D3.3 Big Data Connectors and Catalogue Service v1

WP3 – Advanced Analytics and Knowledge Base

Deliverable Lead: CLS

Dissemination Level: Public

Deliverable due date: 30/11/2016

Actual submission date: 29/11/2016

Version 1.1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 687275

Document Control Page	
Title	Big Data Connectors and Catalogue Service v1
Creator	CLS
Description	Description of the data organization to standardize the service according to INSPIRE, OGC and ISO standards and conventions.
Publisher	EO4wildlife Consortium
Contributors	JM Zigna (CLS)
Creation date	20/09/2016
Type	Text
Language	en-GB
Rights	copyright “EO4wildlife Consortium”
Audience	☒ Public ☐ Confidential ☐ Classified
Status	☐ In Progress ☐ For Review ☐ For Approval ☒ Approved

Disclaimer

This deliverable is subject to final acceptance by the European Commission.

The results of this deliverable reflect only the author's view and the Commission is not responsible for any use that may be made of the information it contains.

Statement for open documents

Table of Contents

List of Figures

List of Tables

EO4wildlife Project Overview

EO4wildlife main objective is to bring large number of multidisciplinary scientists such as biologists, ecologists and ornithologists around the world to collaborate closely together while using European Sentinel Copernicus Earth Observation more heavily and efficiently.

In order to reach such important objective, an open service platform and interoperable toolbox will be designed and developed. It will offer high level services that can be accessed by scientists to perform their respective research. The platform front end will be easy-to-use, access and offer dedicated services that will enable them process their geospatial environmental stimulations using Sentinel Earth Observation data that are intelligently combined with other observation sources.

Specifically, the EO4wildlife platform will enable the integration of Sentinel data, ARGOS archive databases and real time thematic databank portals, including Wildlifetracking.org, Seabirdtracking.org, and other Earth Observation and MetOcean databases; locally or remotely, and simultaneously.

EO4wildlife research specialises in the intelligent management big data, processing, advanced analytics and a Knowledge Base for wildlife migratory behaviour and trends forecast. The research will lead to the development of web-enabled open services using OGC standards for sensor observation and measurements and data processing of heterogeneous geospatial observation data and uncertainties.

EO4wildlife will design, implement and validate various scenarios based on real operational use case requirements in the field of wildlife migrations, habitats and behaviour. These include:

- Management tools for regulatory authorities to achieve real-time advanced decision-making on the protection of protect seabird species;
- Enhancing scientific knowledge of pelagic fish migrations routes, reproduction and feeding behaviours for better species management;
- Enable researchers better understand the movement behaviour of sea turtle populations; and
- Setting up tools to assist marine protected areas and management.

Abbreviations and Glossary

A common glossary of terms for all EO4wildlife deliverables, as well as a list of abbreviations, can be found in the public document “EO4wildlife Glossary” available at EO4wildlife.eu.

Executive Summary

This document provides the description of data federation facilities for organising and accessing large distributed datasets (e.g. ARGOS datasets and Sentinel datasets) in order to serve high-level exploratory and predictive analytics. Given the "Big" and heterogeneous nature of remote, in situ and model data sources, this task will employ the state-of-the-art big data technologies to support *loading, extraction, aggregation, and transformation of large variety of data formats* (including structured, semi-structured, unstructured data) with very high throughput. Specifically, this document provides:

a) An abstract data access layer which allows access to heterogeneous data sources using common interfaces, while developing drivers (i.e. data connectors) to integrate remote data sources and others;

b) Data pre-processing engine, a robust backbone for processing raw data and aggregate them while overcoming for example their multi-modality, spatial resolution heterogeneity, temporal a-synchronicity, conflicts and gaps

c) Catalogue service to maintain the collection of descriptive metadata that can be searched by users.

Thus this document presents the organization of data and services to access it in EO4wildlife.

It covers not only the way data are collected from external providers, but also how is managed the data in the big data infrastructure so as to optimize the data mining and analysis.

It contains the following chapters:

- **Chapter 1:** General overview on existing standards, formats and connectors to acess to data.
- **Chapter 2:** 'EO4wildlife data access services' presents the services available to ingest or display external data in the EO4wildlife infrastructure. It includes external products access and Animal Tracks access.
- **Chapter 3:** The 'management of big data' provides a description of the ingested data management in the platform to optimize the access during the processing. Key performance issues are described here.
- **Chapter 4:** The 'catalogue' is a key component to discover the products and the services involved in the E04wildlife process.
- **Chapter 5:** Deliverables describes the way to deploy and configure the components used to offer the standardized connectors and the catalogue.

1 General Overview

The wide variety of data to deal with can be a blocking factor in the development of EO4wildlife services if no standardization is set in place to discover, to extract, to process and to display them.

On the one hand, we have to face with the diversity of animal positions databases. A first step consists in uploading the data in CSV to facilitate the development and integration of the platform. This possibility is based on existing database export mechanisms already in place in SeaTurtle or Birdlife platforms. A second step consists in refining the information linked to positions to extend the potential of processes, in setting in place standards to ensure the consistency of the information and use the potential of the interoperability. This standardization is realized through a predefined set of adapters so as to open the platform to a larger community of scientists in a secure way.

On the other hand, we have to deal with a large amount of environmental data sources in different standards. According to users' needs, a selection of environmental data is listed in to offer high quality and homogeneous datasets in time to scientists. As far as possible, we standardize the access to such external data through two main connectors: one for CMEMS and AVISO datasets in NetCDF format, the other one for Sentinel data.

1.1 State of the Art for data connectors and Catalogue service

This paragraph describes the different existing technologies to extract and display data. These different tools are based on ISO/OGC standards and compliant with the INSPIRE directive. They are widely used by the scientificc community and in EU projects such as CMEMS. The objective is to reuse them in the EO4wildlife platform for data connection.

1.1.1 THREDDS

The goal of Unidata's Thematic Real-time Environmental Distributed Data Services (THREDDS) is to provide students, educators and researchers with coherent access to a large collection of real-time and archived datasets from a variety of environmental data sources at a number of distributed server sites. The THREDDS Data Server (TDS) is a web server that provides metadata and data access for scientific datasets, using a variety of remote data access protocols.

Some of the technologies in the TDS:

1. THREDDS Dataset Inventory Catalogs are used to provide virtual directories of available data and their associated metadata. These catalogs can be generated dynamically or statically.
2. The Netcdf-Java/CDM library reads NetCDF, OpenDAP, and HDF5 datasets, as well as other binary formats such as GRIB and NEXRAD into a Common Data Model (CDM), essentially an (extended) netCDF view of the data. Datasets that can be read through the Netcdf-Java library are called *CDM datasets*.
3. TDS can use the NetCDF Markup Language (NcML) to modify and create virtual aggregations of CDM datasets.
4. An integrated server provides OPeNDAP access to any CDM dataset. OPeNDAP is a widely used, subsetting data access method extending the HTTP protocol.
5. An integrated server provides bulk file access through the HTTP protocol.
6. An integrated server provides data access through the OpenGIS Consortium (OGC) Web Coverage Service (WCS) protocol, for any "gridded" dataset whose coordinate system information is complete.

7. An integrated server provides data access through the OpenGIS Consortium (OGC) Web Map Service (WMS) protocol, for any "gridded" dataset whose coordinate system information is complete. This software was developed by Jon Blower (University of Reading (UK) E-Science Center) as part of the ESSC Web Map Service for environmental data (aka Godiva2).
8. The integrated ncISO server provides automated metadata analysis and ISO metadata generation.
9. The integrated NetCDF Subset Service allows subsetting certain CDM datasets in coordinate space, using a REST API. Gridded data subsets can be returned in CF-compliant netCDF-3 or netCDF-4. Point data subsets can be returned in CSV, XML, or CF-DSG netCDF files.

The TDS is used to make homogenous the access to gridded data.

1.1.2 MOTU

Motu is a high efficient and robust Web Server which fills the gap between heterogeneous Data Providers to End Users. Motu handles, extracts and transforms oceanographic huge volumes of data without performance collapse.

This CLS open source component is compliant with central authorization servers (https://wiki.jasig.org/display/casum/home) and is used as a front end for THREDDS servers subsettting requests. Requests to extract data can be made via HHTP, in synchronous mode, or asynchronous mode for big data extractions.

It is deployed in CMEMS, AVISO and CLS dissemination units.

1.1.3 OCEANOTRON

Oceanotron (https://forge.ifremer.fr/plugins/mediawiki/wiki/oceanotron/index.php/Accueil) is an IFREMER open-source data server dedicated to marine in-situ observation dissemination. For its extensibility it relies of an ocean business data model.

IFREMER participates to numerous ocean data management projects. IFREMER wished to capitalize its thematic data management expertise in a dedicated data dissemination server called Oceanotron. The development of the server coordinated by IFREMER has started in 2010. Knowing the diversity of data repository formats (RDBMS, netCDF, ODV, MEDATLAS, ...) and the temperamental nature of the standard interoperability interface profiles (OGC/WMS, OGC/WFS, OGC/SOS, OpenDAP, …), the architecture of the software relies on an ocean business data model dedicated to marine in-situ observation features.

The ocean business data model relies on the CSML conceptual modelling (http://csml.badc.rl.ac.uk/) and UNIDATA Common Data Model (http://www.unidata.ucar.edu/software/netcdf-java/CDM/) works and focuses on the most common marine observation features which are: vertical profiles, point series, trajectories and points.

The oceanotron server orchestrates different types of modules handling the ocean business data model objects:

- StorageUnits: which read specific data repository formats (netCDF/OceanSites, netCDF/ARGO, ...).
- TransformationUnits: which apply useful ocean business related transformation to the features (for example conversion of vertical coordinates from pressure in dB to meters under sea surface).
- FrontDesks: which get external requests and send results for interoperable protocols (OpenDAP, WMS, ...).

These modules are chained back and forth to process user requests. They exchange requests and observation features formatted in the ocean business data model. This inner-interoperability level enables to capitalize ocean business expertise in software development without being indentured to specific data

formats or protocols. In addition, different development teams can work together on different modules, depending on their expertise. IFREMER/ALTRAN/EFFITIC has developed the core system, netCDF storageUnits and OpeNDAP frontDesk. University of Reading has developed the OGC/WMS frontdesk.

Oceanotron is the data dissemination server for CMEMS in-situ data providers such as Coriolis data. This software is also be used as an interoperability server for data providers in SeaDataNet2 project.

1.1.4 WMS

A Web Map Service (WMS) is a standard protocol for serving (over the Internet) georeferenced map images generated by a map server using data from a GIS database

WMS specifies a number of different request types, two of which are required by any WMS server.

