6 Ontology models

Based on the data described in Chapter 4 and 5, we produced the following ontologies:

1. Land areas (Section 6.1),
2. Fishing areas (Section 6.2),
3. Biological entities (Section 6.3),
4. Fisheries commodities (Section 6.4),
5. Vessel types and size (Section 6.5),
6. Gear types (Section 6.6).

The corresponding ontologies (both as model only and including the actual data) are all publicly available from the FAO website at the following URL: http://www.fao.org/aims/aos/fi. Exact addresses are given together with the ontology description in the course of this section.

Names of ontology elements (except instances) are based on English (sometimes shortened). They apply the naming conventions reported in Appendix I, which are based on FAO internal naming conventions. In some cases, our naming convention differs from the one proposed in deliverable D1.1.1 [D7.1.1] (i.e., for the naming of classes and instances, for which we use lower cases and underscores), but it is not incompatible with it. The rationale of the naming convention we adopted is based on smoothing the connection with the underlying database where names are in lower case and with underscores to separate words. This choice should also be seen in light of the envisaged way to access the data in the database: instead of extracting the data from the database and writing it to a file, it will reside in the database and be accessed at run-time.

We also adopt a somewhat non-standard naming of datatype properties, in that besides using the xml:lang attribute we also add the two-character ISO code of the language in the name of the properties. Once again, this is done to comply with a widely used FAO convention, and to facilitate the reading of the ontologies by human users (be they ontology engineers and editors, or software developers).

Names of instances are generated by combining meta and database ID for each instance in the database. This is done in order to ensure uniqueness of the instance names (across the entire set of ontologies) as commonly done when dealing with the FIGIS database. Meta codes and IDs are also rendered (individually) as datatype properties in order to stay aligned with current practices of interaction with the database.

All datatypes are derived from the database with no modifications, in order to preserve as much as possible the ability to interact with the database and with any other systems built to access it.

All ontologies are in OWL xml-rdf.

6.1 Land areas

Populated ontology: (300Kb)	http://www.fao.org/aims/aos/fi/land_v1.0.owl
Model only:	http://www.fao.org/aims/aos/fi/land_v1.0_model.owl

Classes (instances):

+ group
  - economic_group (25)
  - geographic_group (8)
+ territory (256)

Disjoint classes:

all

Class restrictions:

economic: *forall* hasMember territory

geographic: *forall* hasMember territory

Datatype properties:

1. hasMeta *Domain: group, territory. Datatype: string (functional)*
2. hasID *Domain: group, territory. Datatype: string (functional)*
3. hasNameShort *Domain: group, territory. Datatype: string*
   a. hasNameShortEN *xml:lang=en (functional)*
   b. hasNameShortES *xml:lang=es (functional)*
   c. hasNameShortFR *xml:lang=fr (functional)*
4. hasNameOfficial *Domain: group, territory. Datatype: string*
   a. hasNameOffEN *xml:lang=en (functional)*
   b. hasNameOffES *xml:lang=es (functional)*
   c. hasNameOffFR *xml:lang=fr (functional)*
5. hasCode *Domain: group, territory. Datatype: string*
   a. hasCodeISO3 *(functional)*
   b. hasCodeISO2 *(functional)*
   c. hasCodeUNDP *(functional)*
   d. hasCodeUN49 *(functional)*
6. hasCoordinate *Domain: territory. Datatype: decimal*
   a. hasMinLat *(functional)*
   b. hasMinLon *(functional)*
   c. hasMaxLat *(functional)*
   d. hasMaxLon *(functional)*
7. hasAreaSize *Domain: territory. Datatype: string (functional)*

8. isValidFrom *Domain: group, territory. Datatype: string (functional)*
9. isValidUntil *Domain: group, territory. Datatype: string (functional)*

Object properties:

1. isInGroup <-> hasMember *Domain: group, territory.*

6.2 Fishing areas

Populated ontology: (130Kb) http://www.fao.org/aims/aos/fi/fishing_areas_v1.0.owl

Model only: http://www.fao.org/aims/aos/fi/fishing_areas_v1.0_model.owl

Classes (instances):

+ fishing_area
    - area (28)
    - subarea (67)
    - division (29)
    - subdivision (10)

Disjoint classes:

all subclasses of fishing_area

Class restrictions:

area: *forall contains* subarea

subarea: *forall contains* division

division: *forall contains* subdivision

Datatype properties:

1. hasMeta	*Domain: fishing_area. Datatype: string (functional)*
2. hasID	*Domain: fishing_area. Datatype: string (functional)*
3. hasName	*Domain: fishing_area. Datatype: string*
a. hasNameEN	*xml:lang=en (functional)*
b. hasNameFR	*xml:lang=fr (functional)*
4. hasCoordinate	*Domain: fishing_area. Datatype: decimal*
a. hasMaxLat *(functional)*	
b. hasMinLat *(functional)*	
c. hasMaxLong *(functional)*	
d. hasMinLong *(functional)*	
5. hasAreaSize	*Domain: fishing_area. Datatype: int (functional)*
6. isInland	*Domain: fishing_area. Datatype: Boolean (functional)*

Object properties:

contains	*Domain: fishing_area. Range fishing_area*

6.3 Biological entities

Populated ontology: (13.5Mb)	http://www.fao.org/aims/aos/fi/species_v1.0.owl
Model only:	http://www.fao.org/aims/aos/fi/species_v1.0_model.owl

Classes (instances):

+ biological_entity
    - group (7)
    - order (112)
    - family (848)
    - species (10604)

Disjoint classes:

all subclasses of biological_entity

Class restrictions:

group: *forall includesOrder* order, *forall includesFamily* family, *forall includesSpecies* species

order: f*orall includesfamily* family, *forall includesSpecies* species

family: *forall includesSpecies* species

Datatype properties:

1. hasMeta	*Domain: biological_entity. Datatype: string (functional)*
2. hasID	*Domain: biological_entity. Datatype: string (functional)*
3. hasName	*Domain: biological_entity. Datatype: string*
a. hasNameEN	*xml:lang=en (functional)*
b. hasNameES	*xml:lang=es (functional)*
c. hasNameFR	*xml:lang=fr (functional)*
4. hasNameFull	*Domain: biological_entity. Datatype: string*
a. hasNameFullEN	*xml:lang=en (functional)*
b. hasNameFullES	*xml:lang=es (functional)*
c. hasNameFullFR	*xml:lang=fr (functional)*
5. hasNameLong	*Domain: biological_entity. Datatype: string*
a. hasNameLongEN	*xml:lang=en (functional)*
b. hasNameLongES	*xml:lang=es (functional)*
c. hasNameLongFR	*xml:lang=fr (functional)*
6. hasNameScientific	*Domain: biological_entity. Datatype: string (functional)*
7. hasCode	*Domain: biological_entity. Datatype: string*

   a. hasCodeTax *(functional)*
   b. hasCodeAlpha3 *(functional)*

Obect properties:

1.	includesOrder	*Domain: biological_entity, range: biological entity*
2.	includesFamily	*Domain: biological_entity, range: biological entity*
3.	includesSpecies	*Domain: biological_entity, range: biological entity*

6.4 Fisheries commodities

Populated ontology: (10.6Mb) http://www.fao.org/aims/aos/fi/commodities_v1.0.owl

Model only: http://www.fao.org/aims/aos/fi/commodities_v1.0_model.owl

Classes (instances):

+ fi_commodity
    - isscfc (1230)
    - hs (140)

Disjoint classes:

all subclasses of fi_commodity

Class restrictions:

hc: *forall hasCorrISSCFC* isscfc

Datatype properties:

