SEMANTIC TECHNOLOGIES SUPPORTING THE EA-COP

FISHERIES LINKED OPEN DATA: STATUS AND PLANS

1 STATUS OF THIS DOCUMENT

Ver.	Date	Description of editing	Author
0.1	18/09/2012	Draft 1: Current status, system requirements, Planned work	Claudio Baldassarre
0.2	25/09/12	Added scenarios of maintenance, added iMarine section under collaboration	Claudio Baldassarre
0.3	26/09/12	Addressed comments from Anton	Claudio Baldassarre
0.4	28/09/12	Addressed comments from Marc	Claudio Baldassarre

2 CURRENT STATUS

2.1 FLOD SYSTEM FEATURES

SPARQL ENDPOINT: is a web application based on HP Joseki[1] exposing the FLOD data to be queried with SPARQL language. The endpoint support CSV, XML and JSON ouput format, and allows for XSL styling. The endpoint support SPARQL 1.1 language standard. The FLOD endpoint is publicly available at: http://www.fao.org/figis/flod/endpoint/flod and is at the base of all applications that uses FLOD data. The FLOD SPARQL endpoint is also registered as a resource in the infrastructure.

DATA EXPLORER: is a web application based on Pubby[2] that allows navigating the data in the triple store across the network of relationships connecting them. It connects directly to the SPARQL endpoint to retrieve the content to navigate.

QUERY SYSTEM: a query interface based on http get/post methods to retrieve data from the triple store. This interface respond with JSON output and enables client applications to:

- Retrieve coded entities matching user input string (e.g. species names matching 'tuna')
- Retrieve text documents matching user input key terms (e.g. PDF publications containing the indexed term 'tuna')
- Retrieve text snippets for a given document that match user input key terms (e.g. paragraphs mentioning 'tuna')
- Retrieve all text documents related to a given FLOD coded entity (e.g. all documents annotated with 'Thunnus albacares')
- Retrieve meta information on sources carrying specific data (e.g. provenance, publisher, rights-holder, and ownership)
- Retrieve codes equivalent to the input code
- Retrieve names and translations of a given coded entity (e.g. species)

FLOD PORTAL: is a web application[3] that offer a graphic UI to users. It uses of the query system, the SPARQL endpoint and the data explorer.

[1] http://www.joseki.org/
[2] http://www4.wiwiss.fu-berlin.de/pubby/

[3] ver.0.4 at the time of writing this document

2.2 FLOD SOFTWARE COMPONENTS

TRIPLE STORE: on-disk store persistence: Apache Jena TDB[4] allows to load the complete network of FLOD datasets with limited system resources. JenaTDB implements the *Quad-Store* technology, where the 4th information added to the triple is a reference to source metadata (e.g. with rights-holder, publisher and provenance).

FLOD KNOWLEDGE BASE: created by FAO/FI consists of multiple RDF datasets organized by Information System. Each IS (e.g. FIGIS, GAUL, WoRMS, Fishbase, etc) is a data contributor and have reciprocal connections among themselves to form interconnected datasets. From the maintenance policy point of view the separation of the dataset per IS systems allows to define regulation that apply to each one individually, and allow for independent evolution lifecycle.

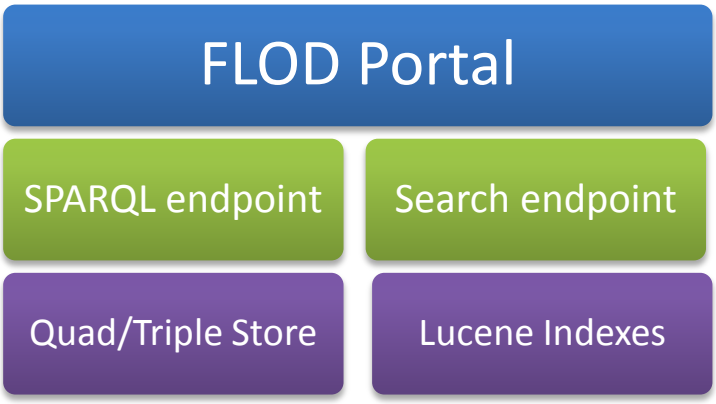

Figure 1 main components of the FLOD system. The portal offers GUI to users for interacting with the repository of FLOD thorugh the SPARQL endpoint, and the Lucene index of Fisheries publications through a keyword based search.

RDF-IER: a group of java components created by FAO/FI, based on Jena API[5], that convert data from different source formats into RDF datasets according to standard RDF vocabularies/ontologies (e.g. Dublin core, Darwin Core, FOAF etc). The RDF-ier also instantiates mappings from a selected source, among entities that already exist in FLOD, such as equivalencies, hierarchies, geographic relationships, biological relationships.