- GetCapabilities - returns parameters about the WMS (such as map image format and WMS version compatibility) and the available layers (map bounding box, coordinate reference systems, URI of the data and whether the layer is mostly opaque or not)
- GetMap - returns a map image. Parameters include: width and height of the map, coordinate reference system, rendering style, image format

Request types that WMS providers may optionally support include:

- GetFeatureInfo - if a layer is marked as 'queryable' then you can request data about a coordinate of the map image.
- DescribeLayer - returns the feature types of the specified layer or layers, which can be further described using WFS or WCS requests. This request is dependent on the SLD Profile of WMS. GetLegendGraphic - return an image of the map's legend image, giving a visual guide to map elements.

1.1.5 WFS

In computing, the Open Geospatial Consortium **Web Feature Service** Interface Standard (**WFS**) provides an interface allowing requests for geographical features across the web using platform-independent calls. One can think of geographical features as the "source code" behind a map, whereas the WMS interface or online tiled mapping portals like Google Maps return only an image, which end-users cannot edit or spatially analyze. The XML-based GML furnishes the default payload-encoding for transporting geographic features, but other formats like shapefiles can also serve for transport. In early 2006 the OGC members approved the OpenGIS GML Simple Features Profile. This profile is designed both to increase interoperability between WFS servers and to improve the ease of implementation of the WFS standard.

The OGC membership defined and maintains the WFS specification. Numerous commercial and open-source implementations of the WFS interface standard exist, including the open-source reference implementations GeoServer and deegree.

The basic Web Feature Service allows querying and retrieval of features. A transactional Web Feature Service (WFS-T) allows creation, deletion, and updating of features.

1.1.6 WCS

An OGC Web Coverage Service provides access to coverage data in forms that are useful for client-side rendering, as input into scientific models, and for other clients. The WCS may be compared to the OGC Web Feature Service (WFS) and the Web Map Service (WMS). As with WMS and WFS service instances, a WCS

allows clients to choose portions of a server's information holdings based on spatial constraints and other query criteria.

Unlike OGC Web Map Service (WMS), which portrays spatial data to return static maps (rendered as pictures by the server), the Web Coverage Service provides available data together with their detailed descriptions; defines a rich syntax for requests against these data; and returns data with its original semantics (instead of pictures) which may be interpreted, extrapolated, etc., and not just portrayed.

Unlike OGC Web Feature Service (WFS), which returns discrete geospatial features, the Web Coverage Service returns coverages representing space/time-varying phenomena that relate a spatio-temporal domain to a (possibly multidimensional) range of properties.

As such, WCS focuses on coverages as a specialized class of features and, correspondingly, defines streamlined functionality.

WCS uses the coverage model of the OGC GML Application Schema for Coverages. Thus, WCS supports all coverage types supported by said Application Schema; it is not constrained to quadrilateral grid coverages like previous WCS versions.

1.1.7 CSW

See Section 4 “Catalogue”.

1.1.8 Geoserver

In computing, GeoServer is an open-source server written in Java that allows users to share, process and edit geospatial data. Designed for interoperability, it publishes data from any major spatial data source using open standards. GeoServer has evolved to become an easy method of connecting existing information to virtual globes such as Google Earth and NASA World Wind as well as to web-based maps such as OpenLayers, Google Maps and Bing Maps. GeoServer functions as the reference implementation of the Open Geospatial Consortium Web Feature Service standard, and also implements the Web Map Service, Web Coverage Service and Web Processing Service specifications.

1.2 Data organization

1.2.1 Netcdf

NetCDF (network Common Data Form) is a set of interfaces for array-oriented data access and a freely distributed collection of data access libraries for C, Fortran, C++, Java, and other languages. The netCDF libraries support a machine-independent format for representing scientific data. Together, the interfaces, libraries, and format support the creation, access, and sharing of scientific data. NetCDF was developed and maintain by UNIDATA, http://www.unidata.ucar.edu/.

NetCDF data is:

- *Self-Describing*. A netCDF file includes information about the data it contains.
- *Portable*. A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- *Scalable*. A small subset of a large dataset may be accessed efficiently.
- *Appendable*. Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
- *Sharable*. One writer and multiple readers may simultaneously access the same netCDF file.

- *Archivable*. Access to all earlier forms of netCDF data will be supported by current and future versions of the software.

The several scientific groups are processing and disseminating their data using Netcdf. The Copernicus Marine Environment Monitoring Service disseminates Oceanographic and weather products using the NetCDF format. On the top of it, conventions such as COARDS-CF are in place to homogenize the naming of variables.

In D3.1, Knowledge Base Service architecture Specification v1 document [1], the EO4wildlife 'Knowledge base design' item gives additional details and samples of typical variables used in EO4wildlife.

1.2.2 Weather and oceanographic data

Gridded data are extracted from external providers using their MOTU interface. Only the relevant data to process is extracted according to the geographical coverage, the time coverage and the variables required for the EO4wildlife process.

When the MOTU interface is not provided natively by a provider, such as for winds data, the EO4wildlife project has imported the required data so as to homogenize the access by the platform.

The extracted data is then ingested in the EO4wildlife platform, assigned to a unique resource identifier in the internal EO4wildlife catalogue and managed by a geoserver to be accessed via WMS or WFS or WCS protocols.

Static data such as bathymetry is ingested once in the EO4wildlife platform and can be requested via WMS and WCS protocols.

End users will have also the possibility to download the data in NetCDF.

1.2.3 Animal tracking data

The main issue is that ARGOS dissemination services are not providing a standardized format to animal tracking end-users. The ARGOS/PRV or the ARGOS/DS formats are familiar to the ARGOS community but not compliant with interoperability. Even the ARGOS/XML dissemination format is organized according to the ARGOS System requirements with satellite paths, and is not a good starting point to process animal tracking data that have nothing to do with the satellite reception and only worried about animal locations and sensors.

Taking into account the Bird Life, Sea Turtle, CLS and AAMP feedbacks, an XML content including all the relevant information to process in the EO4wildlife platform has been defined. To be able to open the EO4wildlife platform to an extended community of scientists, the formats and ontologies defined for Acoustic Animal Tracking by IOOS (Integrated Ocean Observing System) and by the OTN (Ocean Tracking Network) were analysed. The result is a EO4wildlife XML content as close as possible to such communities terminology, with EO4wildlife details. This induces that an adapter should be developed by the external platforms to make their animal positions compliant with the EO4wildlife XML definition.

From the ARGOS XML format delivered by the ARGOS processing center, a first adapter can be used to transform the data into the EO4wildlife XML format, and data can then be converted in the EO4wildlife platform into NetCDF so as to minimize the volume of data (XML is very verbose).

The last step can be the NetCDF files ingestion in Oceanotron or in geoserver to be accessed by standard protocols.

In the first version of the EO4wildlife platform, in an Agile approach and to be pragmatic in the development of the platform, the format of the animal track is only CSV. The XML format will be implemented in a future version of the EO4wildlife platform.

A second step will be to generate NetCDF files compliant with the Oceanotron if Oceanotron is confirmed within the use of the project. The other possibility is to ingest tracks into the geoserver of the EO4wildlife platform.

For an integration of data within Oceanotron, see the implementation constraints on http://www.ifremer.fr/isi/oceanotron/site/apidocs, class fr.ifremer.oceanotron.valueObject.ocsml OceanotronTrajectoryFeatureVO.

2 EO4wildlife data access services

2.1 Weather and Oceanographic External products Extraction services

The table below summarizes the end points to access to external datasets.

Product name	Extraction End point
Absolute Dynamic Topography (aka Sea Surface Height above geoid)	http://motu.aviso.altimetry.fr/aviso-gateway-servlet/Motu -s AvisoDT -d dataset-duacs-dt-global-allsat-madt-h
Absolute Dynamic Topography (aka Sea Surface Height above geoid)	http://motu.aviso.altimetry.fr/aviso-gateway-servlet/Motu -s AvisoNRTover30d2014
Bathymetry ETOPO1	Static, in EO4wildlife geoserver
Global Ocean, Ocean Colour Chlorophyll (Optimal Interpolation)	http://cmems.isac.cnr.it/mis-gateway-servlet/Motu -s OCEANCOLOUR_GLO_CHL_L4_NRT_OBSERVATIONS_009_033-TDS -d dataset-oc-glo-chl-multi-l4-oi_4km_daily-rt-v01 -v CHL -v CHL_error
Global Ocean, Ocean Colour Chlorophyll (Optimal Interpolation)	http://cmems.isac.cnr.it/mis-gateway-servlet/Motu -s OCEANCOLOUR_GLO_CHL_L4_REP_OBSERVATIONS_009_082-TDS -d dataset-oc-glo-chl-multi-l4-oi_4km_daily-rep-v01 -v CHL -v CHL_error
Net Primary Production	CLS product used for the validation for EO4wildlife
"Filaments" (convergence/divergence)	Soon in AVISO+
currents (geostrophic) from Sea Level Anomalies	http://motu.aviso.altimetry.fr/aviso-gateway-servlet/Motu -s AvisoDT -d dataset-duacs-dt-global-allsat-msla-uv
currents (geostrophic) from Sea Level Anomalies	http://motu.aviso.altimetry.fr/aviso-gateway-servlet/Motu -s AvisoNRTover30d2014 -d dataset-duacs-nrt-over30d-global-allsat-msla-uv
currents (geostrophic) from Absolute Dynamic Topography	http://motu.aviso.altimetry.fr/aviso-gateway-servlet/Motu -s AvisoDT -d dataset-duacs-dt-global-allsat-madt-uv
currents (geostrophic) from Absolute Dynamic Topography	http://motu.aviso.altimetry.fr/aviso-gateway-servlet/Motu -s AvisoNRTover30d2014 -d dataset-duacs-nrt-over30d-global-allsat-madt-uv
Global Ocean OSTIA Sea Surface Temperature and Sea Ice Analysis	http://cmems.isac.cnr.it/mis-gateway-servlet/Motu -s SST_GLO_SST_L4_NRT_OBSERVATIONS_010_001-TDS -d METOFFICE-GLO-SST-L4-NRT-OBS-SST-V2 -v analysis_error -v sea_ice_fraction -v mask -v analysed_sst
Global Ocean OSTIA Sea Surface Temperature and Sea Ice Reprocessed (1985-2007)	http://cmems.isac.cnr.it/mis-gateway-servlet/Motu -s SST_GLO_SST_L4_REP_OBSERVATIONS_010_011-TDS -d METOFFICE-GLO-SST-L4-RAN-OBS-SST -v analysis_error -v sea_ice_fraction -v mask -v analysed_sst