1. hasMeta *Domain: fi_commodity. Datatype: string (functional)*
2. hasID *Domain: fi_commodity. Datatype: string (functional)*
3. hasName *Domain: fi_commodity. Datatype: string*
    a. hasNameEN *xml:lang=en (functional)*
    b. hasNameES *xml:lang=es (functional)*
    c. hasNameFR *xml:lang=fr (functional)*
4. hasNameLong *Domain: fi_commodity. Datatype: string*
    a. hasNameLongEN *xml:lang=en (functional)*
    b. hasNameLongES *xml:lang=es (functional)*
    c. hasNameLongFR *xml:lang=fr (functional)*
5. hasNameFull *Domain: fi_commodity. Datatype: string*
    a. hasNameFullEN *xml:lang=en (functional)*
    b. hasNameFullES *xml:lang=es (functional)*
    c. hasNameFullFR *xml:lang=fr (functional)*
6. hasCode *Domain: fi_commodity. Datatype: string*
    a. hasCodeISSCFC *(functional)*
    b. hasCodeHS *(functional)*
    c. hasCodeSITC *(functional)*

Object properties:

1. hasCorrISSCFC *Domain: fi_commodity,*
   Range: fi_commodity

6.5 Vessel types and size

Populated ontology: (100Kb)	http://www.fao.org/aims/aos/fi/vessels_v1.0.owl
Model only:	http://www.fao.org/aims/aos/fi/vessels_v1.0_model.owl

Classes (instances):

+ vessel type
    + by_length
        - grt (12)
        - gt (15)
    - by_type (93)

Disjoint classes:

all

Class restrictions:

none

Datatype properties:

2. hasMeta	*Domain: vessel_type. Datatype: string (functional)*
3. hasID	*Domain: vessel_ type. Datatype: string (functional)*
4. hasName	*Domain: vessel_ type. Datatype: string*
a. hasNameEN	*xml:lang=en (functional)*
b. hasNameES	*xml:lang=es (functional)*
c. hasNameFR	*xml:lang=fr (functional)*
5. hasCode	*Domain: vessel_ type. Datatype: string*
a. hasCodeFAO *(functional)*	
b. hasCodeISSCFV *(functional)*	
6. hasStdAbb	*Domain: vessel_ type. Datatype: string (functional)*
7. hasDescription	*Domain: vessel_ type. Datatype: string*
a. hasDescEN	*xml:lang=en (functional)*
b. hasDescES	*xml:lang=es (functional)*
c. hasDescFR;	*xml:lang=fr (functional)*
8. hasEntryDate	*Domain: vessel_ type. Datatype: datetime (functional)*
9. hasVessClassGRT	*Domain: vessel_ type. Datatype: string*
10. hasVessClassLength	*Domain: vessel_ type. Datatype: string*
11. hasVessClassPower	*Domain: vessel_ type. Datatype: string*
12. hasLowerLimit	*Domain: by_length. Datatype: string*
13. hasUpperLimit	*Domain: by_length. Datatype: string*

6.6 Gear types

Populated ontology: (80Kb) http://www.fao.org/aims/aos/fi/gears_v1.0.owl

Model only: http://www.fao.org/aims/aos/fi/gears_v1.0_model.owl

Classes (instances):

+ isscfg
  - level1
  - level2
  - level3

Disjoint classes:

all

Class restrictions:

none

Datatype properties:

1. hasID	*Domain: isscfg. Datatype: string (functional)*
2. hasMeta	*Domain: isscfg. Datatype: string (functional)*
3. hasName	*Domain: isscfg. Datatype: string*
a. hasNameEN	*xml:lang=en (functional)*
b. hasNameES	*xml:lang=es (functional)*
c. hasNameFR	*xml:lang=fr (functional)*
4. hasDescription	*Domain: isscfg. Datatype: string*
a. hasDescEN	*xml:lang=en (functional)*
b. hasDescES	*xml:lang=es (functional)*
c. hasDescFR	*xml:lang=fr (functional)*
5. hasEntryDate	*Domain: isscfg. Datatype: dateTime (functional)*
6. hasStdAbb	*Domain: isscfg. Datatype: string (functional)*
7. hasCodeISSCFG	*Domain: isscfg. Datatype: string (functional)*

Object property:

hasSublevel	*Domain:isscfg. Range: isscfg*

7 Discussion

All ontologies based on FIGIS consist of a small number of classes, ranging from three (commodities) to five (land areas, water areas, biological entities, vessels), and a medium to large number of instances, ranging from seven (class *group*, in species_v1.0.owl) to 10604 (class *species* in the same ontology).

The modelling style is consistent. The same modelling style if for example adopted in the use of datatype properties for names and codes, in the use of (universal) constraints and in the definition of domain and range of both datatype and object properties.

7.1 Selection of properties

All pieces of information of most common use by the application for fisheries have been included in the ontologies, including the ID of all items in the database and the meta code used to identify the "type" of reference data at hand. Since the combination of ID and meta code is unique within the database we also use this combination to form the names of all instances.

All data included in the ontologies comes from the database, without further additions or modifications. As a consequence, sparsely populated columns in the database are rendered as empty properties. Since this emptiness is to be related to the dynamic nature of the database, we preferred not to impose constraints on required or non-required properties.

Some properties only make sense in conjunction with others, as in the case of geographical coordinates. In the FIGIS database coordinates are given in order to circumscribe areas, so 4 coordinates are usually stored for each area: minimum and maximum latitude and minimum and maximum longitude. In that case it is important to distinguish each coordinate from the other, but also to group all of them together. For this reason we preferred to use one property per coordinate, and to group them as subproperties of the same superproperty (cf. Section 6.1).

7.2 Managing multilinguality

Most of the reference data is available in more than one language, usually English, French and Spanish. Often, two or three names are available in each language, such as a "short name," a "long name" and an "official name." In all cases, names are established on the basis of international agreements that result in a 1-to-1 correspondence between languages.

We rendered each name (either short, long or official) by means of datatype properties, endowed with an rdf label *xml:lang* corresponding to the language at hand. The two-digit ISO codes of the language are also kept in the name of properties (as in "hasNameShortFR") in order to ease the visualization of properties by human editors and to stay close to existing practices adopted in FAO.

Given the sharp 1-1 correspondence between languages, the ontologies based on FIGIS constitute a simpler case of management of multilinguality than the model that is being proposed withing NeOn.[10]

[10] The NeOn model for multilinguality will described in deliverable D2.4.1, due at month 18, in parallel with the present deliverable.

7.3 Different flavours of hierarchies

The FIGIS database is organized according to a hierarchical structure (cf. Section 5.1). Also, most of the coding systems are taxonomic, as in the case of the “taxonomic code” for biological items, FAO divisions for water areas, HS and ISSCFC for commodities, ISSCFV for vessels, ISSCFG for gears. However, not all data with taxonomic codes is stored in the same way.

In the case of FAO water areas, there is a strict correspondence between the FAO taxonomic code and the hierarchy encoded in the group table. This means that the three pairs: major area, subarea; subarea, division; and division, subdivision are present in the group table as pairs of group and member (see Figure 5). Moreover, since this is a hierarchy based on physical inclusion, it is always complete (an area may not have subareas, but all subareas have an area to which they belong).

Biological items have taxonomic codes, but they do not strictly correspond to the hierarchy stored in the group table. In fact, taxonomic codes include a place for the genus, which is not actually used in the statistical database. Moreover, the chain from main group to species may not be complete, as there are species that have no family or order in the database, but only a main group. This is the reason why we used three object properties (includesOrder, includesFamily, includesSpecies) instead of only one (as in the case of the fishing water areas).

Also fisheries commodities have taxonomic codes associated (ISSCFC and HS) but the taxonomy expressed by these codes are not encoded in the group table. That table only represents the *correspondence* between each commodity item in the HS and items in the ISSCFC system. In other words, in that case the parent child structure is used not to express a hierarchy *within* a coding system but to express the relationship between two different systems. This is the reason why commodities are represented in the ontology as two flat lists of instances, despite their complex taxonomic codes.

Finally, the main classification of gear types is based on a filter, not on a meta code. This means that ISSCFG objects do not exist as such in the item table, but can only be reconstructed by looking at the ISSCFG filter code and at the corresponding entries in the parent table.

7.4 Mapping

The creation of mappings between ontologies is out of the scope of this deliverable. However, the ontology of commodities is an example of an ontology that could be managed as two ontologies linked by mapping. In fact, the two classifications for commodities used are maintained by different organizations: the ISSCFC is maintained by FAO, and the HS is maintained by the World Customs Organization (WCO). It could be more effective and safer to manage them in two different ontologies. In order for this strategy to be applied, a sound mechanism for editing, maintaining, and versioning mappings should be in place. Special attention should be paid to the handling of changes undergone by the ontologies to be linked.

