>
    <xsd:documentation xml:lang="en">
    Convenience operation to retrieve default record representations
    by identifier.
    Id - object identifier (a URI) that provides a reference to a
        catalogue item (or a result set if the catalogue supports
        persistent result sets).
    ElementSetName - one of "summary", or "full"
    </xsd:documentation>
  </xsd:annotation>
  <xsd:complexContent>
    <xsd:extension base="csw:RequestBaseType">
      <xsd:sequence>
        <xsd:element name="Id" type="xsd:anyURI" maxOccurs="unbounded"/>
        <xsd:element ref="csw:ElementSetName" minOccurs="0"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

8.2.4.2 GetRecordById Response

If the request is processed successfully, the body of the response message shall include an XML document where the document element has the following infoset properties:

- A [local name] of '*GetRecordByIdResponse*',
- A [namespace name] of '*http://www.opengis.net/cat/csw*' (usually represented by the '*csw*' prefix).

The child elements must be registry object representations (i.e. *rim:RegistryObject* or some valid substitution element) corresponding to the requested property set. If a match for an external identifier is found, the parent registry object is included. If there are no matching records, an empty response is returned.

8.2.4.3 Exceptions

If the request is deemed invalid for any reason (e.g. missing a required element), the the service must return an *ows:ExceptionReport* containing a service exception with the code *wrs:InvalidRequest*.

8.2.5 GetRepositoryItem Operation

The *GetRepositoryItem* operation is used to retrieve the repository item corresponding to some extrinsic object. If available, the item is included in the body of the response; it must be an instance of a MIME media type, as indicated by the value of the *Content-Type* header field.

An extrinsic object may also be used to catalogue an external repository item that is managed by another party. In this case, the *ExtrinsicObject* must be associated (using the '*RepositoryItemFor*' association) with an *ExternalLink* that specifies an absolute URL for retrieving the item.

8.2.5.1 GetRepositoryItem Request

The request is bound only to the GET method. All reserved characters appearing in parameter values must be suitably percent-encoded. The request parameters are listed in the following table:

Table 14 — GetRepositoryItem Operation Parameters

Parameter	Data type and value	Optionality
service	Character String. Fixed value of '*urn:x-ogc:specification:csw-ebrim:Service:OGC-CSW:ebRIM*'	Mandatory
request	Character String. Fixed value of '*GetRepositoryItem*'	Mandatory
id	CharacterString. Absolute URI that refers to some extrinsic object	Mandatory

8.2.5.2 GetRepositoryItem Response

If the request is processed successfully and a repository item is accessible, the body of the response message shall include the repository item as a MIME entity. If any additional encodings have been applied to the resource (e.g., compression using gzip), these must be specified by the *Content-Encoding* header field.

In some cases the resource may reside in an external repository maintained by another party. In this case, the catalogue shall redirect the client using the standard HTTP redirection mechanism (i.e., status code 303, "See Other") and set the value of location header field according to the value of *ExternalLink/@externalURI* attribute.

8.2.5.3 Exceptions

If the request is deemed invalid for any reason (e.g., missing identifier), then the service must return an *ows:ExceptionReport* containing a service exception with the code *wrs:InvalidRequest*. If the supplied identifier does not match any registry object or if a repository item cannot be located, the response must include an exception with the code *wrs:NotFound*.

8.2.6 Harvest Operation

The *Harvest* operation is described in Subclause 10.1 of [OGC 05-025r3]. It allows a user to request the catalogue to harvest a repository item from a specified network location, thereby realizing a 'pull' model for publishing registry content. If the catalogue successfully retrieves the resource and successfully processes it, then one or more corresponding registry objects are created or updated. Brief representations of all modified records are returned to the client when processing is complete.

8.2.6.1 Harvest Request

This request is only bound to the HTTP POST method. The XML representation of the entity body must conform to the *csw:Harvest* element declaration.

The *csw:Source* element specifies a URL from which the resource may be retrieved. The scheme component should correspond to a protocol supported by the catalogue; support for the '*http*' scheme is required by all conforming implementations, and '*HTTP/1.1*' must be listed in the capabilities document as a value for the '*harvest-protocols*' system property.

If specified, the *csw:ResourceType* element must indicate the object type of the corresponding extrinsic object. It may be possible for the catalogue to deduce this from the content of the resource (for example, a data set description that conforms to the ISO 19139 schemas). The value should correspond to a type supported by the catalogue, as identified in the *objectType* classification scheme.

