
SEMANTIC TECHNOLOGIES SUPPORTING THE EA-COP

IRD ECOSCOPE: STATUS AND PLANS

1 STATUS OF THIS DOCUMENT

Ver.	Date	Description of editing	Author
0.1	28/09/12	Draft 1: Current status, Planned work	Julien Barde and Pascal Cauquil

Status of this document 2

Current status.....**Error! Bookmark not defined.**

 ECOSCOPE System Features..... 3

 ECOSCOPE storage of RDF 4

 ECOSCOPE content..... 4

 ECOSCOPE underlying RDF schemas 5

ECOSCOPE System requirements 6

 Maintenance 6

Planned work..... 7

 Core products..... 7

 Collaboration prospects 7

2 CURRENT STATUS

The Ecoscope knowledge base is used and made available in different ways.

ECOSCOPE SYSTEM FEATURES

SPARQL ENDPOINT: same as FLOD SPARQL endpoint and available at:

<http://ecoscopebc.mpl.ird.fr/joseki/ecoscope.html>

This endpoint enables external applications to use the content of Ecoscope underlying knowledge base (RDF /SPARQL).

Ecoscope Web portal: is an application developed by IRD on top of the knowledge base to visualize its content through different html pages (or “views”) related to the objects of the domain:

- species,
- fishing vessels,
- agents,
- information resources (images, databases, spatial information, publications, plots...).

Among other functionalities the Web portal embeds :

- *a graphical browser / fact sheets generator* (developed by IRD, similar to FLOD data explorer) to browse the content of the knowledge base with dedicated GUIs (fact sheets about species, vessels...),
- *a search engine* (similar to FLOD QUERY SYSTEM) **WORKING WITH JENA METHODS AND / OR SPARQL QUERIES AND MORE RECENTLY WITH OPENSEARCH**. This search engine manages the following options for domain entities (species, vessel..) and related information resources discovery:
 - auto-completion or free text search,
 - multilingual search, synonyms, and other kinds of queries expansions,
 - clustering of results,
 - spatial search (ongoing).

ECOSCOPE WEB PORTAL IS DEVELOPED IN JAVA, USES JENA API AND SPARQL QUERIES AS WELL AS STRUTS2 FRAMEWORK TO SET UP HTML PAGES. ECOSCOPE WEB PORTAL COULD WORK WITH AN EXTERNAL SPARQL ENDPOINT.

Ecoscope conversion code

Ecoscope can be used by external applications as an agent to convert codifications systems. For now, Ecoscope is used by the application “Fishframe export” to convert IRD codes (for species, vessels..) into Fishframe compliant codes. This could be done by using another knowledge base / SPARQL endpoint (like FLOD).

Ecoscope opensearch

Recently, opensearch has been made available by IRD on top of Ecoscope to enable external applications to discover data in the knowledge base without having to deal with the GUIs. This will enable the iMarine infrastructure to use the content of Ecoscope KB (FORTH with xSearch, Geospatial Cluster for data discovery...). Opensearch results are delivered in html or RDF (genesi-dec RDF schema).

ECOSCOPE STORAGE OF RDF

TRIPLE STORE CAN BE MANAGED IN TWO DIFFERENT WAYS:

- on-disk store persistence: Apache Jena TDB1, like FLOD.
- Postgres RDBMS persistent storage

ECOSCOPE CONTENT

The prior goal of Ecoscope KB is to deliver a summary of the knowledge managed by IRD by using raw data / observations on some marine ecosystems: predation, by-catch....

Ecoscope should be seen as an example of a basis component to deliver knowledge to upper level / meta-portal like FLOD, GBIF..that are summarizing but not producing knowledge from raw data / observations.

RDF triples are generated according to the content of different data sources managed by IRD, UMR EME research unit:

- databases or semi-structured datasets (csv, xls...),
- spatial information,
- images,
- bibliographic references,
- agents (persons, projects, institutes...).

Ecoscope don't aim to replicate the content of these different images resources. The goal is to summarize the related knowledge.