Also, some of the ontologies produced can be easily put in “contact’ to one another. Consider for example the relation between land areas and water areas, or the relation between commodities and biological species. These relations would be best represented by means of mappings across ontologies instead of relations within the same ontologies.

Finally, much information highly related to the domains at hand could be taken from other sources (other databases, text documents, human knowledge) and would increase enormously the potential of the ontologies produced. These pieces of information could be added within single ontologies (for example borders of land areas, territories and groups), or they could “link” two ontologies (for example about shores, thus involving territories and water areas). Also linguistic information such as names of territories and commodities in languages not included in the reference tables could be added by some form of mapping.

8 Lessons learned

In this chapter we summarize the lessons learned during the generation of ontologies based on FIGIS.

8.1 Using non-integrated tools is error prone and time consuming

In order to populate the ontologies from the database, we followed the steps listed in Table 4. Numbers in the first column correspond to the sequential order for the action, which is listed in the second column, while the tools we used are listed in the third column.

Step	Action	Tool used
1	Inspect and query the DB	MySql or Oracle client, Navicat, Access
2	Edit and visualize the ontology model	Protégé, NeOn toolkit
3	Edit the R2O and query files	XMLSpy
4	Run the ODEMapster processor	Windows command prompt
5	Generate a single file with model and instances	Shell on Windows environment (cygwin)
6	Inspect the ontology (class definition and instances) as a whole	Protégé, NeOn toolkit, Semantic Networks

Table 4. Steps followed for the creation and population of the fisheries ontologies.

Often one or more steps in the sequence had to be iterated; and each step involved the use of a specific tool. In particular, steps 1-2, 2-3 and 3-4-5-6 are often repeated as groups.

We found that the use of different tools, totally independent and not integrated into a single environment, is time consuming and cumbersome to use. In particular, we run into the following issues:

1. Use of different language encoding. FIGIS uses UTF-8 by default, but in other applications this is not the case and/or it may be difficult to choose the desired encoding.
2. Selection of the right datatype to assign to ontology properties. The same datatypes used in the FIGIS database should be assigned to the ontology properties. In order to do this, it is necessary to inspect the target database in a seamless manner, and visualize at least the table descriptions and thier datatypes.
3. Use of different flavours of OWL dialects. It happens that different tools (e.g., Protégé and NeOn toolkit) support different fragments of OWL: this fact determines what operations can be performed with a tool and what can be saved.
4. Different default namespaces (some cumbersome editing was necessary in order to harmonize the namespaces manually given at the time of the creation of the ontology model, and the namespace automatically created by ODEMapster).
5. The window-based tools, including Protégé and the NeOn toolkit, had problems with opening and manipulating large ontologies (files) of that size. For this reasons, we largely used shell commands to inspect and edit the instances generated by ODEMapster and also the ontologies including both instances and model. This is then one more reason for

accessing the data directly from the database and putting the semantic layer on top of it, and to develop efficient mechanisms for managing ontology modules (cf., Chapter 7).

At least the two functionalities of ontology editing/visualization and the creation of the schema to query the DB and populate the ontologies should be integrated in a single environment.

8.2 Self-joins are critical to working with FIGIS

The process of accessing data from the database and converting it into an ontological model was found to be a complex task. Such complexity was not due to the number of classes to define (small), nor from their definition or the number of instances available for each class, but from the way hierarchies are stored in the database. In fact, as we described in Chapter 5, the reference tables are organized in the database as a “family” of hierarchies, where the structure of the main hierarchy is stored in the meta table and each individual domain hierarchy is stored in a separate group table. Then, in all those cases in which the domain hierarchy is deeper than two levels (i.e., the first level of parent-child relation), it is necessary to apply a self-join.

Since the version of ODEMapster that we used for our work did not support self-joins, we had to pre-process the tables and create new areas where the hierarchy is unrolled. This pre-processing is not applicable in a realistic scenario (it clashes with our requirements concerning maintenance and updating of the data) and it is both time consuming and error prone. Based on the feedback provided during this work, a new version of ODEMapster is under development, one which addresses this specific problem and in so doing it will be able to deal with structures such as the one used by the RMTS.

8.3 Graphical interfaces are critical, but they should also be flexible

The graphical interface (GUI) available for ODEMapster was rather basic, so we manually wrote the R2O schema and the query files. We found that XMLSpy [XMLSpy] was the tool most useful for the task (even though the files are not valid XML files). Manual editing of the files had the advantage of giving us the maximum degree of flexibility but it also made the process time consuming and very error-prone. Therefore, an appropriate GUI for the task should allow the user to compose schema and query files without actually having to know all the syntax details of the language. At the same time the experienced user should also be allowed to inspect and edit the code directly. In that case, appropriate documentation based on examples, syntax specification and tutorials should be available. A further improvement of the tool would consist in a mechanism to automatically generate the mapping to the database from a given ontology model. Finally, special attention should be paid to the debugging environment, which should be improved in order to clearly highlight the nature of the error and the place in the code where the error is located.

Finally, when legacy data is left in databases and accessed and queried by ontologies, it would extremely useful to be able to use the ontology as an interface for editing the data residing in the database. Once again, this implies that the visualization tools be flexible and adaptable enough to support this type of operation, and that permissions, versions and backup be correctly handled.

8.4 If efficiency is an issue, modularization is required

Some of the ontologies created are rather large (e.g., biological entities is 13.5 Mb), which causes problems when loading them into most software (see Chapter 8) and when operating with them. Some notions of “modularization” is required in order select and load only the portion of ontologies that need to be visualized and accessed. Given the fact that the ontologies based on reference tables are richer in instances than in classes, a convenient solution would be to select only the specific sets of properties, for example only one or more languages or only one or more classification codes out of the many available. Appropriate facilities for visualization, editing, and versioning these modules should be available.

9 Conclusions and next steps

In this deliverable we presented (Chapter 6) and discussed (Chapter 7) the first set of fisheries ontologies created within WP7, that are available at the following URL: http://www.fao.org/aims/aos/fi. All ontologies are based on fisheries reference tables. We populated the ontologies with data stored in a relational database. This allowed us to verify that it is possible to extract data from the database according to an ontological model, but not all the machinery needed is currently in place. In order to allow the NeOn partners to overcome this problem, we described in detail the structure of the database at hand and the problems we ran across during our work. We also listed the lessons learned during this exercise (Chapter 8), which will be useful to the advancement of the project.

We succeeded in creating the desired ontologies from the database but this was only possible by pre-processing the data. This solution is obviously not sustainable in a real setting, because of the work involved and because in practice it would make the ontologies non-updatable and therefore non usable.

Future work includes the creation of additional ontologies, including also ASFA and AGROVOC. This work will be reported on in Deliverable D7.2.3 (Enhanced Networked Fishery Ontology) due at M30. We will proceed by incrementally adding ontologies based on the inventory [D7.2.1], connecting them to one another to form a network, and experimenting with adding relations not present in the FIGIS database but available from other sources and with adding relations across the ontologies created so far. In order to do this, appropriate mapping mechanisms and tools for editing, visualizing, storing and versioning mappings should be in place. We listed possible examples of these mappings in Chapter 7.

Annex I. Naming conventions

URI based: www.fao.org/aims/aos/fi/

Ontology names: lower letters, words separated by underscore. The name should include the version number, in the form: "_vx.x", just before the entension.

Example: name_ontology_version.owl

Classes: lower letters, with underscore instead of spaces.

Example: http://www.fao.org/aims/aos/fi/

Properties and Relations: Camel style.

Example: hasMeta, hasCodeISO2, hasNameEN.

Instances: combination of meta code and id from the database. For example: "54002_ 105".

Annex II. Glossary of fisheries terms

Catch The total number (or weight) of fish caught by fishing operations. Catch should include all fish killed by the act of fishing, not just those landed. The catch is usually expressed in terms of wet weight. It refers sometimes to the total amount caught, and sometimes only to the amount landed. The catches which are not landed are called discards.