The *harvest* operation definition shall advertise the support for EO Metadata *resourceType* using the following values:

- hma:EarthObservationProduct,
- sar:EarthObservationProduct
- ohr:EarthObservationProduct
- atm:EarthObservationProduct

Note 1 – the 3 letters acronyms (hma, sar, ohr and atm) are not XML prefixes. ResourceType is not a QName. To avoid such a confusing behavior, we suggest that these should become URNs register by the OGC as in [05-025r3].

Example 4 – Harvest Operation Definition in the Capabilies

```
<Operation name="Harvest">
    <DCP>
        <HTTP>
            <Get xlink:href="http://...." xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink"/>
            <Post xlink:href="http://..." xlink:type="simple" xmlns:xlink="http://www.w3.org/1999/xlink"/>
        </HTTP>
    </DCP>
    <Parameter name="resourceType">
        <Value>hma:EarthObservationProduct</Value>
        <Value>sar:EarthObservationProduct</Value>
        <Value>ohr:EarthObservationProduct</Value>
        <Value>atm:EarthObservationProduct</Value>
        <Value>ISO19139</Value>
        <Value>Context</Value>
        <Value>SOS</Value>
        <Value>WMS</Value>
        <Value>WCS</Value>
        <Value>WFS</Value>
```

```
    </Parameter>
    <Parameter name="source">
    </Parameter>
</Operation>
```

8.2.6.2 Harvest Response

If the request is processed successfully, the body of the response message shall include an XML document where the document element has the following infoset properties:

- A [local name] of '*HarvestResponse*',
- A [namespace name] of '*http://www.opengis.net/cat/csw*' (usually represented by the '*csw*' prefix).

The document element must include a *csw:TransactionResponse* element that contains the *csw:InsertResults* child element; this element must list all registry objects that were created as a result of the harvesting operation.

8.2.6.3 Exceptions

If the resource cannot be retrieved from the source URL, an exception with the code *wrs:NotFound* must be included in an *ogc:ExceptionReport*. If the resource format is not supported by the catalogue or the object type is not recognized, an exception with code *wrs:NotSupported* must be returned. In the event that the transaction cannot be completed for any reason, an exception with the *wrs:TransactionFailed* code must be returned.

Annex A
(normative)

Abstract test suite

In each Implementation Specification document, Annex A shall specify the Abstract Test Suite, as specified in Clause 9 and Annex A of ISO 19105. That Clause and Annex specify the ISO/TC 211 requirements for Abstract Test Suites. Examples of Abstract Test Suites are available in an annex of most ISO 191XX documents, one of the more useful is in ISO 191TBD. Note that this guidance may be more abstract than needed in an OGC™ Implementation Specification.

Inclusion of the Abstract Test Suite is expected in version 1.0.0 of each OGC Implementation Specification. In earlier versions, the following paragraph can be used:

An abstract test suite is not provided in this version of this Implementation Specification, but will be provided in version 1.0.0.

A.1 General

A paragraph.

Annex B
(normative)

XML Schema Documents

This template recommends referencing the XML Schema Documents here, and not including the schema document contents in an OWS specification. However, some readers prefer including the schema document contents in a specification. The personal preferences of the specification editor(s) should not be a significant consideration. This template leaves the choice up to the editor(s).

The term “XML schema“ means all the XML schema parts having the same XML namespace, usually separated into multiple XML Schema Document files (with the file type “.xsd“. The XML schema parts in one XML namespace are usually separated into multiple XML Schema Documents to ease human understanding.

Annex C
(informative)

Implementation Architectures

This specification describes interface and behavior of ebRIM Catalogues able to deal with Earth Observation Product Metadata. Such Catalogues can be deployed to manage EO Resources, in a ‘stand-alone’ mode (the catalogue acts as a registry and a repository where metadata are stored and indexed), or to provide an OGC compliant layer upon legacy catalogues, already deployed and serving existing data.

This annex focuses on these two possibilities, and provides some guidance on the way to implement both architectures.

C.1 Stand-Alone Architecture

This is the simplest architecture. Catalogue is used as a repository (for storing data) and a registry (for indexing data). It is OGC compliant and provides all capabilities and operations defined in OGC ebRIM Application Profile for CSW document. It is able to map incoming data (in this case EO Products) to objects defined by the Data Model described in this sepcification.

Every resource (EarthObservation Product) is stored within the Catalogue and indexed to allow complex queries and fast retrieving. If needed, additional ebRIM associations and classifications (e.g., to additional business-related taxonomies) can extend the EO data model and provide additionnal metadata discovery facilities to implement new use-cases. EO Products might, for example, be linked to an object or structure describing their pricing model, in an eBusiness use-case.