Global Ocean Wind L4 Delayed Time 6 hourly Observations	http://cmems.isac.cnr.it/mis-gateway-servlet/Motu -s WIND_GLO_WIND_L4_REP_OBSERVATIONS_012_003-TDS -d CERSAT-GLO-REP_WIND_L4-OBS_FULL_TIME_SERIE -v surface_downward_northward_stress_rms -v wind_stress -v land_ice_mask -v eastward_wind -v surface_downward_eastward_stress -v wind_speed -v surface_downward_northward_stress -v sampling_length -v wind_stress_rms -v wind_speed_rms -v surface_downward_eastward_stress_rms -v eastward_wind_rms -v northward_wind -v northward_wind_rms
Global Ocean Wind L4 Near real Time 6 hourly Observations	http://cmems.isac.cnr.it/mis-gateway-servlet/Motu -s WIND_GLO_WIND_L4_NRT_OBSERVATIONS_012_004-TDS -d CERSAT-GLO-BLENDED_WIND_L4-V5-OBS_FULL_TIME_SERIE -v wind_speed_rms -v eastward_wind_rms -v wind_stress -v land_ice_mask -v surface_downward_eastward_stress -v eastward_wind -v sampling_length -v surface_downward_northward_stress -v wind_speed -v northward_wind -v northward_wind_rms
Sea Level Anomalies (aka Sea Surface Height above a mean sea surface)	http://motu.sltac.cls.fr/sltac-gateway-servlet/Motu -s SEALEVEL_GLO_SLA_MAP_L4_NRT_OBSERVATIONS_008_026-TDS -d dataset-duacs-nrt-global-merged-allsat-msla-l4 -v crs -v lon_bnds -v err -v sla -v lat_bnds -v nv
Sea Level Anomalies (aka Sea Surface Height above a mean sea surface)	http://motu.sltac.cls.fr/sltac-gateway-servlet/Motu -s SEALEVEL_GLO_SLA_MAP_L4_REP_OBSERVATIONS_008_027-TDS -d dataset-duacs-rep-global-merged-allsat-msla-l4 -v crs -v lon_bnds -v err -v sla -v lat_bnds -v nv
10 metre U wind component+10 metre V wind component+Mean sea level pressure	Soon in CLS datastore
U wind component at model levels+V wind component at model levels	Soon in CLS datastore

Table 1: End points to access to external datasets

If you use HTTP REST request to extract data with MOTU, please convert the variables, services and datasets options as attributes of the HTTP request.

The result of an extraction request to an external service is a NetCDF file incuding the selected variable on the area of interest for the time period of interest.

2.2 Animal tracks access services

2.2.1 Principles

In the first implementation of the EO4wildlife platform, the user gets the possibility to upload CSV files in his private workspace in the EO4wildlife platform. This upload can be done using an FTP transfer, but can be performed automatically if external platforms, like Birdlife or Sea Turtle platforms, implement this upload functionality.

In the following versions of the EO4wildlife platform, the user will be able to upload Argos data, in existing Argos data formats. The first step will be to convert such private ARGOS formats into the EO4wildlife XML format. This XML will be then converted into NetCDFas a second step for the processing workflow and display via a geoserver or Oceanotron.

The NetCDF content will be updated with all relevant EO4wildlife information for the processing.

Currently, ARGO program drifters are already providing in-situ data which are ingested and disseminated via OCEANOTRON in CMEMS. Their NetCDF files, given in annex C, is a reference that will be enriched for EO4wildlife.

2.2.2 Argos data formats

Current ARGOS PRV, ARGOS DS format, ARGOS XML format can be converted in the EO4wildlife XML thanks to adapters.

2.2.2.1 From CSV to XML

The conversion of CSV exports from Birdlife, Sea Turtle or external platforms into the EO4wildlife XML should be performed by the platforms.

An XML description of the EO4wildlife XML format is provided in annex A.

2.2.2.2 From XML to Netcdf

An adapter is available to convert the EO4wildlife XML into NetCDF.

2.2.2.3 Ingestion in Oceanotron

This possibility is to be confirmed during the implementation phase of the next versions of the EO4wildlife platform.

2.2.3 Other animal tracking data

Note that any animal tracking data, collected through any telecommunication system, can be ingested in EO4wildlife using the EO4wildife XML definition. This adaptation has to be taken into account by the data provider.

3 Management of big data

3.1 Data management

In this section, the ingestion and management of the data, especially the retention policy, on the platform is described. The SparkInData platform provides several components to manage the data on the platform. Concerning the data management, several cases have to be considered depending on the external data source:

- The external data sources provide standard interfaces to access to the metadata and data
- The data has to be manually uploaded into the platform

3.1.1 External data source with standard interfaces

If the external data sources offer standard interface to retrieve the metadata and download the data, a connector can be implemented for the data source considered. This case comprises the Peps and CMEMS data source. In this case, three components take part in the management of this data:

- The watching service
- The ingestion service
- The data management service

The interaction between these different components is described in section 1.3 from the D2.1 System architecture and operational scenarios v1 [2].

The watching service polls regularly the external data source and updates the internal SparkInData catalogue in order for it to be up-to-date with the external catalogue. The internal SparkInData product is represented by the acquisition date, the update time, the version, a list of download URLs that are various alternatives to download the product and the size of the equivalent downloaded product, the credential type to be able to download the product thanks to the provided URL (None, Basic, OAuth, Cas, …), the source and the geometry (Point, LineString, Polygon, BoundingBox) of the product.

The ingestion service allows downloading data referred in the internal SparkInData catalogue into the platform.

The data management service (DMS) manages the data in the platform. When a resource is required by another service, for example by the processing pipeline, it notifies the DMS about this data and a lock is set on the data. The DMS checks if the data is already present on the platform. If the data is already present, it is not downloaded again. If the data is not present, the data ingestion is launched thanks to the ingestion service of the required data based on the download URL given in the metadata associated to this data. When the data is not needed any more by a service, the service notifies the DMS about it and the data is unlocked. The notification to lock and unlock a data is done thanks to requests to the DMS. In case a data is present on the platform and there are no locks on this data, the DMS automatically delete it.

3.1.1.1 CMEMS Connector

A CMEMS product is composed of one or several datasets. A dataset contains various variables (temperature, salinity, wind …) or computation on these variables for a given temporal resolution (daily value, monthly mean, seasonal mean …). A dataset has one URL per type of access. The MOTU Subsetter is

the way to download CMEMS product that will be used because it offers the most flexibility to retrieve a subset of a dataset. In the CMEMS catalogue, the URL is then the generic motu subsetter URL to which some parameters are required to be added to retrieve part of the dataset. The parameters allow filtering depending on the time of interest, area of interest, variables of interest.

On the contrary, a SparkInData product shall be downloadable, that means that the URL for a product should be sufficient to download the product without additional parameters.

For this purpose, a CMEMS product is mapped to several SparkInData products. A SparkInData product corresponds to a subset of a dataset of a CMEMS product that can be downloaded. The finest granularity on the time available for a dataset has been chosen to divide the datasets (one product for each day for a daily dataset, one product for each month for a monthly dataset …). The parameters of the internal SparkInData product are the one defined in the section above.

3.1.1.2 PEPS Connector

A PEPS product is mapped to a product in the SparkInData catalogue. The parameters of the internal SparkInData product are the one defined in the section above.

3.1.2 Data manually uploaded into the platform

In case data are manually uploaded to the platform by a user, the data are stored into the user workspace. Tracking data are part of this case.

A second case of manually uploaded data concerns EO data for which no connectors can be implemented because of the fact that no standard interfaces are available to get access to the data and metadata. In this case, the data can be for example manually uploaded into Geoserver if the data format is supported. This is the case for ETOPO1 Bathymetry data. Moreover, the upload of this data into Geoserver allows accessing this data thanks to WCS in order to determine the area of interest.

3.2 Data access for processing

In this section, the use of both EO and tracking data in processes is described. As detailed in the section above, either the data has been manually uploaded into the platform or when needed to be used in a process, the data has been downloaded by the Data Management Service if not already present on the platform thanks to a request sent to the DMS by the processing pipeline before the effective launch of the WPS process.

In both cases, the data is made accessible through a dedicated URI which is used as input of the process.

This URI corresponds either to a file location on the platform or to an endpoint available on GeoServer.

3.3 Data access for display

GeoServer component provides workspaces management and WFS or WMS endpoints.

Result of a WPS process or any product can be uploaded on GeoServer, and for compliant formats, be accessible through GeoServer WFS or WMS endpoints.

3.4 Data access for download

3.4.1 Data from connectors

Data are inserted into the platform thanks to connectors which upload them in the filesystem, in the private workspace of the user. Then, they can be downloaded by the user from the platform thanks to the download service based on the HTTP protocol.

3.4.2 Gridded data

Any gridded data resulting from an upload or from a process, compliant with Geoserver, can be uploaded to Geoserver and published by Geoserver. It will then be possible to extract the data through WFS, WMS or WCS requests for download.

3.4.3 Tracking data

A download service will be available on the platform to download the tracking data uploaded to the platform. This download service allows downloading file through HTTP protocol.

4 Catalogue

4.1 General description

Catalog Service for the Web (CSW), sometimes seen as Catalog Service - Web, is a standard for exposing a catalogue of geospatial records in XML on the Internet (over HTTP). The catalogue is made up of records that describe geospatial data, geospatial services, and related resources.

CSW is one part (or "profile") of the OGC Catalog Service, which defines common interfaces to discover, browse, and query metadata about data, services, and other potential resources. Version 2.0 of the specification was released in May 2004. The most recent release is 2.0.2, which was published in 2007.

Metadata models are made of the description of XML records compliant with the standard. Typically the records include Dublin Core, ISO 19139 or FGDC metadata, encoded in UTF-8 characters. Each record must contain certain core fields including: Title, Format, Type (e.g. Dataset, DatasetCollection or Service), BoundingBox (a rectangle of interest, expressed in latitude and longitude), Coordinate Reference System, and Association (a link to another metadata record).

Operations defined by the CSW standard include.

- GetCapabilities: "allows CSW clients to retrieve service metadata from a server"
- DescribeRecord: "allows a client to discover elements of the information model supported by the target catalogue service. The operation allows some or all of the information model to be described".
- GetRecords: search for records, returning record IDs
- GetRecordById: "retrieves the default representation of catalogue records using their identifier"
- GetDomain (optional): "used to obtain runtime information about the range of values of a metadata record element or request parameter"
- Harvest (optional): create/update metadata by asking the server to 'pull' metadata from somewhere
- Transaction (optional): create/edit metadata by 'pushing' the metadata to the server

Requests can encode the parameters in three different ways:

- GET with URL parameters
- POST with form-encoded payload
- POST with XML payload

Responses are in XML.

The catalogue candidate in the EO4wildlife platform is GeoNetwork, http://geonetwork-opensource.org. GeoNetwork is used for the storage of EO4wildlife product and service metadata metadata. It is also used by SparkInData Data Management Service, which is responsible for the lifecycle management of the internal products made of extractions producst for processing.

GeoNetwork catalogue comes along with a CSW endpoint for catalogue access from the EO4wildlife front end or to be harvested by external catalogues.

4.2 Metadata model

A metadata schema describes:

1. the names, descriptions and any value codelists of the elements in the metadata schema
2. how the elements of the metadata schema are laid out in a metadata document (structure)
3. constraints on elements and content in a metadata document
4. documentation on how the elements of the metadata schema should be used
5. sample metadata documents and metadata templates
6. scripts to convert metadata documents to and from other metadata schemas

A metadata schema is usually an implementation of a metadata standard.