Commodity Goods and services which are the result of production processes normally intended for sale on the market at a price that is designed to cover their costs of production.

Fishery Generally, a fishery is an activity leading to harvesting of fish. It may involve capture of wild fish or raising of fish through aquaculture. A fishery can also be taken as a unit determined by an authority or other entity that is engaged in raising and/or harvesting fish. Typically, the unit is defined in terms of some or all of the following: people involved, species or type of fish, area of water or seabed, method of fishing, class of boats and purpose of the activities.

Fishery Fleet The term "fishery fleet" or "fishery vessels" refers to mobile floating objects of any kind and size, operating in freshwater, brackish water and marine waters which are used for catching, harvesting, searching, transporting, landing, preserving and/or processing fish, shellfish and other aquatic organisms, residues and plants.

Fishing Vessel The term "fishing vessel" is used instead when the vessel is engaged only in catching operations.

Non-Fishing Vessel The term "non-fishing vessel" applies to vessels performing other functions related to fisheries, such as supplying, protecting, rendering assistance or conducting research or training.

Gear A fishing gear is a tool used to catch fish, such as hook and line, trawl, gill net, trap, spear, etc.

Gross Register Tonnage (GRT) The Gross Register Tonnage represented the total measured cubic content of the permanently enclosed spaces of a vessel, with some allowances or deductions for exempt spaces such as living quarters (1 gross register ton = 100 cubic feet = 2.83 cubic metres).

Gross Tonnage (GT) The Gross Tonnage for ships of 24 metres in length and over refers to the volume of all ship's enclosed spaces (from keel to funnel) measured to the outside of the hull framing.

Inland Water The surface water existing inland, including lakes, ponds, streams, rivers, natural or artificial watercourses and reservoirs, and coastal lagoons and artificial water bodies.

Nominal Catch The sum of the catches that are landed (expressed as live weight equivalent). Nominal catches do not include unreported discards.

Production The total living matter (biomass) produced by a stock through growth and recruitment in a given unit of time (e.g. daily, annual production). The "net production" is the net amount of living matter added to the stock during the time period, after deduction of biomass losses through mortality. Also: The total elaboration of new body substance in a stock in a unit of time, irrespective of whether or not it survives to the end of that time. Also called: net production.

Annex III. List of acronyms

ASFIS Aquatic Sciences and Fisheries Information System

CWP Coordinating Working Party on Fishery Statistics

FIES FAO Fisheries and Aquatic Information and Statistical Service

GRT Gross Registered Tonnage

GT Gross Tonnage

HS Harmonized Commodity Description and Coding System

ISO International Organization for Standardization

ISSCAAP International Standard Statistical Classification of Aquatic Animals and Plants

ISSCFC International Standard Statistical Classification of Fishery Commodities

ISSCFG International Standard Statistical Classification of Fishing Gears

ISSCFV International Standard Statistical Classification of Fishing Vessels

RT Reference Tables

SITC Standard International Trade Classification of the UN

Annex IV. Reengineering the XML schema for FI Factsheets to OWL

Generalities

A large amount of information about fisheries, aquaculture and related subjects, including fishing techniques, fishing areas, fishery and aquaculture country profiles, is available in the form of fact sheets [FS]. All Fisheries fact sheets in FAO are in documents in XML format, structured according to a comprehensive XML schema [FSschema] that includes all elements used in all types of fact sheets.

Fact sheets are organized into domains (e.g., Aquaculture species, Fishing equipment, Fishery, Gear type), each corresponding to an element of the schema, under FIGISdoc, the root of any fact sheet (XML document). Domains are fully specified by means of nested elements. Each element includes a description meant for human use.

The nested elements that specify FIGIS documents (fact sheets) can represent fishery-related content, but also identifiers, values, types of things referred to, etc.

The schema makes use of existing standard element sets such as Dublin Core [DC], Extended Dublin Core [EDC], AGMES [AGMES] and AIDA [AIDA]. It also incorporates wherever possible existing classification schemes such as ISO standards for countries, currencies, languages, and other fisheries-related international classification schemes.

It is important to note that the schema was conceived as a means for editors to create structured documentation, and as such was not created based on a relational or ontological model, but was rather organised following hierarchical document formatting conventions. A dictionary of the elements used in the schema is available online [FSdic].

In this section, we describe the results of a bulk translation of the schema into an OWL model, what are the issues arising from this translation, and what can be done in order to take advantage of the knowledge contained in the fact sheets, without being overloaded with specifications bound to the documental form, rather than to the conceptual form of fishery information.

We firstly describe the generalities of the XSD schema, then the way the conversion to OWL has been performed, the issues for the usability and usefulness of the ontology, and finally the possible solutions for those issues.

Conversion of the FI XSD to OWL

We have translated the fact sheet schema to OWL, and made a preliminary analysis, trying to understand the semantic issues that derive from the translation.

The schema [FSschema] is available from the FI web site: http://www.fao.org/aims/aos/fi/fi.owl

The schema uses the following additional schemas that are common in the metadata community: AGMES [AGMES], AIDA [AIDA], Dublin Core [DC], Dublin Core Terms [DCT].

Using the TopBraid Composer [TBC] tool (XSD2OWL) for translating from XML Schema (XSD) to OWL, we have translated the FS schema into an OWL model. The TopBraid tool assumes a fairly common semantics for translating XSD; in particular, *elements*, e.g. *FisheryArea*, are converted both as owl:Classes, e.g. *Class:FisheryArea*, and as owl:ObjectProperties, e.g. *ObjectProperty:hasFisheryArea*. A trailing "has" is added in order to distinguish classes and properties resp. that are converted from the same element.

The reason for this translation is that XSD elements are associated to other elements by means of operators like *choice*, and of cardinalities. When an association is stated, a *type* is assumed to be filled. When such a type is an XSD datatype (e.g. string, float, etc.), the element can be converted to an owl:DatatypeProperty. When the type is not an XSD datatype, a trailing "type" is added to the name element in order to declare the new type to be applied in that case. E.g. in Figure 1 the FisheryArea element is associated to the FishingGround element with a "FishingGroundType". This structure is converted in OWL as an owl:ObjectProperty *hasFishingGround*, an owl:Class *FishingGround*, and an owl:Restriction *(hasFishingGround allValuesFrom FishingGround)* that subsumes the owl:Class *FisheryArea* (Figure 2).

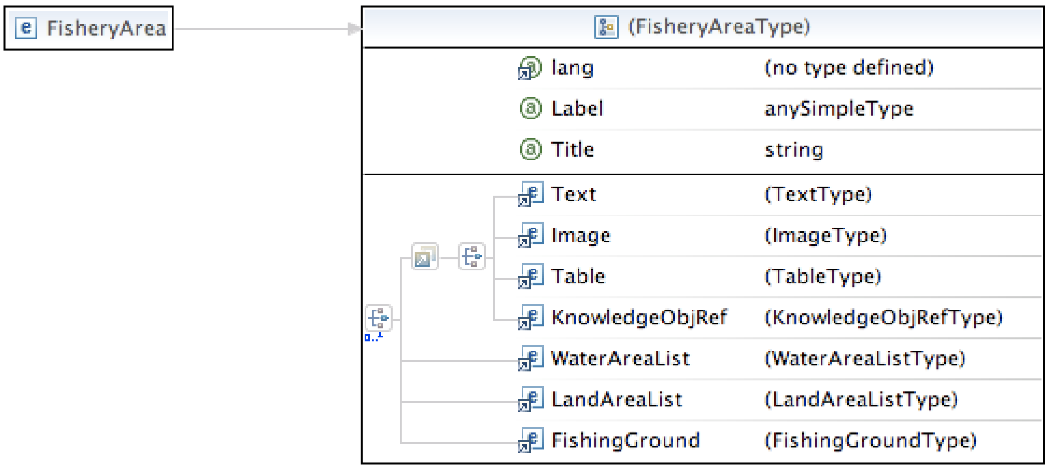


Figure 1. A visualization of the XSD element FisheryArea.