N-QUAD-FIER: java component created by FAO/FI, based on Jena API, that converts triples into "quads" by adding source reference IDs to RDF statements. The reference ID (i.e. a URI) can dereference other

[4] http://jena.apache.org/documentation/tdb/
[5] http://jena.apache.org/documentation/rdf/

metadata such as: provenance, publisher, rights-holder, and ownership of the data in the triple. Quads are the format to be stored in the Quad-store. N-Quadi-fier transforms and upload quads in to the Quad-store. The additional metadata for the sources will be generated with the RDF-ier.

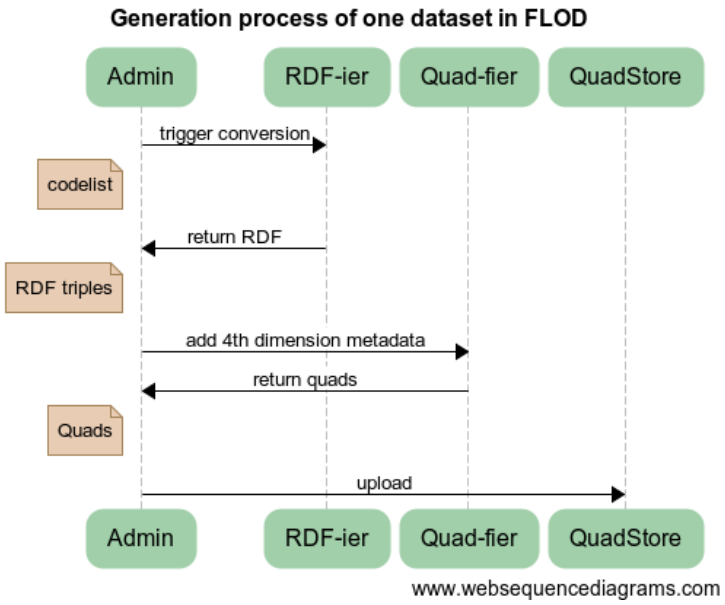

Figure 2 Sequence diagram[6] of the components involved in the creation and storage of a dataset in FLOD

ANNOTATOR: a java component created by FAO/FI that processes text to find named entities from the collections of entities in FLOD. The annotator produces RDF output associating the unique reference to the text document with the URIs of the annotating FLOD entities. The output will form part of FLOD Knowledge base as a separate but interconnected dataset.

INDEXING FRAMEWORK: created by FAO/FI based on Solr[7] framework for indexing and retrieving textual document. Solr offers great flexibility in indexing and querying the indexes for document retrieval. In FLOD it is used in combination with the dataset, as the bases for the search engine of the portal.