A metadata profile is an adaptation of a metadata schema to suit the needs of a particular community. A metadata profile contains all the components of a metadata schema but may extend, restrict or redefine these components. Existing metadatmodel compliant with ISO19139/ISO19115-3 and INSPIRE is the strating point to enrich it, if necessary, with EO4wildlife information. See D3.1 Knowledge Base Service architecture Specification v1 [1].

4.2.1 Metadata model for data

The starting point for gridded data is the CMEMS metadata model (See Annex B for the XML schema description). It is already compliant with in situ observations: animal tracking description can be simplified or enriched according to EO4wildlife needs. In the metadata model, the use of “keyword” elements describing both the type of information and the associated value is generic to represent any information.

The Insipre directive is implemented using links to access to the discovery, display and download services.

4.2.2 Metadata model for services

In the EO4wildlife platform, services are standardized using WPS. Once the WPS service is published, a draft of the service metadata is created harvesting the WPS service from the catalogue so as to ensure the consistency between the service definition and its metadata. The service metadata is then enriched to detail the service description.

In case of a service made as a workflow of services, the sevice metadata will include links to each step metadata in the workflow.

The knowledge base implements the workflow definition and the dependencies between services and products or auxiliary data. See D3.1 Knowledge Base Service architecture Specification v1 [1].

4.3 Ontologies

A semantic reconciliation service provides the correspondence between the user terminologies, the variable name to extract, the extracted standard name of the variable from external providers, generally compliant with CF conventions, the standard URI from the vocabulary when it exists.

The ontologies can include a reference to the ISO metadata content in the catalogue for E04wildlife products. Dependencies between products and services can also be represented as linked data.

For more details see D3.1 Knowledge Base Service architecture Specification v1 [1].

4.4 Linked Data Platform API

SparkInData semantic toolkit component provides an implementation of Linked Data Platform which provides management and storage of resources according to the hierarchical data structure defined in LinkedDataPlatform standard.

LDP resources can be either RDF (Resource Description Framework) resources, expressed as triplets, or non-RDF resources (binary or text).

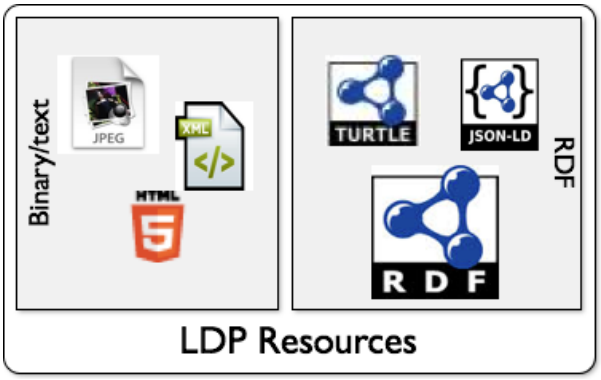

Figure 1: LDP Resources

Storage is performed in two different ways for RDF or non-RDF resources:

- for RDF resource (expressed as triplets): storage is performed into Virtuoso triple store
- For non-RDF resource (binary / text file): storage of a "pointer" is performed inside Virtuoso triple store and effective storage of the resource is done into a NoSQL Database (MongoDB) available in SparkInData components.

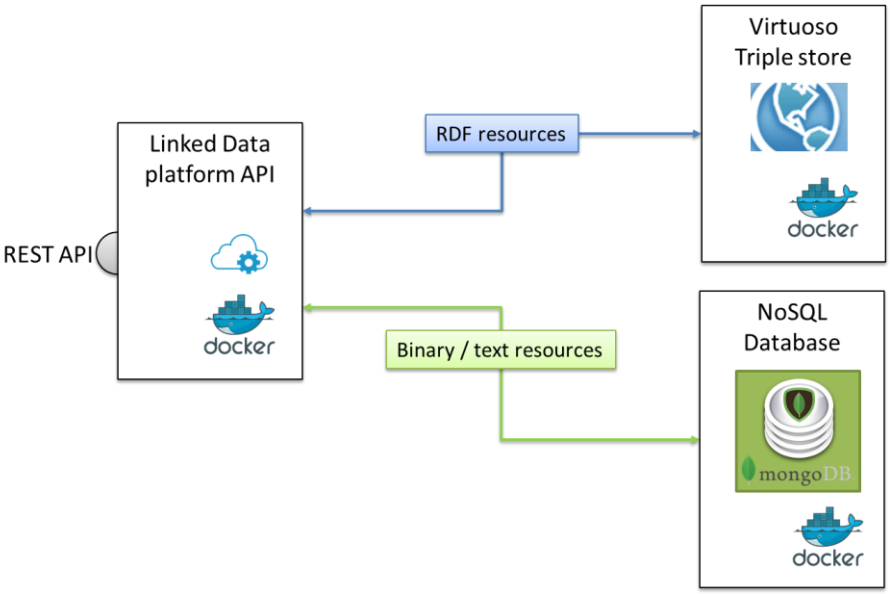

Figure 2: Linked data storage overview

The LDP API provided by SparkInData Semantic toolkit provides the following functionalities:

- Upload a data to the triple store

- Get a resource from the server
- Update a resource (partial or full update)
- Delete a resource
- Retrieve meta-information associated to a resource (
- Get the available options of a resource
- Handling of SparQL queries.

4.5 Virtuoso

Virtuoso, also known as Virtuoso Universal Server, is a multi-protocol Relational Database management system (RDBMS) that includes an object-relational database engine for SQL, XML, RDF, and free-text.

It also includes JAVA and .NET runtime hosting, Web Application Server, Web Services, Web Content Management, Data portability such as controlling, sharing, and moving data freely from system to system.

Virtuoso is deployed as a storage module into SparkInData semantic toolkit through a Docker container deployed in a Kubernetes Pod.

In EO4wildlife context, Virtuoso is used as a triple store for RDF resource storage and is accessible through Linked Data Platform API for RDF resource management and SPARQL queries.

5 Deliverables

5.1 Extraction

In section 3.4, two extraction mechanisms have been detailed depending on the data.

The first case handles data that can be published into Geoserver and extract from the platform through WFS, WMS or WCS request. All the information about Geoserver can be found at http://docs.geoserver.org/2.8.1/user and http://docs.geoserver.org/2.8.1/developer. The installation of a geoserver is described in the geoserver website, http://docs.geoserver.org/stable/en/user/installation.

The second case handles all others data that means data that cannot be uploaded into Geoserver, data coming from connectors or uploaded by the user into his workspace. These data can be downloaded from the platform thanks to a dedicated EO4wildlife component. This component is installed as part of the EO4wildlife platform v2.

5.2 OCEANOTRON

Oceanotron is an opensource component. The entire document for the installations is available on ftp://ftp.ifremer.fr/ifremer/sismer/oceanotron/doc.

5.3 GeoNetwork

GeoNetwork is the component provided with the SparkInData platform for product and service metadata management. More information about the installation of this component can be found at http://geonetwork-opensource.org/manuals/trunk/eng/users/overview/index.html.

5.4 Linked Open Data API

SparkInData semantic toolkit component provides an implementation of Linked Data Platform which provides management and storage of resources according to the hierarchical data structure defined in LinkedDataPlatform standard. More information about Linked Data Platform can be found at https://www.w3.org/TR/ldp.

5.5 Virtuoso

All information about Virtuoso, also known as Virtuoso Universal Server, can be found at https://virtuoso.openlinksw.com.

6 Conclusion

OGC standards, INSPIRE directive and opensource software components were the main drivers to find an efficient solution to provide a standardized way to connect the EO4wildlife platform to a wide diversity of data and services.

Standardized interfaces provide an interoperable access to the EO4wildlife platform, not only for the EO4wildlife components implementation, such as the front end component, but also to make easier the connectivity of EO4wildlife end users platforms. Moreover, any scientist will be able to discover and test the EO4wildlife services and products by a simple adaptation of its own data to the EO4wildlife Canonical Data Format.

Existing and robust components from the CMEMS service or from various open source communities are reused to implement these interfaces and prepare the EO4wildlife platform to the operational phase.

References

[1] G. Correndo, J-M Zigna, A. Haugommard, 'D3.1 Knowledge Base Service architecture Specification v1', Deliverable of EO4wildlife project, 2016

[2] A. Haugommard, F. Martin, and D. Rodera, 'D2.1 System architecture and operational scenarios v1', Deliverable of the EO4wildlife project, 2016

Annex A: EO4wildlife XML description

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <!-- DATA DECLARATION -->
    <xs:element name="data">

        <xs:complexType>
            <xs:sequence>
                <xs:element name="datasets" type="datasets" minOccurs="1"
maxOccurs="unbounded"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>


    <!-- DATASETS DECLARATION -->
    <xs:complexType name="datasets">
        <xs:sequence>
            <xs:element name="dataset">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="coOwners" type="owner" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element name="species" type="species" minOccurs="0"/>
                        <xs:element name="programs" type="programs" minOccurs="1"
                            maxOccurs="unbounded"/>
                    </xs:sequence>
                    <xs:attribute name="name" type="xs:string"/>
                    <xs:attribute name="version" type="xs:string"/>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
        <xs:attribute name="count" type="xs:int"/>
    </xs:complexType>

    <!-- OWNER DECLARATION -->
    <xs:complexType name="owner">
        <xs:sequence>
            <xs:element name="name" type="xs:string" minOccurs="0"/>
            <xs:element name="address" type="xs:string" minOccurs="0"/>
            <xs:element name="mailAddress" type="xs:string" minOccurs="0"/>
            <xs:element name="affiliation" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <!-- SPECIES DECLARATION -->
    <xs:complexType name="species">
        <xs:sequence>
            <xs:element name="name" type="xs:string" minOccurs="0"/>
            <xs:element name="vernacularName" type="xs:string" minOccurs="0"/>
            <xs:element name="authorityAccepted" type="xs:string" minOccurs="0"/>
            <xs:element name="kingdom" type="xs:string" minOccurs="0"/>
            <xs:element name="phylum" type="xs:string" minOccurs="0"/>
            <xs:element name="classname" type="xs:string" minOccurs="0"/>
```