The ontology converted from the schema (fi.owl) is publicly available from the following address: http://www.fao.org/aims/aos/fi/fi.owl.

fi.xsd is made of XSD elements and XSD attributes, and a few XSD groups. No complex types and subtypes have been used. As typical with XSD, names conform to the following pattern:

<element name> <element name[Type]>, where the type has the same name, but with a trailing 'type' attached, e.g.:

FishTechnique --- FishTechniqueType

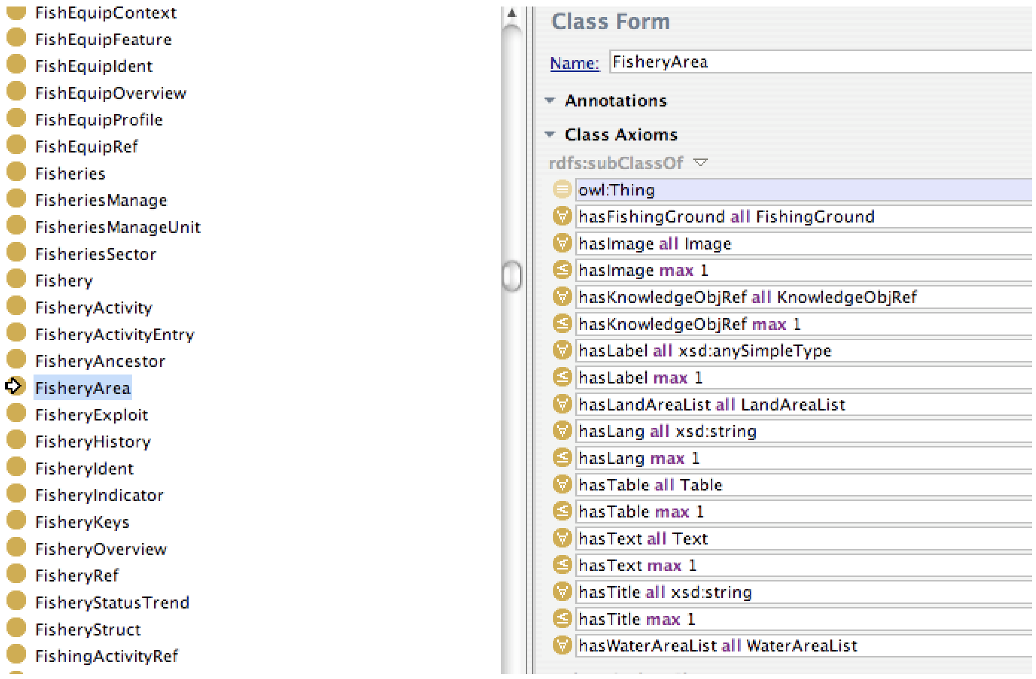
In fi.owl, this has been converted into the following pattern for OWL naming:

[has]<property name> <class name>, where class and property names are identical, the trailing 'type' has been removed, but the trailing 'has' prevents name clashes in OWL1.0, e.g.:

hasFishTechnique --- FishTechnique

The conversion tool also converts the schemas that are used in fi.xsd (e.g. Dublin Core). Although this conversion can also be interesting, after a discussion with the maintainers of the schema, we have decided to ignore their full conversion into OWL, because we do not need to maintain the conversions of the used schemas as well. Anyway, the elements used by the schema are still present in the converted OWL file with an appropriate namespace. For example, the owl:Class *KnowledgeObjRef* uses the owl:ObjectProperty *hasCreator* translated from the DublinCore Elements namespace (see Figure 4).

The converted fi.owl ontology consists of 653 named classes, 575 object properties, 163 datatype properties, and 15 hybrid properties, which have both object- and datatype-ranges (see issues below). Classes are characterized by means of 8424 occurrences of subsuming restrictions (with an average set of about 13 for each class). Among the occurrences of restrictions, 4695 are universals (allValuesFrom), which translate the integrity constraints from the XSD; 3332 are maxCardinality, 373 are (exact) cardinality, and 24 are minCardinality.



Class Form

Name: FisheryArea

Annotations

Class Axioms

rdfs:subClassOf

```
owl:Thing
hasFishingGround all FishingGround
hasImage all Image
hasImage max 1
hasKnowledgeObjRef all KnowledgeObjRef
hasKnowledgeObjRef max 1
hasLabel all xsd:anySimpleType
hasLabel max 1
hasLandAreaList all LandAreaList
hasLang all xsd:string
hasLang max 1
hasTable all Table
hasTable max 1
hasText all Text
hasText max 1
hasTitle all xsd:string
hasTitle max 1
hasWaterAreaList all WaterAreaList
```

Figure 2. A visualization of the owl:Class FisheryArea, translated from an XSD element.

Evaluating the design of the converted ontology

Converting XSD to OWL is very useful, but the results should be usually reengineered in order to reach an appropriate syntactic and semantic quality that makes the ontology usable and useful. We have carried out an analysis that has made them emerge four issues, which are dealt with solutions in Section 4.

The analysis of the OWL FI schema after the conversion has been annotated by means of owl:versionInfo annotations within the fi.owl file, augmented with a "TODO:" keyword at the beginning of the annotation.

(1) *Compatibility with existing tools and syntactic checks*

fi.owl is loadable on TopBraid Composer and SemanticWorks [SW], while Swoop and Protégé have some syntactic problems, whose cause has not been identitied yet. The NeOn toolkit loads it, but the nominals (oneOf) derived from XSD joins are not visualized, because the NeOn toolkit does not support nominals yet. When loaded, syntactic checks reveal the well-formedness of the ontology.

(2) *Semantic checks and OWL species*

A semantic check has been performed with the Pellet reasoner, and reveals no inconsistencies on the Class hierarchy (quite trivially, since no disjointness axioms or negations are asserted). The presence of 15 hybrid properties causes the ontology to be OWL-Full. At a closer glance, most of the 15 properties can be easily fixed to be either ObjectProperties of DatatypeProperties only. Without this flaw, the ontology is in OWL-DL, due to the many DataRange nominals (about 300), used in order to translate XSD "joins".

(3) *Design issue I: a hybrid domain of interpretation for the ontology?*

As Figure 3 shows, certain classes refer to textual material or full documents, rather than to objects of the fishery domain. For example, *KnowledgeObjRef* is one of such classes. In fact, when the OWL translation occurs, heterogeneous elements like *FisheryArea*, and *KnowledgeObjRef* become both owl:Classes. Due to the typical conventions in creating OWL ontologies, which suggest to give intuitive names to classes and properties, at a first glance the domain of interpretation of the ontology resulting from the FI schema contains fishery objects (areas, techniques, species, etc.), data, or documents. As said above, this intuition is misleading, because all elements in the FI schema are defined having in mind the management of documents, therefore even those classes that apparently describe fishery objects have to be understood as classes including documental objects that *express* descriptions of fishery objects, or *refer* to fishery objects. This situation is typical in web technology, because html documents can link, refer, or describe either other pages, or real world entities and concepts, by using the same `href` mechanism. In other words, in schemas designed for web applications, it is common to find a mixture of documental and real world knowledge.

Recent research has addressed this issue, known as *identity and reference over the web*, or simplier, as the *identity crisis*. An ontology like IRE (Identity, Reference, and Entities)[11], which is also aligned to the NeOn C-ODO ontology [D211] is able to describe the difference between different web references, between web pages and real world entities, between URIs of ontology elements and URIs of web pages, etc. Based on this analysis, the FI ontology is not homogeneous with the other Fishery ontologies, whose interpretation domain directly address fishery objects.

In the next section, we propose that an appropriate analysis of the documental references in the FI ontology can lead to a reengineering that makes it emerge a more conventional fishery ontology.