According to data sources, Ecoscope has developed some codes to describe, replicate or summarize their content with dedicated RDF triples.

So far, observations are still stored in RDBMS as RDF is not used by researchers for data analysis.

Ecoscope KB has been filled through an ad hoc approach for demonstration purposes. Each data source corresponds to a set of RDF triples.

However in the long term we consider that the RDF data-flow between information resources and the KB should be automated as much as possible. This would be achieved through a wrapper.

2.1 ECOSCOPE UNDERLYING RDF SCHEMAS

When possible Ecoscope reused existing RDF schemas to describe following objects:

- information resources with Dublin Core and its extension Dublin Core terms,
- agents (persons, projects, organizations..) with FOAF,
- SKOS for semantic properties of objects (synonyms, multilingualism..)
- ...

However as no existing schema was covering the needs of Ecosystem Approach to Fisheries domain we developed our own schema. We aim to switch or map this schema for a common (top level) ontology.

3 ECOSCOPE SYSTEM REQUIREMENTS

3.1 MAINTENANCE

See Ecoscope content section. Improving the maintenance of Ecoscope is an important step for the coming months or years. We aim to use mediator / wrapper for data access (like SDMX) to generate the RDF triples that summarizes the content of the different data sources we are dealing with.

This is going to be achieved by recreating a new data flow based on MDST application from another research project (MACROES).

4 PLANNED WORK

4.1 CORE PRODUCTS

Fact sheet generator: when a partner is using the same RDF schemas to describe same or different objects then it becomes possible to run the Ecoscope fact sheet generator on top of the SPARQL endpoint of this partner. This could be achieved with foaf: agents or publications with DCMI. Of course the main interest would be to achieve this use case with species or vessels by sharing a top level ontology. A first demo has been done with FLOD SPARQL endpoint to get worms properties of marine species on the fly (taxon ranks).

XSEARCH:

as IRD has set up a SPARQL endpoint and opensearch on top of Ecoscope knowledge base, it will become possible for xSearch to query Ecoscope with other data sources and to cluster results with some URIs of Ecoscope ontology or TLO when available. A second step would be to run the same use case with opensearch RDF results (genesi-dec RDF schema) without having to deal with entity mining. A last step could consist in delivering xSearch services in a generic way (list of opensearch applications, list of SPARQL endpoints...)...

Bibliographic references tagging: either by enriching existing RDF bibliographic references or by creating new references from entity mining in pdf or doc files.

4.2 COLLABORATION PROSPECTS

iMarine: once we adopt a top level ontology, iMarine connected data sources could be used to generate compliant RDF triples and make them available through an iMarine SPARQL endpoint that becomes usable by different applications (FLOD, Ecoscope and others.): for example a fact sheet generator. ICIS VRE (with SDMX format), code lists manager are interesting sources for RDF generation. This would be particularly interesting for some partners who didn't publish so far the content of their data sources through SPARQL endpoints and RDF triples: WORMS, Fishbase, OBIS...iMarine can offer these partners an opportunity to bridge this gap. For example the reference SPARQL for species should be Fishbase and Worms, not FAO that is focusing on exploited species mainly.

FAO / FLOD: IRD would like FLOD to become usable for codes alignments through its SPARQL endpoint without having to deal with FLOD underlying RDF schema. This topic is related to the TLO. Same expectation to reuse FLOD content for automated fact sheet generation (as already done by FAO by making worms properties available through FLOD SPARQL endpoint).

FAO / AGINFRA: the approach of Agrotagger could be compared to bibliographic references indexing work with FORTH entity mining or FAO FCPPS applications.

FORTH / TOP LEVEL ONTOLOGY FOR XSEARCH AND OTHER USE CASES: the Top Level Ontology (TLO) is a model to abstract on conceptualizations in the EAF domain (in RDF) provided by iMarine partners (e.g. FAO, IRD). The use case for the TLO is primarily to inject that additional knowledge between a central query system (e.g. xSearch) and the search points of targeted systems (e.g. FLOD, Ecoscope), so that one query can be redirected and meaningfully translated to have consistent result set.