```
            <xs:element name="ordername" type="xs:string" minOccurs="0"/>
            <xs:element name="family" type="xs:string" minOccurs="0"/>
            <xs:element name="genus" type="xs:string" minOccurs="0"/>
            <xs:element name="redListCategory" type="redListCategoryEnum"
minOccurs="0"/>
            <xs:element name="habitat" type="habitatEnum" minOccurs="0"/>
            <xs:element name="altitude" type="xs:double" minOccurs="0"/>
            <xs:element name="migratoryStatus" type="migratoryStatusEnum"
minOccurs="0"/>
            <xs:element name="threat" type="threatEnum" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <!-- DEPLOYMENT LOCATION DECLARATION -->
    <xs:complexType name="deploymentInformation">
        <xs:sequence>
            <!-- GMT -->
            <xs:element name="dateTime" type="xs:dateTime" minOccurs="0"/>
            <!-- range : -90Â°..+90Â° -->
            <xs:element name="latitude" type="xs:double" minOccurs="0"/>
            <!-- range : -180Â°..+180Â° -->
            <xs:element name="longitude" type="xs:double" minOccurs="0"/>
            <!-- unit : meter -->
            <xs:element name="altitude" type="xs:double" minOccurs="0"/>
            <xs:element name="place" type="xs:string" minOccurs="0"/>
            <xs:element name="tagger" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <!-- DETACHMENT DECLARATION -->
    <xs:complexType name="detachmentInformation">
        <xs:sequence>
            <!-- GMT -->
            <xs:element name="dateTime" type="xs:dateTime" minOccurs="0"/>
            <!-- range : -90Â°..+90Â° -->
            <xs:element name="latitude" type="xs:double" minOccurs="0"/>
            <!-- range : -180Â°..+180Â° -->
            <xs:element name="longitude" type="xs:double" minOccurs="0"/>
            <!-- unit : meter -->
            <xs:element name="altitude" type="xs:double" minOccurs="0"/>
            <xs:element name="place" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <!-- PLATFORMS DECLARATION -->
    <xs:complexType name="programs">
        <xs:sequence>
            <xs:element name="program" type="program" minOccurs="1"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="count" type="xs:int"/>
    </xs:complexType>

    <!-- PROGRAM DECLARATION -->
    <xs:complexType name="program">
        <xs:sequence>
            <xs:element name="programName" type="xs:string" minOccurs="0"/>
```

```
            <xs:element name="programNumber" type="xs:int" minOccurs="0"/>
            <xs:element name="country" type="xs:string" minOccurs="0"/>
            <xs:element name="countractor" type="xs:string" minOccurs="0"/>
            <xs:element name="platforms" type="platforms" minOccurs="1"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>

    <!-- PLATFORMS DECLARATION -->
    <xs:complexType name="platforms">
        <xs:sequence>
            <xs:element name="platform" type="platform" minOccurs="1"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="count" type="xs:int"/>
    </xs:complexType>

    <!-- PLATFORM DECLARATION -->
    <xs:complexType name="platform">
        <xs:sequence>
            <xs:element name="platformId" type="xs:int" minOccurs="0"/>
            <xs:element name="platformType" type="xs:string" minOccurs="0"/>
            <xs:element name="platformName" type="xs:string" minOccurs="0"/>
            <xs:element name="platformModel" type="xs:string" minOccurs="0"/>
            <xs:element name="platformManufacturer" type="xs:string" minOccurs="0"/>
            <xs:element name="platformHexId" type="xs:string" minOccurs="0"/>
            <xs:element name="animal" type="animal" minOccurs="0" maxOccurs="1"/>
            <xs:element name="deploymentInformation" type="deploymentInformation"
minOccurs="0"
                maxOccurs="1"/>
            <xs:element name="detachmentInformation" type="detachmentInformation"
minOccurs="0"
                maxOccurs="1"/>
            <xs:element name="observations" type="observations" minOccurs="1"/>
        </xs:sequence>
    </xs:complexType>

    <!-- OBSERVATIONS DECLARATION -->
    <xs:complexType name="observations">
        <xs:sequence>
            <xs:element name="observation" type="observation" minOccurs="1"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="count" type="xs:int"/>
    </xs:complexType>


    <!-- ANIMAL DECLARATION -->
    <xs:complexType name="animal">
        <xs:sequence>
            <xs:element name="animalId" type="xs:int" minOccurs="0"/>
            <xs:element name="sex" type="sexEnum" minOccurs="0"/>
            <xs:element name="age" type="ageEnum" minOccurs="0"/>
            <xs:element name="breedStage" type="breedEnum" minOccurs="0"/>
            <xs:element name="length" type="xs:float" minOccurs="0"/>
            <xs:element name="lengthUnit" type="xs:string" minOccurs="0"/>
```

```
            <xs:element name="mass" type="xs:float" minOccurs="0"/>
            <xs:element name="massUnit" type="xs:string" minOccurs="0"/>
            <xs:element name="girth" type="xs:float" minOccurs="0"/>
            <xs:element name="girthUnit" type="xs:string" minOccurs="0"/>
            <!-- wild or hatchery -->
            <xs:element name="hatchery" type="xs:boolean" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <!-- RED LIST CATEGORY DECLARATION -->
    <xs:simpleType name="redListCategoryEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="EXTINCT (EX)"/>
            <xs:enumeration value="EXTINCT IN THE WILD (EW)"/>
            <xs:enumeration value="CRITICALLY ENDANGERED (POSSIBLY EXTINCT) CR (PE)"/>
            <xs:enumeration value="CRITICALLY ENDANGERED (CR)"/>
            <xs:enumeration value="ENDANGERED (EN)"/>
            <xs:enumeration value="VULNERABLE (VU)" />
            <xs:enumeration value="NEAR THREATENED (NT)" />
            <xs:enumeration value="LEAST CONCERN (LC)"/>
            <xs:enumeration value="DATA DEFICIENT (DD)" />
            <xs:enumeration value="NOT EVALUATED (NE)" />
            <xs:enumeration value="NOT RECOGNISED (NR)" />
        </xs:restriction>
    </xs:simpleType>

    <!-- HABITAT CATEGORY DECLARATION -->
    <xs:simpleType name="habitatEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Forest"/>
            <xs:enumeration value="Savanna"/>
            <xs:enumeration value="Shrubland)"/>
            <xs:enumeration value="Grassland"/>
            <xs:enumeration value="Wetlands (inland)"/>
            <xs:enumeration value="Rocky areas" />
            <xs:enumeration value="Caves and subterranean habitats (non-aquatic)" />
            <xs:enumeration value="Desert"/>
            <xs:enumeration value="Sea" />
            <xs:enumeration value="Coastline" />
            <xs:enumeration value="Artificial landscapes (terrestrial)" />
            <xs:enumeration value="Artificial landscapes (aquatic)" />
            <xs:enumeration value="Introduced/exotic vegetation" />
            <xs:enumeration value="Other" />
            <xs:enumeration value="Unknown" />
        </xs:restriction>
    </xs:simpleType>

    <!-- MIGRATORY STATUS DECLARATION -->
    <xs:simpleType name="migratoryStatusEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Nomadic"/>
            <xs:enumeration value="Migratory"/>
            <xs:enumeration value="Altitudinal migrant"/>
            <xs:enumeration value="Non-migratory"/>
        </xs:restriction>
    </xs:simpleType>
```

```
    <!-- THREAT DECLARATION -->
    <xs:simpleType name="threatEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="Residential &amp; commercial development"/>
            <xs:enumeration value="Agriculture &amp; aquaculture)"/>
            <xs:enumeration value="Energy production &amp; mining"/>
            <xs:enumeration value="Transportation &amp; service corridors"/>
            <xs:enumeration value="Biological resource use"/>
            <xs:enumeration value="Human intrusions &amp; disturbance" />
            <xs:enumeration value="Natural system modifications" />
            <xs:enumeration value="Invasive &amp; other problematic species &amp;
genes"/>
            <xs:enumeration value="Pollution" />
            <xs:enumeration value="Geological events" />
            <xs:enumeration value="Climate change &amp; severe weather" />
            <xs:enumeration value="Other" />
        </xs:restriction>
    </xs:simpleType>

    <!-- SEX DECLARATION -->
    <xs:simpleType name="sexEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="male"/>
            <xs:enumeration value="female"/>
            <xs:enumeration value="unknown"/>
        </xs:restriction>
    </xs:simpleType>

    <!-- AGE DECLARATION -->
    <xs:simpleType name="ageEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="adult"/>
            <xs:enumeration value="sub-adult"/>
            <xs:enumeration value="yearling"/>
            <xs:enumeration value="immature"/>
            <xs:enumeration value="juvenile"/>
            <xs:enumeration value="fledgling"/>
            <xs:enumeration value="unknown"/>
        </xs:restriction>
    </xs:simpleType>

    <!-- BREED DECLARATION -->
    <xs:simpleType name="breedEnum">
        <xs:restriction base="xs:string">
            <xs:enumeration value="pre-egg"/>
            <xs:enumeration value="incubation"/>
            <xs:enumeration value="brood-guard"/>
            <xs:enumeration value="post-guard"/>
            <xs:enumeration value="chick-rearing"/>
            <xs:enumeration value="creche"/>
            <xs:enumeration value="breeding"/>
            <xs:enumeration value="fail(breeding)"/>
            <xs:enumeration value="migration"/>
            <xs:enumeration value="winter"/>
            <xs:enumeration value="sabbatical"/>
            <xs:enumeration value="pre-moult"/>
            <xs:enumeration value="moulting"/>
```

```
            <xs:enumeration value="non-breeding"/>
            <xs:enumeration value="unknown"/>
        </xs:restriction>
    </xs:simpleType>


    <!-- observation DECLARATION -->
    <xs:complexType name="observation">
        <xs:sequence>
            <!-- GMT -->
            <xs:element name="ArgosLocation" type="ArgosLocation" minOccurs="0"/>
            <xs:element name="GPSLocation" type="GPSLocation" minOccurs="0"/>
            <xs:element name="otherLocation" type="otherLocation" minOccurs="0"/>
            <xs:element name="bestDate" type="xs:dateTime" minOccurs="0"/>
            <xs:element name="equinoxe" type="xs:boolean" minOccurs="0"/>
            <xs:element name="sensors" type="sensors" minOccurs="0"
maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>


    <!-- SENSORS DECLARATION -->
    <xs:complexType name="sensors">
        <xs:sequence>
            <xs:element name="sensor" type="sensor" minOccurs="1"
maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:attribute name="count" type="xs:int"/>
    </xs:complexType>

    <!-- ARGOS LOCATION DECLARATION -->
    <xs:complexType name="ArgosLocation">
        <xs:sequence>
            <!-- GMT -->
            <xs:element name="locationDate" type="xs:dateTime" minOccurs="0"/>
            <!-- range : -90Â°..+90Â° -->
            <xs:element name="latitude" type="xs:double" minOccurs="0"/>
            <!-- range : -180Â°..+180Â° -->
            <xs:element name="longitude" type="xs:double" minOccurs="0"/>
            <!-- unit : meter -->
            <xs:element name="altitude" type="xs:double" minOccurs="0"/>
            <xs:element name="locationClass" type="xs:string" minOccurs="0"/>
            <!-- unit : meter/second -->
            <xs:element name="diagnostic" type="diagnostic" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <!-- GPS LOCATION DECLARATION -->
    <xs:complexType name="GPSLocation">
        <xs:sequence>
            <!-- GMT -->
            <xs:element name="locationDate" type="xs:dateTime" minOccurs="0"/>
            <!-- range : -90Â°..+90Â° -->
            <xs:element name="latitude" type="xs:double" minOccurs="0"/>
            <!-- range : -180Â°..+180Â° -->
            <xs:element name="longitude" type="xs:double" minOccurs="0"/>
            <!-- unit : meter -->
```