[6] http://tiny.cc/1jlclw
[7] http://lucene.apache.org/solr/

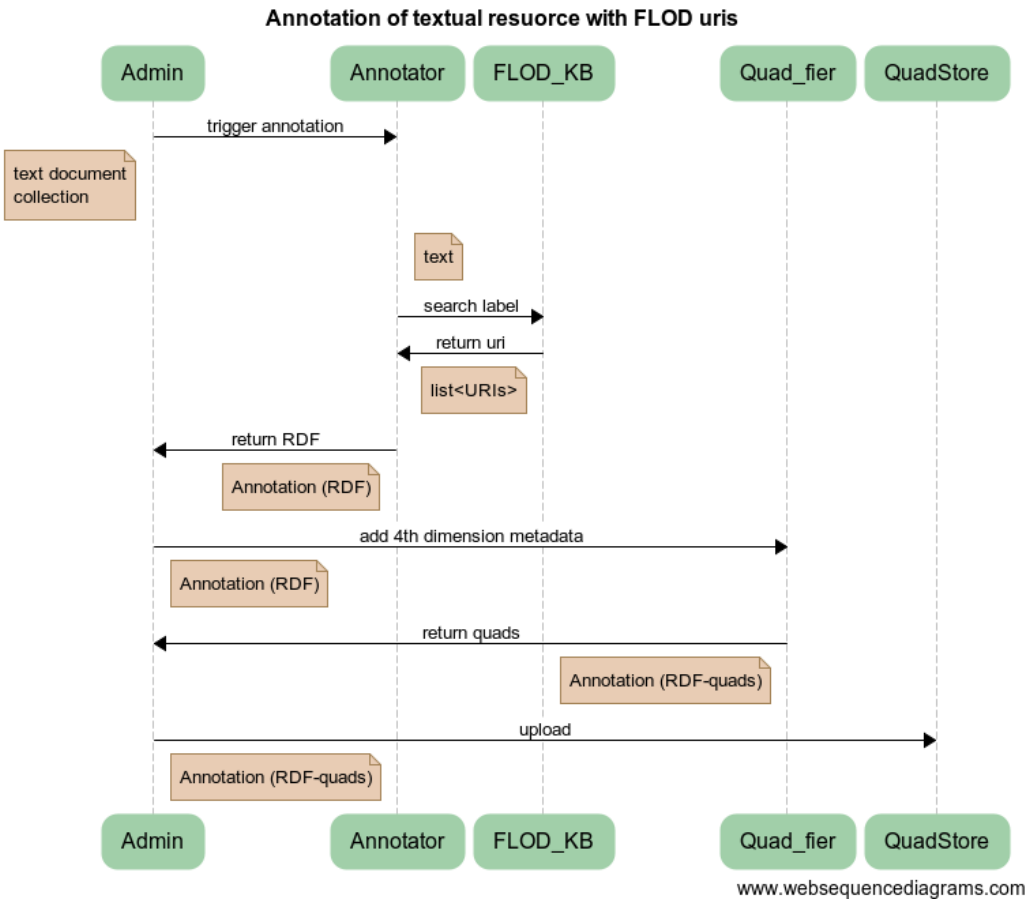

Figure 3 Sequence diagram[8] for annotating textual document with FLOD uris and store them

[8] http://tiny.cc/s4mclw

3 FLOD SYSTEM REQUIREMENTS

3.1 MAINTENANCE

FLOD dataset has incrementally added code lists, and the relationships among code lists extracted from existing systems in FI and other online sources (e.g. fishbase, aquamaps, worms, vliz, gaul). The procedure of adding data in FLOD dataset repetitively executes the following phases:

1. the identification of a source of data to include in the dataset,
2. the implementation of a specific RDF-ier java adapter to the source data format,
3. Generation of the RDF dataset according to the FLOD ontology model
4. Transformation of dataset in quads with addition of metadata about the source according given policies,
5. Upload to the Quad-store.

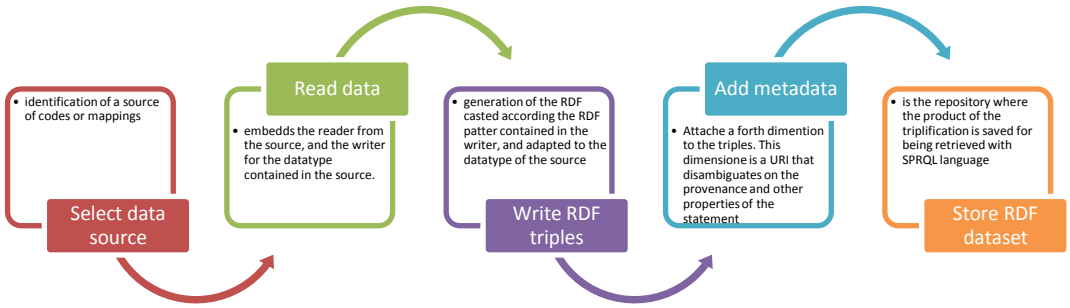

Figure 4 workflow of maintenance of a generic dataset in FLOD

The maintenance of FLOD consists in re-exectuting the workflow of these 5 phases for each source providing codes or mappings. In this sense the RDF-ier adapter is the component that adjusts to the evolution of the sources. Each RDF-ier has to facets: input format and the data types exposed by the source (e.g. vesseltype, geartype, species, watereareas, etc). With respect to the input format an RDF-ier will implement the appropriate reader. With respect to the data type the RDF-ier will implement the writer, which embeds the RDF pattern to cast the data in to FLOD compliant triples. In terms of maintenance one data type will have a reference RDF pattern which will be instantiated every time, for instance, when a new list of codes for vessels is required to be added. A new datatype requires to design a new RDF pattern and implemente the correct writer. Here below are two exemplary scenarios of maintenance.

3.1.1 SCENARIOS OF MAINTENANCE

3.1.1.1 NEW CODE LIST FOR AN ALREADY EXISTING FLOD DATATYPE

If a datatype (e.g. vesseltype) is already conceptualized in the FLOD ontology, this implies that an adapter with its writer already exists. When a new source of data (e.g. vessel list from Smartfish) has to be included in FLOD a reader for the RDF-ier adapter has to be coded to be bound to the SmartFish list. The adapter reads from the source and produces and RDF raw version of it (phase 2). This raw version is then processed to be casted by the RDF patterns producing FLOD ontology compliant triples (phase3).