(4) *Design issue II: partial consistency in the structure of the documental ontology*

As previously said, the FIGIS XSD has not been defined in order to create a database, or as a typical ontology or conceptual model, but rather in order to maintain a document management system. For example, the notion of "Fish Technique" is spread out into several classes in order to

[11] http://www.loa-cnr.it/ontologies/IRE/IRE.owl

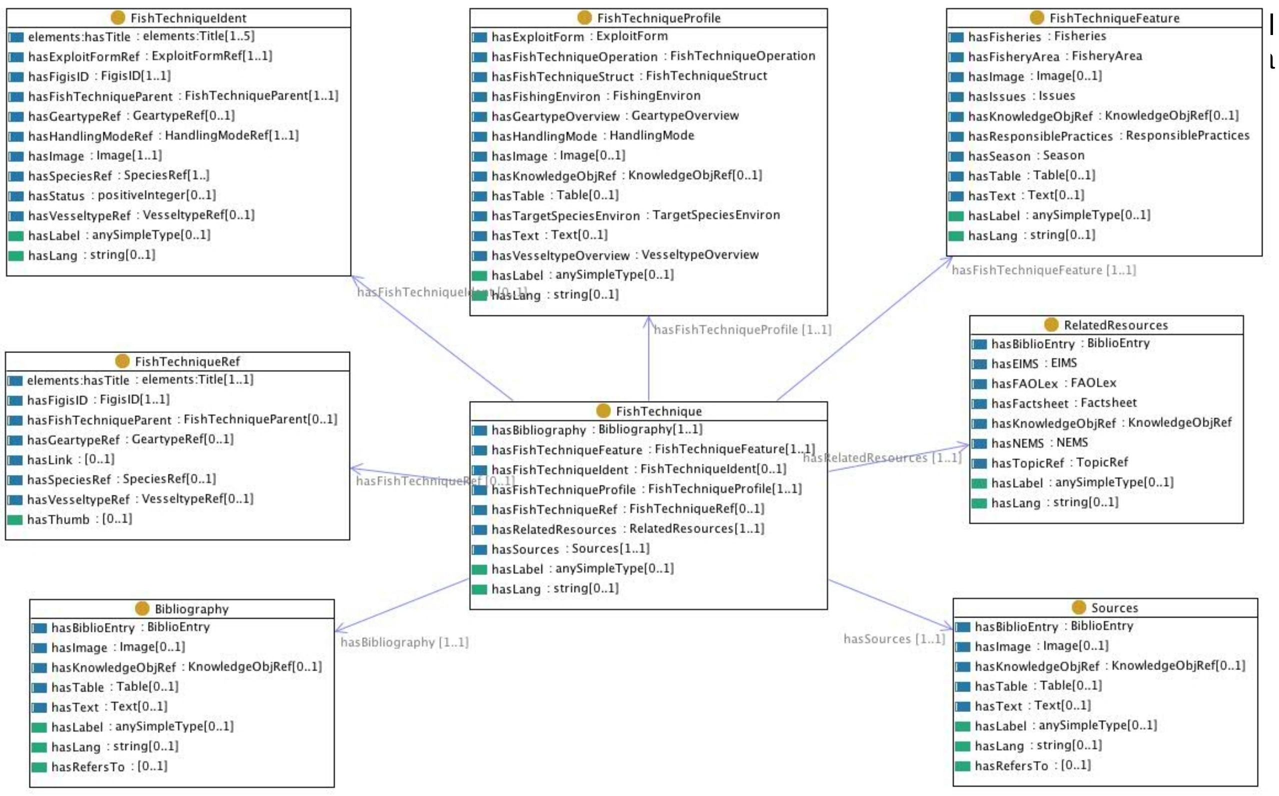

FishTechniqueIdent
elements:hasTitle : elements:Title[1..5]
hasExploitFormRef : ExploitFormRef[1..1]
hasFigisID : FigisID[1..1]
hasFishTechniqueParent : FishTechniqueParent[1..1]
hasGeartypeRef : GeartypeRef[0..1]
hasHandlingModeRef : HandlingModeRef[1..1]
hasImage : Image[1..1]
hasSpeciesRef : SpeciesRef[1..]
hasStatus : positiveInteger[0..1]
hasVesseltypeRef : VesseltypeRef[0..1]
hasLabel : anySimpleType[0..1]
hasLang : string[0..1]
FishTechniqueProfile
hasExploitForm : ExploitForm
hasFishTechniqueOperation : FishTechniqueOperation
hasFishTechniqueStruct : FishTechniqueStruct
hasFishingEnviron : FishingEnviron
hasGeartypeOverview : GeartypeOverview
hasHandlingMode : HandlingMode
hasImage : Image[0..1]
hasKnowledgeObjRef : KnowledgeObjRef[0..1]
hasTable : Table[0..1]
hasTargetSpeciesEnviron : TargetSpeciesEnviron
hasText : Text[0..1]
hasVesseltypeOverview : VesseltypeOverview
hasLabel : anySimpleType[0..1]
hasLang : string[0..1]
FishTechniqueFeature
hasFisheries : Fisheries
hasFisheryArea : FisheryArea
hasImage : Image[0..1]
hasIssues : Issues
hasKnowledgeObjRef : KnowledgeObjRef[0..1]
hasResponsiblePractices : ResponsiblePractices
hasSeason : Season
hasTable : Table[0..1]
hasText : Text[0..1]
hasLabel : anySimpleType[0..1]
hasLang : string[0..1]
hasFishTechniqueIdent [0..1]
hasFishTechniqueProfile [1..1]
hasFishTechniqueFeature [1..1]
RelatedResources
hasBiblioEntry : BiblioEntry
hasEIMS : EIMS
hasFAOLex : FAOLex
hasFactsheet : Factsheet
hasKnowledgeObjRef : KnowledgeObjRef
hasNEMS : NEMS
hasTopicRef : TopicRef
hasLabel : anySimpleType[0..1]
hasLang : string[0..1]
FishTechniqueRef
elements:hasTitle : elements:Title[1..1]
hasFigisID : FigisID[1..1]
hasFishTechniqueParent : FishTechniqueParent[0..1]
hasGeartypeRef : GeartypeRef[0..1]
hasLink : [0..1]
hasSpeciesRef : SpeciesRef[0..1]
hasVesseltypeRef : VesseltypeRef[0..1]
hasThumb : [0..1]
FishTechnique
hasBibliography : Bibliography[1..1]
hasFishTechniqueFeature : FishTechniqueFeature[1..1]
hasFishTechniqueIdent : FishTechniqueIdent[0..1]
hasFishTechniqueProfile : FishTechniqueProfile[1..1]
hasFishTechniqueRef : FishTechniqueRef[0..1]
hasRelatedResources : RelatedResources[1..1]
hasSources : Sources[1..1]
hasLabel : anySimpleType[0..1]
hasLang : string[0..1]
hasFishTechniqueRef [0..1]
hasRelatedResources [1..1]
Bibliography
hasBiblioEntry : BiblioEntry
hasImage : Image[0..1]
hasKnowledgeObjRef : KnowledgeObjRef[0..1]
hasTable : Table[0..1]
hasText : Text[0..1]
hasLabel : anySimpleType[0..1]
hasLang : string[0..1]
hasRefersTo : [0..1]
hasBibliography [1..1]
hasSources [1..1]
Sources
hasBiblioEntry : BiblioEntry
hasImage : Image[0..1]
hasKnowledgeObjRef : KnowledgeObjRef[0..1]
hasTable : Table[0..1]
hasText : Text[0..1]
hasLabel : anySimpleType[0..1]
hasLang : string[0..1]
hasRefersTo : [0..1]

Figure 3. A visualization of the owl:Class *FishTechnique* from fi.owl.