```
            <xs:element name="altitude" type="xs:double" minOccurs="0"/>
            <!-- unit : meter/second -->
            <xs:element name="gpsSpeed" type="xs:float" minOccurs="0"/>
            <!-- range : 0Â°..+360Â°, clockwise -->
            <xs:element name="gpsHeading" type="xs:float" minOccurs="0"/>
            <xs:element name="diagnostic" type="diagnostic" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

    <!-- OTHER LOCATION DECLARATION -->
    <xs:complexType name="otherLocation">
        <xs:sequence>
            <!-- GMT -->
            <xs:element name="locationDate" type="xs:dateTime" minOccurs="0"/>
            <!-- range : -90Â°..+90Â° -->
            <xs:element name="latitude" type="xs:double" minOccurs="0"/>
            <!-- range : -180Â°..+180Â° -->
            <xs:element name="longitude" type="xs:double" minOccurs="0"/>
            <!-- unit : meter -->
            <xs:element name="altitude" type="xs:double" minOccurs="0"/>
            <!-- device -->
            <xs:element name="deviceName" type="xs:string" minOccurs="0"/>
            <!-- device -->
            <xs:element name="deviceType" type="xs:string" minOccurs="0"/>
            <!-- diagnostic -->
            <xs:element name="diagnostic" type="diagnostic" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>

     <!-- DIAGNOSTIC DECLARATION -->
    <xs:complexType name="diagnostic">
        <xs:sequence>
            <!-- range : -90Â°..+90Â° -->
            <xs:element name="latitude2" type="xs:float" minOccurs="0"/>
            <!-- range : -180Â°..+180Â° -->
            <xs:element name="longitude2" type="xs:float" minOccurs="0"/>
            <!-- unit : meter -->
            <xs:element name="altitude2" type="xs:float" minOccurs="0"/>
            <xs:element name="index" type="xs:int" minOccurs="0"/>
            <xs:element name="nopc" type="xs:int" minOccurs="0"/>
            <!-- unit : meter -->
            <xs:element name="errorRadius" type="xs:float" minOccurs="0"/>
            <!-- unit : meter -->
            <xs:element name="semiMajor" type="xs:float" minOccurs="0"/>
            <!-- unit : meter -->
            <xs:element name="semiMinor" type="xs:float" minOccurs="0"/>
            <!-- range 0Â°..180Â°, clockwise -->
            <xs:element name="orientation" type="xs:float" minOccurs="0"/>
            <xs:element name="hdop" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>


  <!-- SENSOR DECLARATION -->
    <xs:complexType name="sensor">
        <xs:sequence>
            <xs:element name="order" type="xs:int" minOccurs="0"/>
```

```
            <xs:element name="name" type="xs:string" minOccurs="0"/>
            <xs:element name="valueType" type="xs:string" minOccurs="0"/>
            <xs:element name="value" type="xs:string" minOccurs="0"/>
            <xs:element name="valueStr" type="xs:string" minOccurs="0"/>
            <xs:element name="unit" type="xs:string" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>


</xs:schema>
```

Annex B: CMEMS products metadata model

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
targetNamespace="http://www.isotc211.org/2005/gmd"
xmlns:gmx="http://www.isotc211.org/2005/gmx"
xmlns:gco="http://www.isotc211.org/2005/gco" xmlns:gml="http://www.opengis.net/gml"
xmlns:gmd="http://www.isotc211.org/2005/gmd"
xmlns:xlink="http://www.w3.org/1999/xlink">
  <xs:import namespace="http://www.isotc211.org/2005/gco" schemaLocation="gco.xsd"/>
  <xs:import namespace="http://www.isotc211.org/2005/gmx" schemaLocation="gmx.xsd"/>
  <xs:import namespace="http://www.opengis.net/gml" schemaLocation="gml.xsd"/>
  <xs:import namespace="http://www.w3.org/1999/xlink" schemaLocation="xlink.xsd"/>
  <xs:element name="MD_Metadata">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:fileIdentifier"/>
        <xs:element ref="gmd:language"/>
        <xs:element ref="gmd:characterSet"/>
        <xs:element ref="gmd:hierarchyLevel"/>
        <xs:element ref="gmd:hierarchyLevelName"/>
        <xs:element ref="gmd:contact"/>
        <xs:element ref="gmd:dateStamp"/>
        <xs:element ref="gmd:metadataStandardName"/>
        <xs:element ref="gmd:metadataStandardVersion"/>
        <xs:element ref="gmd:referenceSystemInfo"/>
        <xs:element ref="gmd:identificationInfo"/>
        <xs:element maxOccurs="unbounded" ref="gmd:contentInfo"/>
        <xs:element ref="gmd:distributionInfo"/>
        <xs:element ref="gmd:dataQualityInfo"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="fileIdentifier">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="characterSet">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_CharacterSetCode"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_CharacterSetCode">
    <xs:complexType>
      <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
      <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="hierarchyLevel" type="gmd:gmd.MD_ScopeCode"/>
  <xs:element name="hierarchyLevelName">
```

```
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="contact" type="gmd:gmd.CI_ResponsibleParty"/>
  <xs:element name="dateStamp">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:DateTime"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="metadataStandardName">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="metadataStandardVersion">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="referenceSystemInfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_ReferenceSystem"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_ReferenceSystem">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:referenceSystemIdentifier"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="referenceSystemIdentifier">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:RS_Identifier"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="RS_Identifier">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="gmd:gmd.code">
          <xs:sequence>
            <xs:element ref="gmd:codeSpace"/>
          </xs:sequence>
        </xs:extension>
```

```
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="codeSpace">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="identificationInfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_DataIdentification"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_DataIdentification">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:citation"/>
        <xs:element ref="gmd:abstract"/>
        <xs:element ref="gmd:credit"/>
        <xs:element maxOccurs="unbounded" ref="gmd:pointOfContact"/>
        <xs:element ref="gmd:resourceMaintenance"/>
        <xs:element ref="gmd:graphicOverview"/>
        <xs:element maxOccurs="unbounded" ref="gmd:descriptiveKeywords"/>
        <xs:element maxOccurs="unbounded" ref="gmd:resourceConstraints"/>
        <xs:element maxOccurs="unbounded" ref="gmd:aggregationInfo"/>
        <xs:element ref="gmd:spatialResolution"/>
        <xs:element ref="gmd:language"/>
        <xs:element ref="gmd:topicCategory"/>
        <xs:element ref="gmd:extent"/>
        <xs:element ref="gmd:supplementalInformation"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="citation" type="gmd:gmd.CI_Citation"/>
  <xs:element name="abstract">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="credit">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="pointOfContact" type="gmd:gmd.CI_ResponsibleParty"/>
  <xs:element name="resourceMaintenance">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_MaintenanceInformation"/>
```

```
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_MaintenanceInformation">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:maintenanceAndUpdateFrequency"/>
        <xs:element ref="gmd:updateScopeDescription"/>
        <xs:element ref="gmd:maintenanceNote"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="maintenanceAndUpdateFrequency">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_MaintenanceFrequencyCode"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_MaintenanceFrequencyCode">
    <xs:complexType>
      <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
      <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="updateScopeDescription">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_ScopeDescription"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_ScopeDescription">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:other"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="other">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="maintenanceNote">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="graphicOverview">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_BrowseGraphic"/>
```

```
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_BrowseGraphic">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:fileName"/>
        <xs:element ref="gmd:fileDescription"/>
        <xs:element ref="gmd:fileType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="fileName">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="fileDescription">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="fileType">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="descriptiveKeywords">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_Keywords"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_Keywords">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="gmd:keyword"/>
        <xs:element ref="gmd:type"/>
        <xs:element ref="gmd:thesaurusName"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="keyword">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmx:Anchor"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="type">
```

```
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_KeywordTypeCode"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_KeywordTypeCode">
    <xs:complexType>
      <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
      <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="thesaurusName" type="gmd:gmd.CI_Citation"/>
  <xs:element name="resourceConstraints">
    <xs:complexType>
      <xs:choice>
        <xs:element ref="gmd:MD_Constraints"/>
        <xs:element ref="gmd:MD_LegalConstraints"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_Constraints">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:useLimitation"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="useLimitation">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_LegalConstraints">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:accessConstraints"/>
        <xs:element ref="gmd:otherConstraints"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="accessConstraints">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_RestrictionCode"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_RestrictionCode">
    <xs:complexType>
      <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
      <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="otherConstraints">
```

```
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="aggregationInfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_AggregateInformation"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_AggregateInformation">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:aggregateDataSetIdentifier"/>
        <xs:element ref="gmd:associationType"/>
        <xs:element ref="gmd:initiativeType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="aggregateDataSetIdentifier" type="gmd:gmd.MD_Identifier"/>
  <xs:element name="associationType">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:DS_AssociationTypeCode"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DS_AssociationTypeCode">
    <xs:complexType>
      <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
      <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="initiativeType">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:DS_InitiativeTypeCode"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DS_InitiativeTypeCode">
    <xs:complexType>
      <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
      <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="spatialResolution">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_Resolution"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_Resolution">
```

```
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:distance"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="distance">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:Distance"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="topicCategory">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_TopicCategoryCode"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_TopicCategoryCode" type="xs:NCName"/>
  <xs:element name="supplementalInformation">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="contentInfo">
    <xs:complexType>
      <xs:choice>
        <xs:element ref="gmd:MD_CoverageDescription"/>
        <xs:element ref="gmd:MD_FeatureCatalogueDescription"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_CoverageDescription">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:attributeDescription"/>
        <xs:element ref="gmd:contentType"/>
        <xs:element maxOccurs="unbounded" ref="gmd:dimension"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="attributeDescription">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:RecordType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="contentType">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_CoverageContentTypeCode"/>
      </xs:sequence>
```

```
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_CoverageContentTypeCode">
    <xs:complexType>
      <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
      <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="dimension">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_RangeDimension"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_RangeDimension">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:descriptor"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="descriptor">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_FeatureCatalogueDescription">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:includedWithDataset"/>
        <xs:element ref="gmd:featureTypes"/>
        <xs:element ref="gmd:featureCatalogueCitation"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="includedWithDataset">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:Boolean"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="featureTypes">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:LocalName"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="featureCatalogueCitation">
    <xs:complexType/>
  </xs:element>
  <xs:element name="distributionInfo">
    <xs:complexType>
```

```
      <xs:sequence>
        <xs:element ref="gmd:MD_Distribution"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_Distribution">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="gmd:distributor"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="distributor">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_Distributor"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_Distributor">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:distributorContact"/>
        <xs:element ref="gmd:distributorFormat"/>
        <xs:element ref="gmd:distributorTransferOptions"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="distributorContact">
    <xs:complexType/>
  </xs:element>
  <xs:element name="distributorFormat">
    <xs:complexType/>
  </xs:element>
  <xs:element name="distributorTransferOptions">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:MD_DigitalTransferOptions"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MD_DigitalTransferOptions">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:onLine"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="onLine" type="gmd:gmd.CI_OnlineResource"/>
  <xs:element name="dataQualityInfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:DQ_DataQuality"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DQ_DataQuality">
```

```
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:scope"/>
        <xs:element ref="gmd:report"/>
        <xs:element ref="gmd:lineage"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="scope">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:DQ_Scope"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DQ_Scope">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:level"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="level" type="gmd:gmd.MD_ScopeCode"/>
  <xs:element name="report">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:DQ_DomainConsistency"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DQ_DomainConsistency">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:result"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="result">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:DQ_ConformanceResult"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="DQ_ConformanceResult">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:specification"/>
        <xs:element ref="gmd:explanation"/>
        <xs:element ref="gmd:pass"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="specification" type="gmd:gmd.CI_Citation"/>
  <xs:element name="explanation">
    <xs:complexType>
      <xs:sequence>
```

```
      <xs:element ref="gco:CharacterString"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="pass">
  <xs:complexType>
    <xs:attribute ref="gco:nilReason" use="required"/>
  </xs:complexType>
</xs:element>
<xs:element name="lineage">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="gmd:LI_Lineage"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="LI_Lineage">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="gmd:statement"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="statement">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="gco:CharacterString"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="language">
  <xs:complexType>
    <xs:choice>
      <xs:element ref="gco:CharacterString"/>
      <xs:element ref="gmd:LanguageCode"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
<xs:element name="LanguageCode">
  <xs:complexType>
    <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
    <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
  </xs:complexType>
</xs:element>
<xs:complexType name="gmd.MD_ScopeCode">
  <xs:sequence>
    <xs:element ref="gmd:MD_ScopeCode"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="MD_ScopeCode">
  <xs:complexType>
    <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
    <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
  </xs:complexType>
</xs:element>
<xs:complexType name="gmd.CI_ResponsibleParty">
  <xs:sequence>
```

```
      <xs:element ref="gmd:CI_ResponsibleParty"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="CI_ResponsibleParty">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" ref="gmd:individualName"/>
        <xs:element ref="gmd:organisationName"/>
        <xs:element ref="gmd:contactInfo"/>
        <xs:element ref="gmd:role"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="individualName">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
      <xs:attribute ref="gco:nilReason"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="organisationName">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="contactInfo">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:CI_Contact"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CI_Contact">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:address"/>
        <xs:element minOccurs="0" ref="gmd:onlineResource"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="address">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:CI_Address"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CI_Address">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:electronicMailAddress"/>
      </xs:sequence>
    </xs:complexType>
```

```
  </xs:element>
  <xs:element name="electronicMailAddress">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="onlineResource" type="gmd:gmd.CI_OnlineResource"/>
  <xs:element name="role">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:CI_RoleCode"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CI_RoleCode">
    <xs:complexType>
      <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
      <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="gmd.code">
    <xs:sequence>
      <xs:element ref="gmd:code"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="code">
    <xs:complexType>
      <xs:choice>
        <xs:element ref="gco:CharacterString"/>
        <xs:element ref="gmx:Anchor"/>
      </xs:choice>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="gmd.CI_Citation">
    <xs:sequence>
      <xs:element ref="gmd:CI_Citation"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="CI_Citation">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:title"/>
        <xs:element minOccurs="0" ref="gmd:alternateTitle"/>
        <xs:element ref="gmd:date"/>
        <xs:sequence minOccurs="0">
          <xs:element ref="gmd:edition"/>
          <xs:element ref="gmd:editionDate"/>
        </xs:sequence>
        <xs:element minOccurs="0" ref="gmd:identifier"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="title">
    <xs:complexType>
      <xs:sequence>
```

```
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="alternateTitle">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="edition">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="editionDate">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:Date"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="identifier" type="gmd:gmd.MD_Identifier"/>
  <xs:complexType name="gmd.MD_Identifier">
    <xs:sequence>
      <xs:element ref="gmd:MD_Identifier"/>
    </xs:sequence>
  </xs:complexType>
  <xs:element name="MD_Identifier" type="gmd:gmd.code"/>
  <xs:element name="extent">
    <xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element ref="gmd:EX_Extent"/>
        <xs:element ref="gml:TimePeriod"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="EX_Extent">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:description"/>
        <xs:element ref="gmd:geographicElement"/>
        <xs:element ref="gmd:temporalElement"/>
        <xs:element ref="gmd:verticalElement"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="geographicElement">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:EX_GeographicBoundingBox"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