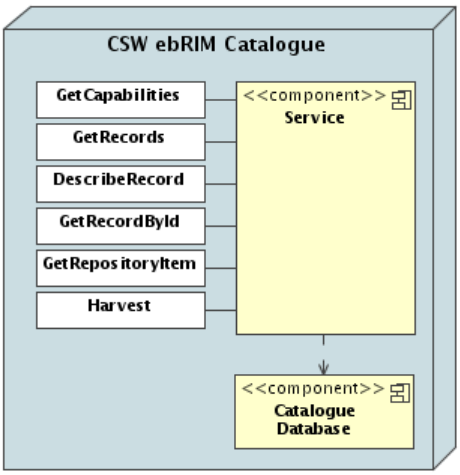

Figure 15: Stand-Alone Architecture

C.2 Proxy Architecture

If Earth Observation Products are already stored in legacy (non OGC-compliant) catalogues, the goal of this architecture is to provide an OGC compliant interface onto existing data.

From a very high-level point of view, two modes can be considered:

- The OGC layer is viewed as a **front-end** layer on the legacy catalogue: incoming requests are linked on the fly to the legacy catalogue, result sets are converted from the legacy format to the OGC CSW ebRIM format and responses are sent back to the user.

- The OGC layer is viewed as a **replication** of the legacy catalogue: metadata are harvested from the legacy catalogue, resulting in indexes (and eventually metadata themselves) available for discovery directly in the OGC layer. Queries are processed without querying on the legacy catalogue. Synchronization mechanisms exist between the two catalogues.

C.2.1 Front-End Architecture

In this architecture, OGC queries must be translated to their native version (understandable by legacy catalogues) and executed on proxied catalogue in real time. Native responses must then be translated to be sent to the final user in an OGC compliant way.

OGC CSW ebRIM Interface operations must either be directly mapped to operations available on the legacy catalogue, or processed (or emulated) by the CSW front-end. It can result in some limitations; the legacy catalogue might not provide all the functionnalities needed to implement a compliant OGC CSW ebRIM interface.

Such architecture requires a permanent connection between the legacy catalogue and the front-end, and can considerably increases network traffic. Performance is strongly dependant of the legacy infrastructure. Each required CSW operation should be emulated using a combination of one or more legacy operations. The results will then be processed and formatted to be returned in an OGC compliant way.

A source of limitation and complexity appears if the legacy catalogue is not able to serve the metadata in the EO GML format. Indeed, if requested, the EO GML should be generated on-the-fly, and the information available in the legacy catalogue might not be sufficient to fill in the EO GML structure. Moreover, the legacy metadata might be organized in very different way, needing multiple requests to gather needed information.

If metadata are rather static in the legacy database, a minimal caching mechanism can be used in the front-end layer to speed up queries, and avoid roundtrips to the legacy catalogue.

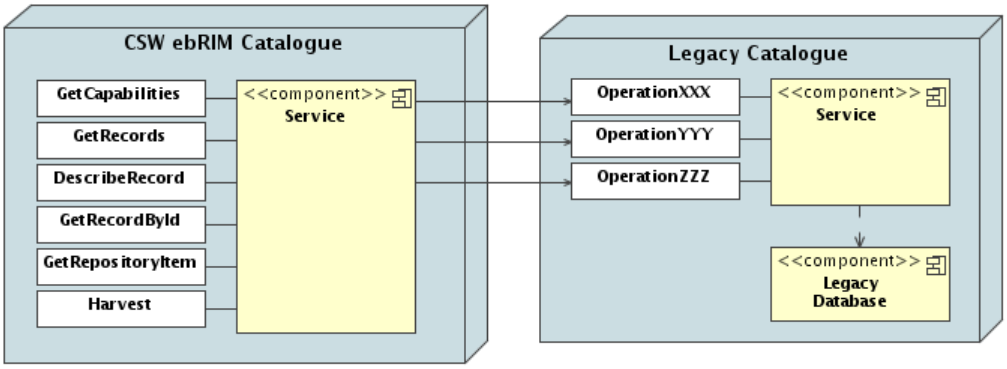

Figure 16: Front-End Architecture

This architecture avoids replication of data and synchronization mechanism to keep published data updated, indeed nothing is stored in the front-end layer.

C.2.2 Replication Architecture

The OGC layer is viewed as a **replication** of the legacy catalogue: indexes on metadata - and eventually metadata themselves - are available for discovery directly in the OGC layer. Queries are processed without querying in real-time the legacy catalogue. Synchronization (periodical or permanent) is needed between the two catalogues. This synchronization may be bidirectional if the replicated OGC Catalogue allows harvesting and/or transactional operations.

Harvest operation provides a publication ability to replicate every EarthObservation Products coming from the legacy catalogue to the OGC compliant one. Every resource is periodically submitted to the ebRIM Catalogue for storing and indexing, allowing complex queries and fast retrieving.