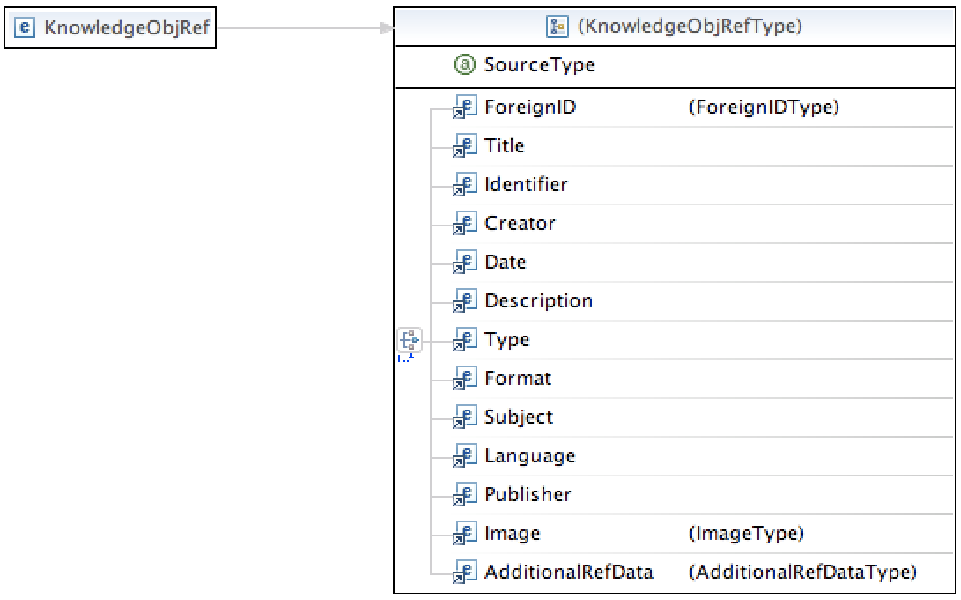
By looking at the attributes and properties of each class (see Figure 3), we can understand the rationale adopted case by case:

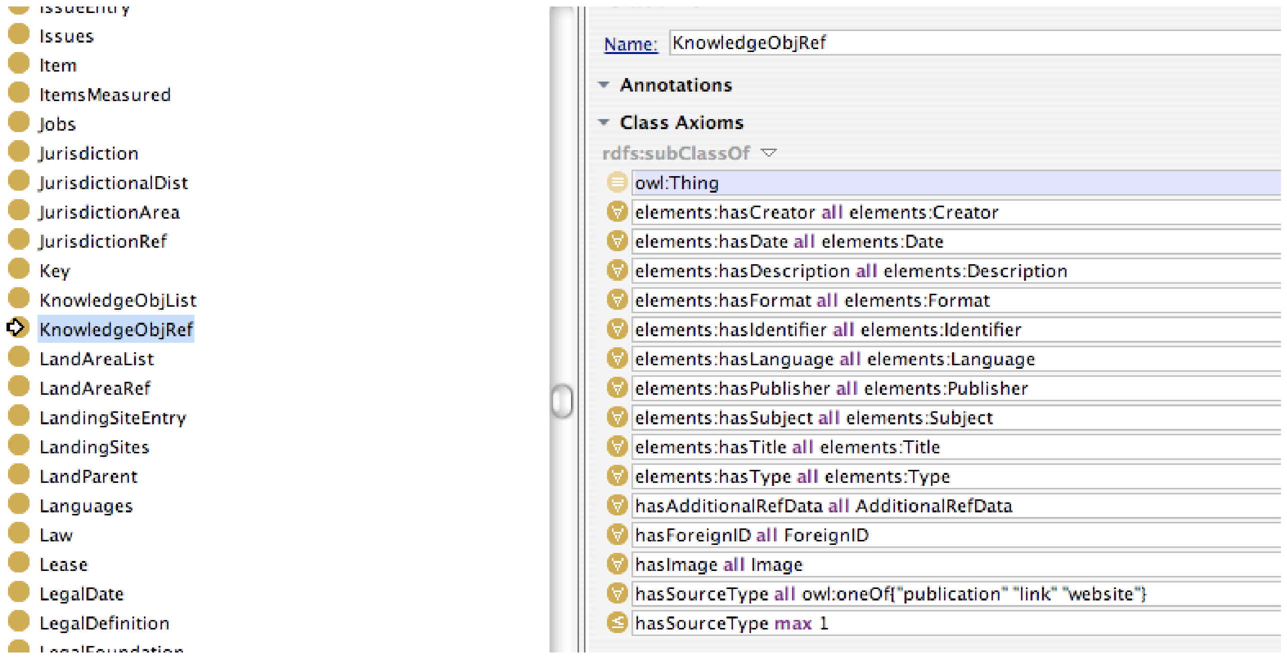
- the FishTechnique class is used to collect (directly or indirectly) all the structured information types that are specific to fish techniques: Ident, Ref, Feature, Parent, Profile, Operation, Struct, but also other, non-specific information, such as Season, FisheryArea, etc., and purely documental information, such Biblography, Sources, etc.
- the FishTechniqueIdent class is used to refer to the actual entities (fish techniques), e.g. by relating them to exploitation forms, geartypes, handling modes, species, vessel types, but also images and “parent” techniques (via the FishTechniqueParent class)
- the FishTechniqueRef is used to distill the basic information required to construct a FishTechnique document
- the FishTechniqueOverview class is used to declare the structure of the document, with images, tables, language type, etc.
- etc.

Unfortunately, there is not a stable pattern for encoding different kinds of information into classes with different but related names. For example, with the notion of “Aquatic Resource”, contrary to what is designed for fish techniques, the AqRes class encodes the properties referring mostly to actual entities, while AqResIdent encodes properties of the related information.

As a result, the OWL version of the schema has classes with properties mixing up data about the entities referred in fact sheets (e.g. *FishTechniqueIdent*), data about textual descriptions of those entities (e.g. *KnowledgeObjRef*), and data about the document that expresses textual descriptions (e.g. *Sources*).

Figure 4. A visualization of the XSD element *KnowledgeObjRef*





Issues
Item
ItemsMeasured
Jobs
Jurisdiction
JurisdictionalDist
JurisdictionArea
JurisdictionRef
Key
KnowledgeObjList
KnowledgeObjRef
LandAreaList
LandAreaRef
LandingSiteEntry
LandingSites
LandParent
Languages
Law
Lease
LegalDate
LegalDefinition

Name: KnowledgeObjRef

Annotations

Class Axioms

rdfs:subClassOf

owl:Thing
elements:hasCreator all elements:Creator
elements:hasDate all elements:Date
elements:hasDescription all elements:Description
elements:hasFormat all elements:Format
elements:hasIdentifier all elements:Identifier
elements:hasLanguage all elements:Language
elements:hasPublisher all elements:Publisher
elements:hasSubject all elements:Subject
elements:hasTitle all elements:Title
elements:hasType all elements:Type
hasAdditionalRefData all AdditionalRefData
hasForeignID all ForeignID
hasImage all Image
hasSourceType all owl:oneOf{"publication" "link" "website"}
hasSourceType max 1

Dealing with the issues arisen in the evaluation

(1) *Loadability*

Since there is no apparent reason behind the loading errors in Protégé and Swoop (the syntactic check goes plain on Neon Toolkit, TopBraid Composer, and SemanticWorks), we will simply send a note to the maintainers of those tools.

(2) *Hybrid properties*

As said above, the 15 properties causing the ontology to be OWL-Full will be fixed in the next phase of reengineering, and the results and lessons learnt will be reported in the next deliverable. The intervention requires to fix the properties that are used either as owl:ObjectProperty or as owl:DatatypeProperty. For example, *hasStatus* is used with the datatype range *xsd:positiveInteger*, with sets of values (strings), e.g.: *(owl:oneOf{"Received" "Processed" "Screened" "Filled" "Sent"}),* and also with the owl:Class *Status*. This can be due to the lack of a unique name assumption in the development of the schema, or to the tolerance of XSD, which does not check for the consistent use of element names across the schema.

(3) *Hybrid domain of interpretation*

While assuming that the actual domain of interpretation for fi.owl includes only documental objects solves the issue (3), which can be confined to a problem in the naming patterns used for XSD elements, still there is a feeling that the apparently documental knowledge encoded in fi.owl could be partly reused to produce a properly ontology for fishery. In the following we formulate a proposal for an activity that will possibly be reported in the next deliverable.

The proposal basically suggests to consider, in the vein of the “identity crisis”, that relations between documental objects often *mirror* relations between domain objects, in this case fishery objects. In practice, it is useful to assume that the *textual* associations between fishery-related document types that are declared in the FI ontology correspond to *factual* associations between the fishery objects described in a document, e.g. a fishery area associated with a fishing ground. Interestingly enough, it is possible to reengineer the FI ontology, so that it only contains axioms with a strict fishery interpretation, e.g. that the owl:Class *FisheryArea* (once interpreted as a class of geographical areas) has the owl:Restriction *(hasFishingGround allValuesFrom FishingGround)* (interpreted as a class of fishing-relevant places).