```
<xs:element name="EX_GeographicBoundingBox">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="gmd:westBoundLongitude"/>
      <xs:element ref="gmd:eastBoundLongitude"/>
      <xs:element ref="gmd:southBoundLatitude"/>
      <xs:element ref="gmd:northBoundLatitude"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="westBoundLongitude" type="gmd:gco.Decimal"/>
<xs:element name="eastBoundLongitude" type="gmd:gco.Decimal"/>
<xs:element name="southBoundLatitude" type="gmd:gco.Decimal"/>
<xs:element name="northBoundLatitude" type="gmd:gco.Decimal"/>
<xs:element name="temporalElement">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="gmd:EX_TemporalExtent"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="EX_TemporalExtent">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="gmd:extent"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="verticalElement">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="gmd:EX_VerticalExtent"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="EX_VerticalExtent">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="gmd:minimumValue"/>
      <xs:element ref="gmd:maximumValue"/>
      <xs:element ref="gmd:verticalCRS"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="minimumValue" type="gmd:gco.Real"/>
<xs:element name="maximumValue" type="gmd:gco.Real"/>
<xs:element name="verticalCRS">
  <xs:complexType>
    <xs:attribute name="uuidref" use="required" type="xs:NMTOKEN"/>
  </xs:complexType>
</xs:element>
<xs:complexType name="gmd.CI_OnlineResource">
  <xs:sequence>
    <xs:element ref="gmd:CI_OnlineResource"/>
  </xs:sequence>
</xs:complexType>
<xs:element name="CI_OnlineResource">
```

```
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:linkage"/>
        <xs:sequence minOccurs="0">
          <xs:element ref="gmd:protocol"/>
          <xs:element ref="gmd:name"/>
          <xs:element ref="gmd:description"/>
          <xs:element ref="gmd:function"/>
        </xs:sequence>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="linkage">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:URL"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="URL" type="xs:anyURI"/>
  <xs:element name="protocol">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="name">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="function">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:CI_OnLineFunctionCode"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CI_OnLineFunctionCode">
    <xs:complexType>
      <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
      <xs:attribute name="codeListValue" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="date">
    <xs:complexType>
      <xs:sequence minOccurs="0">
        <xs:element ref="gmd:CI_Date"/>
        <xs:element ref="gco:Date"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CI_Date">
    <xs:complexType>
```

```
      <xs:sequence>
        <xs:element ref="gmd:date"/>
        <xs:element ref="gmd:dateType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="dateType">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gmd:CI_DateTypeCode"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="CI_DateTypeCode">
    <xs:complexType>
      <xs:attribute name="codeList" use="required" type="xs:anyURI"/>
      <xs:attribute name="codeListValue" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="description">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="gco:CharacterString"/>
      </xs:sequence>
      <xs:attribute ref="gco:nilReason"/>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="gco.Decimal">
    <xs:sequence>
      <xs:element ref="gco:Decimal"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="gco.Real">
    <xs:sequence>
      <xs:element ref="gco:Real"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Annex C: ARGO NetCDF description

Dataset type: NetCDF-3/CDM

NetCDF Header (variables and global attributes):

```
{
  dimensions:
    TIME = UNLIMITED;   // (7 currently
    DEPTH = 16;
    LATITUDE = 7;
    LONGITUDE = 7;
    POSITION = 7;
    STRING32 = 32;
    STRING256 = 256;
  variables:
    double TIME(TIME=7);
      :long_name = "time";
      :standard_name = "time";
      :_FillValue = 999999.0; // double
      :units = "days since 1950-01-01T00:00:00Z";
      :valid_min = 0.0; // double
      :valid_max = 90000.0; // double
      :QC_indicator = 1; // int
      :QC_procedure = 1; // int
      :uncertainty = " ";
      :comment = " ";
      :axis = "T";

    byte TIME_QC(TIME=7);
      :long_name = "quality flag";
      :conventions = "OceanSites reference table 2";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 9B; // byte
      :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
      :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";

    float LATITUDE(LATITUDE=7);
      :long_name = "Latitude of each location";
      :standard_name = "latitude";
      :_FillValue = 99999.0f; // float
      :units = "degrees_north";
      :valid_min = -90.0; // double
      :valid_max = 90.0; // double
      :QC_indicator = 1; // int
      :QC_procedure = 1; // int
      :uncertainty = " ";
      :comment = " ";
      :axis = "Y";

    float LONGITUDE(LONGITUDE=7);
      :long_name = "Longitude of each location";
```

```
      :standard_name = "longitude";
      :_FillValue = 99999.0f; // float
      :units = "degrees_east";
      :valid_min = -180.0; // double
      :valid_max = 180.0; // double
      :QC_indicator = 1; // int
      :QC_procedure = 1; // int
      :uncertainty = " ";
      :comment = " ";
      :axis = "X";

    byte POSITION_QC(POSITION=7);
      :long_name = "quality flag";
      :conventions = "OceanSites reference table 2";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 9B; // byte
      :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
      :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";

    char DC_REFERENCE(TIME=7, STRING32=32);
      :long_name = "Station unique identifier in data centre";
      :conventions = "Data centre convention";
      :_FillValue = " ";

    char DATA_MODE(TIME=7);
      :long_name = "Delayed mode or real time data";
      :conventions = "R: real time; D:delayed mode; A:real time with adjustment";
      :_FillValue = " ";

    char DIRECTION(TIME=7);
      :long_name = "Direction of the profiles";
      :conventions = "A: Ascending profile, D: descending profile";
      :_FillValue = " ";

    byte DATA_MODE_CORA(TIME=7);
      :long_name = "Cora analysis";
      :conventions = "0: not analyzed in CORA; 1: analyzed in CORA";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 1B; // byte
      :flag_values = "0b, 1b";
      :flag_meanings = "not_cora_station cora_station";

    float DEPH(TIME=7, DEPTH=16);
      :long_name = "Depth";
      :standard_name = "depth";
      :units = "meter";
      :_FillValue = -99999.0f; // float
      :valid_min = 0.0f; // float
      :valid_max = 12000.0f; // float
      :axis = "Z";
      :positive = "down";

    byte DEPH_QC(TIME=7, DEPTH=16);
```

```
      :long_name = "quality flag";
      :conventions = "OceanSites reference table 2";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 9B; // byte
      :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
      :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";

    float PSAL(TIME=7, DEPTH=16);
      :long_name = "Practical salinity";
      :standard_name = "sea_water_salinity";
      :units = "psu";
      :_FillValue = 9.96921E36f; // float

    byte PSAL_QC(TIME=7, DEPTH=16);
      :long_name = "quality flag";
      :conventions = "OceanSites reference table 2";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 9B; // byte
      :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
      :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";

    float TEMP(TIME=7, DEPTH=16);
      :long_name = "Sea temperature";
      :standard_name = "sea_water_temperature";
      :units = "degree_Celsius";
      :_FillValue = 9.96921E36f; // float

    byte TEMP_QC(TIME=7, DEPTH=16);
      :long_name = "quality flag";
      :conventions = "OceanSites reference table 2";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 9B; // byte
      :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
      :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";

  // global attributes:
  :data_type = "OceanSITES vertical profile";
  :format_version = "1.2";
  :platform_code = "9900855";
  :date_update = "2016-04-14T10:07:07Z";
  :institution = "UNKNOWN INSTITUTION";
  :institution_edmo_code = "";
  :site_code = " ";
  :wmo_platform_code = "9900855";
  :coriolis_platform_code = "9900855";
  :platform_name = "Unknown Sea Mammal";
  :wmo_inst_type = "995";
  :source = "SEA MAMMAL, ANIMAL : observation";
```

```
  :history = "2016-04-14T10:07:07Z : Creation";
  :data_mode = "R";
  :quality_control_indicator = "6";
  :quality_index = "A";
  :references = "http://marine.copernicus.eu,http://www.coriolis.eu.org";
  :comment = " ";
  :conventions = "OceanSITES Manual 1.2, InSituTac-Specification-Document";
  :netcdf_version = "3.5";
  :title = "Global Ocean - In Situ Observation Copernicus";
  :summary = " ";
  :naming_authority = "OceanSITES";
  :id = "GL_LATEST_PR_TE_9900855_20160405";
  :cdm_data_type = "vertical profile";
  :family_label = "etc";
  :family_code = "TE";
  :area = "Global Ocean";
  :geospatial_lat_min = "-64.27000";
  :geospatial_lat_max = "-64.19000";
  :geospatial_lon_min = "81.09000";
  :geospatial_lon_max = "81.26000";
  :geospatial_vertical_min = "4.00";
  :geospatial_vertical_max = "834.00";
  :time_coverage_start = "2016-04-05T05:10:00Z";
  :time_coverage_end = "2016-04-05T22:10:00Z";
  :institution_references = " ";
  :contact = "codac@ifremer.fr";
  :author = "Coriolis and Copernicus data provider";
  :data_assembly_center = "Coriolis";
  :pi_name = " ";
  :distribution_statement = "These data follow Copernicus standards; they are public
and free of charge. User assumes all risk for use of data. User must display citation
in any publication or product using data. User must contact PI prior to any commercial
use of data.";
  :citation = "These data were collected and made freely available by the Copernicus
project and the programs that contribute to it";
  :update_interval = "daily";
  :qc_manual = "OceanSITES User\'s Manual v1.2";
}
```