In this case, the only source of limitation and complexity comes from the quality and organization of the metadata coming from the legacy Catalogue. The legacy catalogue might not able to serve the metadata in the EO GML format. The EO GML should be generated on-the-fly when requested, or generated during the Harvesting operation and stored in the replicated OGC Catalogue. Once again, the information available in the legacy catalogue might not be sufficient to fill in the EO GML structure.

Note that, if needed, like for the stand-alone architecture, additional ebRIM associations and classifications (e.g., to additional business-related taxonomies) can extend the EO data model and provide additional metadata discovery facilities to implement new use-cases. EO Products might, for example, be linked to an object or structure describing their pricing model, in an eBusiness use-case.

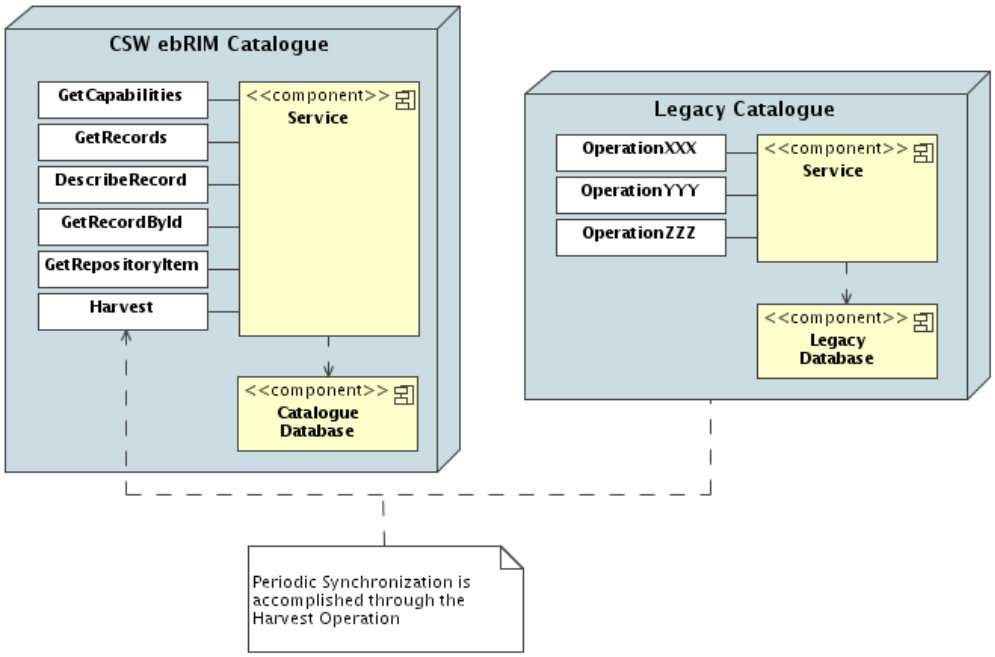

Figure 17: Replication Architecture

C.2.3 'Front-End' vs 'Replication' Comparison

Following table highlights benefits and disadvantages in both proxied architectures:

Table 15 — 'Front-End' vs 'Replication' Comparison

Front-End Architecture	Replication Architecture
(+) No database duplication, **(+)** Always 'synchronized' with the legacy catalogue,	**(+)** Full compliance on query requirements can be fulfilled, **(+)** Use of additional classifications or associations if needed, **(+)** No direct access to the legacy database, metadata are available in CSW database,

Front-End Architecture	Replication Architecture
(-) Inherits limitations from the legacy database and data access, **(-)** Permanent connection required to access legacy database, **(-)** Time and network-traffic consuming, **(-)** Full power of ebRIM cannot be exploited (additional classifications or associations),	**(-)** Database duplication, **(-)** Synchronization process needed,

Annex D
(informative)

UML model

A UML model must be included in an OGC™ Implementation Specification before it can be submitted to ISO/TC 211. This requirement is stated in ISO 19119, copied as OGC™ Abstract Specification Topic 12. This template thus includes this annex as the place for recording this UML model. Early drafts and versions of OGC™ Implementation Specifications often omit this UML model. We recommend that this annex be retained as a place holder for where the UML model will eventually be included.

Annex E
(informative)

Example XML documents

This annex can be included if useful to provide more XML document examples.

D.1 Introduction

This annex provides more example XML documents than given in the body of this document. TBD

D.2 TBD

Bibliography

[1] Guidelines for Successful OGC Interface Specifications, OGC document 00-014r1

A **Bibliography**, if useful, shall appear after the last annex. The bibliography may include

a) documents that are not publicly available,

b) documents to which only informative reference is made, and

c) documents which have merely served as references in the preparation of the document.