3.1.1.2 NEW CODE LIST FOR A NON EXISTING FLOD DATATYPE

If FLOD ontology does not have the module for a datatype, the relevant concepts and properties have to be introduced in the ontology. A module is a self-contained conceptual artifact (a mini ontology) required to instantiate the data from the new datatype. The module or a portion of it provides the RDF pattern to cast the data into FLOD ontology compliant triples (phase3). The creation of the module proceeds according the practices followed to design the already existing modules[9].

[9] The modules composing the FLOD ontologies are available here http://www.fao.org/figis/flod/onto/flod.owl

4 PLANNED WORK

4.1 CORE PRODUCTS

OPEN SEARCH: The query capabilities will be implemented using Open Search standard. This choice follows a technological direction in iMarine, and from other partners (e.g. IRD, FORTH), and thus will ease the process of integration within the infrastructure and interoperation among loosely coupled components. The implementation of the Open Search query interface implies a design (on collaborative basis) also of output schema and formats to be compatible with client applications. The query capabilities of the query interface will initially include the features offered by the current query system, in a later iteration user/application requirements will shape the new interface design.

ICIS-FLOD: One source of data, and maintenance workflow, will be established from ICIS to FLOD. The middleware between ICIS system and FLOD has been identified to be OpenSDMX component that exposes ICIS content in SDMX. The data to be pulled out from ICIS will firstly be individual code lists curated through ICIS and possibly new to FLOD; mappings among codelists of fisheries entities that are already in FLOD can be initially maintained through FLOD's existing data workflows, but should also be pulled out from ICIS facilities.

XSEARCH: The use of xSearch at his actual status still does not present a concrete use case that can satisfy FAO business cases for search. Nonetheless FI has been internally developing search components that specifically address needs originated within the division.A collaboration should be sought for xSearch to include such components, and develop them further; then xSearch can become the system to interface the document collections that FI wants to expose through user search. In this respect new user requirements will be produced so that xSearch team understands the feasibility of a development plan.

4.2 COLLABORATION PROSPECTS

IMARINE: FLOD system components, can benefit from the infrastructure to carry out computational tasks demanding large resources, or workflows involving big scale data. FAO/FI developed the components that features the functionalities needed to satisfy the community needs, and these components are so far envisaged as placeholders for "higher quality" services provided by the infrastructure. FAO/FI expects to delegate the computation of single tasks, to be executed within the infrastructure.

Specific features offered by FLOD can already serve scenarios of document tagging.

- Tagging documents with FLOD entities creates a network of documents relevant to each other by co-referencing the same entities. The same tagging mechanism can be enabled on text documents, GIS maps, images, or statistical time-series, with an entity mining process specific to the resource.
- enrich context of search, either expanding user searches using the network of relationships available, or aggregating more content still capitalizing the connections among entities, hence among documents.
- fill dynamically the sections of a factsheet. Such a use case has already had a pilot study from IRD in Ecoscope web portal, and will be improved by serving data coming from multiple repository of the same kind of FLOD (e.g. WoRMS).
- Inline entity suggestions for scenarios like FCPPS will support the editors in selecting the correct entity while the are typing in specific report sections (e.g. title). Upon accepted suggestions and inclusion as part of the text, the FCPPS report will result also annotated with the FLOD entity uri and exploited for the best document search/aggregation scenario.

OPEN ARCHIVE: The FAO project for the management of the FAO publications repository has it's own platform shipped with search capacities. However, the features developed in FI (to meet division needs in finding publications) are not yet covered. Open Archive, being an initiative serving the whole organization, could embed the components that offer such capacities developed under FI/iMarine program.

AGINFRA: is a project developing semantic technologies of interest to iMarine. Specifically in the context of searching publications, AgInfra has invested considerable effort, through its partners, in analysing and indexing documents with advanced text processing technologies. If AgInfra software components can be used to process and generate an index of target document collection, and the result improves on the indexing component currently available, a closer collaboration can see the two projects building on one's another objectives.

FORTH AND IRD: the Top Level Ontology (TLO) is a model to abstract on conceptualizations in the EA 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.

ECOSCOPE: the web portal from IRD partially based on the content of a knowledge base in RDF. The KB content focuses on species observation, while does not cover, and will benefit from an integration with FLOD, data about species codes, codelist, nomenclature. In general IRD has shown interest in being part of the FLOD network implementing locally a sibling data infrastructure similar to FLOD. The data exchange between the two systems (i.e. FLOD and Ecoscope) is either through high level query interface (e.g. OpenSearch) or via native language to query RDF dataset (e.g. SPARQL) through public endpoints. With IRD FAO/FI has in plan to develop a dynamic factsheet component that pulls up data from the merging of the

two knowledge bases, and populate the relevant sections. IRD already has a product that leverages the Ecoscope KB to generate HTML pages content.