Header of a Coriolis dataset (dimensions, variables and global attributes)

```
Dataset type: NetCDF-3/CDM

netcdf {
  dimensions:
    TIME = UNLIMITED;   // (23 currently
    DEPTH = 1;
    LATITUDE = 23;
    LONGITUDE = 23;
    POSITION = 23;
    STRING32 = 32;
    STRING256 = 256;
  variables:
    double TIME(TIME=23);
      :long_name = "time";
      :standard_name = "time";
      :_FillValue = 999999.0; // double
      :units = "days since 1950-01-01T00:00:00Z";
      :valid_min = 0.0; // double
      :valid_max = 90000.0; // double
      :QC_indicator = 1; // int
      :QC_procedure = 1; // int
      :uncertainty = " ";
      :comment = " ";
      :axis = "T";

    byte TIME_QC(TIME=23);
      :long_name = "quality flag";
      :conventions = "OceanSites reference table 2";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 9B; // byte
      :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
      :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";

    float LATITUDE(LATITUDE=23);
      :long_name = "Latitude of each location";
      :standard_name = "latitude";
      :_FillValue = 99999.0f; // float
      :units = "degrees_north";
      :valid_min = -90.0; // double
      :valid_max = 90.0; // double
      :QC_indicator = 1; // int
      :QC_procedure = 1; // int
      :uncertainty = " ";
      :comment = " ";
      :axis = "Y";

    float LONGITUDE(LONGITUDE=23);
      :long_name = "Longitude of each location";
      :standard_name = "longitude";
      :_FillValue = 99999.0f; // float
      :units = "degrees_east";
```

```
      :valid_min = -180.0; // double
      :valid_max = 180.0; // double
      :QC_indicator = 1; // int
      :QC_procedure = 1; // int
      :uncertainty = " ";
      :comment = " ";
      :axis = "X";

    byte POSITION_QC(POSITION=23);
      :long_name = "quality flag";
      :conventions = "OceanSites reference table 2";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 9B; // byte
      :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
      :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";

    char POSITIONING_SYSTEM(POSITION=23);
      :long_name = "Positioning system";
      :_FillValue = " ";
      :flag_values = "A, G, L, N, U";
      :flag_meanings = "Argos, GPS, Loran, Nominal, Unknown";

    char DC_REFERENCE(TIME=23, STRING32=32);
      :long_name = "Location unique identifier in data centre";
      :conventions = "Data centre convention";
      :_FillValue = " ";

    float DEPH(TIME=23, DEPTH=1);
      :long_name = "Depth";
      :standard_name = "depth";
      :units = "meter";
      :_FillValue = -99999.0f; // float
      :valid_min = 0.0f; // float
      :valid_max = 12000.0f; // float
      :axis = "Z";
      :positive = "down";

    byte DEPH_QC(TIME=23, DEPTH=1);
      :long_name = "quality flag";
      :conventions = "OceanSites reference table 2";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 9B; // byte
      :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
      :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";

    char DEPH_DM(TIME=23, DEPTH=1);
      :long_name = "method of data processing";
      :conventions = "OceanSITES reference table 5";
      :flag_values = "R, P, D, M";
      :flag_meanings = "realtime post-recovery delayed-mode mixed";
      :_FillValue = " ";
```

```
    float ATMS(TIME=23, DEPTH=1);
      :long_name = "Atmospheric pressure at sea level";
      :standard_name = "air_pressure_at_sea_level";
      :units = "hectopascal";
      :_FillValue = 9.96921E36f; // float

    byte ATMS_QC(TIME=23, DEPTH=1);
      :long_name = "quality flag";
      :conventions = "OceanSites reference table 2";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 9B; // byte
      :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
      :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";

    char ATMS_DM(TIME=23, DEPTH=1);
      :long_name = "method of data processing";
      :conventions = "OceanSITES reference table 5";
      :flag_values = "R, P, D, M";
      :flag_meanings = "realtime post-recovery delayed-mode mixed";
      :_FillValue = " ";

    float TEMP(TIME=23, DEPTH=1);
      :long_name = "Sea temperature";
      :standard_name = "sea_water_temperature";
      :units = "degree_Celsius";
      :_FillValue = 9.96921E36f; // float

    byte TEMP_QC(TIME=23, DEPTH=1);
      :long_name = "quality flag";
      :conventions = "OceanSites reference table 2";
      :_FillValue = -128B; // byte
      :valid_min = 0B; // byte
      :valid_max = 9B; // byte
      :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
      :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";

    char TEMP_DM(TIME=23, DEPTH=1);
      :long_name = "method of data processing";
      :conventions = "OceanSITES reference table 5";
      :flag_values = "R, P, D, M";
      :flag_meanings = "realtime post-recovery delayed-mode mixed";
      :_FillValue = " ";

  // global attributes:
  :data_type = "OceanSITES time-series data";
  :format_version = "1.2";
  :platform_code = "74541";
  :date_update = "2016-04-06T02:53:32Z";
  :institution = "Unknown institution";
  :institution_edmo_code = "";
  :site_code = " ";
```

```
  :wmo_platform_code = "74541";
  :coriolis_platform_code = "74541";
  :platform_name = " ";
  :source = "BUOY/MOORING: SURFACE, DRIFTING : observation";
  :history = "2016-04-06T02:53:32Z : Creation";
  :data_mode = "R";
  :quality_control_indicator = "6";
  :quality_index = "A";
  :references = "http://marine.copernicus.eu,http://www.coriolis.eu.org";
  :comment = " ";
  :conventions = "OceanSITES Manual 1.2, InSituTac-Specification-Document";
  :netcdf_version = "3.5";
  :title = "Global Ocean - In Situ Observation Copernicus";
  :summary = " ";
  :naming_authority = "OceanSITES";
  :id = "GL_LATEST_TS_DB_74541_20160405";
  :cdm_data_type = "Time-series";
  :family_label = "drifter";
  :family_code = "DB";
  :area = "Global Ocean";
  :geospatial_lat_min = "-58.58600";
  :geospatial_lat_max = "-58.41000";
  :geospatial_lon_min = "83.39400";
  :geospatial_lon_max = "83.48200";
  :geospatial_vertical_min = " ";
  :geospatial_vertical_max = " ";
  :time_coverage_start = "2016-04-05T00:00:00Z";
  :time_coverage_end = "2016-04-05T23:00:00Z";
  :institution_references = " ";
  :contact = "codac@ifremer.fr";
  :author = "Coriolis and Copernicus data provider";
  :data_assembly_center = "Coriolis";
  :pi_name = " ";
  :distribution_statement = "These data follow Copernicus standards; they are public
and free of charge. User assumes all risk for use of data. User must display citation
in any publication or product using data. User must contact PI prior to any commercial
use of data.";
  :citation = "These data were collected and made freely available by the Copernicus
project and the programs that contribute to it";
  :update_interval = "daily";
  :qc_manual = "OceanSITES User\'s Manual v1.2";
}
```

Sample of variables attributes:

Variable "TEMP"

```
float TEMP(TIME=23, DEPTH=1);
  :long_name = "Sea temperature";
  :standard_name = "sea_water_temperature";
  :units = "degree_Celsius";
  :_FillValue = 9.96921E36f; // float
```

Variable "TEMP_QC"

```
byte TEMP_QC(TIME=23, DEPTH=1);
  :long_name = "quality flag";
  :conventions = "OceanSites reference table 2";
  :_FillValue = -128B; // byte
  :valid_min = 0B; // byte
  :valid_max = 9B; // byte
  :flag_values = 0B, 1B, 2B, 3B, 4B, 5B, 6B, 7B, 8B, 9B; // byte
  :flag_meanings = "no_qc_performed good_data probably_good_data
bad_data_that_are_potentially_correctable bad_data value_changed not_used nominal_value
interpolated_value missing_value";
```