This reengineering process needs to remove from the FI ontology all the axioms that refer to strictly documental associations, for example, the owl:Restriction *(hasText allValuesFrom Text)* should be removed in the process.

Other document-oriented associations can be used in different ways, in order to reengineer other kinds of knowledge contained in fact sheets. For example, the *hasTable* and *hasImage* properties (see their application in Figures 4 and 5) can be used jointly with an image annotator and an information extractor from tables, in order to add new explicit data e.g. to a fishery area that is assumed to be described in a fact sheet.

A preliminary workflow to refactor the parts of fi.owl that can be interpreted as an actual fishery ontology is sketched here.

- Step 1. We need to distinguish properties referring to actual entities, vs. properties referring to descriptions of entities, vs. properties referring to information objects, vs properties referring to extrinsic, e.g. management-oriented information

- Step 2. We need to partition the class space into classes of actual entities (e.g. aquatic resources, species, fish techniques), possibly inferring also sublass axioms with the help of experts (or semi-automatically, if applicable), and classes of information objects (e.g. images, tables, lists, formats)

- Step 3. We need to redistribute the properties according to the class space partition, and to link the two types of classes appropriately. These reengineering patterns [cf. D251, forthcoming] are extracted from the InformationObjects[12] and the IRE ontologies.

- Step 4. We need to create a modular structure into the reengineered FI ontology, which possibly matches the FSDAS modular architecture, so that the alignment between parts of the FI ontology and the FSDAS ontologies is made easier.

(4) *Structural sparseness*

Issue (4) requires a finer-grained approach, which lets the ontology designer be aware of the actual patterns used for documental reasons, and to decide if we should take action to fix them or use them consistently across the ontology, or if we just need to extract those patterns that are useful to fix the issue (3). This finer-grained approach is left to future work, which will be reported in the next deliverable on month 30.

[12] http://www.loa-cnr.it/ontologies/DUL.owl

References

[AGMES] FAO. Agricultural Metadata Element Set. http://www.fao.org/aims/agmes_intro.jsp

[AIDA] http://www.fao.org/fi/figis/devcon/schema/3_6/aida.xsd

[D2.1.1] D2.1.1. Design rationale for collaborative development of networked ontologies. NeOn deliverable. February 2007.

[D2.5.1] D2.5.1. Library of formal models and design patterns for collaborative development of networked ontologies. NeOn project report. To appear.

[DC] Dublin Core Metadata Initiative. http://dublincore.org/

[DCT] Dublin Core Terms. http://dublincore.org/documents/dcmi-terms/

[EDC] Extended Dublin Core. http://dublincore.org/schemas/xmls/qdc/2003/04/02/dc.xsd

[FS] FAO. Fisheries fact sheets. http://www.fao.org/fi/website/FIRetrieveAction.do?dom=topic&fid=16062&lang=en

[FSdic] FIGIS XML. List of elements. http://www.fao.org/fi/figis/devcon/diXionary/figisdoc3.5.html

[FSschema] XML schema for Fisheries Fact Sheets. http://www.fao.org/fi/figis/devcon/schema/3_6/fi.xsd

[SW] SemanticWorks. http://www.altova.com/

[TBC] TopBraid Composer. http://www.topbraidcomposer.com/

Bibliography

[AGMES] FAO. Agricultural Metadata Element Set. http://www.fao.org/aims/agmes_intro.jsp

[AGROVOC] FAO. AGROVOC thesaurus. http://www.fao.org/aims/ag_intro.htm

[ASFA] FAO. ASFA thesaurus. http://www4.fao.org/asfa/asfa.htm

[BAR03] J. Barrasa and O. Corcho and A. Gomez-Perez. Fundfinder -- a case study of database-to-ontology mapping. In Proc. ISWC Semantic integration workshop. 2003.

[BAR06] J. Barrasa. Semantic upgrade and publication of legacy data, Ontologies for Software Engineering and Software Technology, 2006

[BAR07] J. Barrasa. Modelo para la definición automática de correspondencias semánticas entre ontologías y modelos relacionales. PhD thesis. Facultad de Informativa, Universidad Politecnica de Madrid. Madrid, Spain. March 2007. Isbn: 90-75176-81-3. http://eprints.eemcs.utwente.nl/7146/

[BIZ03] C. Bizer. D2R MAP - A Database to RDF Mapping Language. In Proc. of 12th International World Wide Web Conference. 2003.

[DC] Dublin Core Metadata Initiative. http://dublincore.org/

[D1.1.1] D1.1.1. Networked Ontology Model. NeOn project report. http://www.neon-project.org/web-content/index.php?option=com_weblinks&catid=17&Itemid=35

[D7.1.1] D7.1.1. Specification of user requirements on the case study. 2006. NeOn project report. http://www.neon-project.org/web-content/index.php?option=com_weblinks&catid=17&Itemid=35

[D7.2.1] D7.2.1. Inventory of fishery resources and information management systems. 2007. NeOn project report. http://www.neon-project.org/web-content/index.php?option=com_weblinks&catid=17&Itemid=35

[D7.4.1] D7.4.1. Software Architecture for managing the Fisheries Ontologies Lifecycle. NeOn project report. To appear.

[EDC] Extended Dublin Core. http://dublincore.org/schemas/xmls/qdc/2003/04/02/dc.xsd

[FA] FAO. Fisheries Fact Sheet. http://www.fao.org/fi/website/FISearch.do?dom=factsheets

[FAOdiv] CWP Handbook of Fishery Statistical Standards. Fishing Areas for Statistical Purposes. http://www.fao.org/fi/website/FIRetrieveAction.do?dom=ontology&xml=sectionH.xml

[FISTAT] FAO. Fisheries and Aquaculture Department. Statistics. http://www.fao.org/fi/website/FIRetrieveAction.do?dom=topic&fid=16062

[FS] FAO. Fisheries fact sheets. http://www.fao.org/fi/website/FIRetrieveAction.do?dom=topic&fid=16062&lang=en

[FSdic] FIGIS XML. List of elements. http://www.fao.org/fi/figis/devcon/diXionary/figisdoc3.5.html

[FSschema] XML schema for Fisheries Fact Sheets. http://www.fao.org/fi/figis/devcon/schema/3_6/fi.xsd

[GAN04WW] Gangemi A. WonderWeb Deliverable D16: “Reusing semi-structured terminologies for ontology building: A realistic case study in fishery information systems“, http://wonderweb.semanticweb.org, 2004.

[HBFSS] Coordinating Working Party on Fishery Statistics (CWP). CWP Handbook of Fishery Statistical Standards. Partially available at: http://www.fao.org/fi/website/FISearch.do?dom=ontology

[HS07] World Customs Organizations. Harmonized Commodity Description and Coding System. 2007 Edition. http://www.wcoomd.org/ie/En/Topics_Issues/HarmonizedSystem/DocumentDB/TABLE_OF_CONTENTS_2007.html

[ISO2] International Standard Organization (ISO): ISO 3166 ALPHA-2, 2006.

[ISO3] International Standard Organization (ISO): ISO 3166 ALPHA-3, 2006.

[ISSCAAP99] FAO. International Standard Statistical Classification of Aquatic Animals and Plants (ISSCAAP). Version in use until 1999 available at: ftp://ftp.fao.org/FI/DOCUMENT/cwp/handbook/annex/AnnexS1listISSCAAPold.pdf

[ISSCAAP00] FAO. International Standard Statistical Classification of Aquatic Animals and Plants (ISSCAAP). Version in use from 2000 available at: ftp://ftp.fao.org/FI/DOCUMENT/cwp/handbook/annex/AnnexS2listISSCAAP2000.pdf

[ISSCFVgrt] International Standard Statistical Classification of fishery Vessels (ISSCFV) by GRT Categories. in use until 1995. ftp://ftp.fao.org/FI/DOCUMENT/cwp/handbook/annex/annexL1ISSCFVgrt.pdf

[ISSCFV] International Standard Statistical Classification of Fishery Vessels (ISSCFV) by Vessel Types, in use until 1995. 1984. ftp://ftp.fao.org/FI/DOCUMENT/cwp/handbook/annex/annexLII.pdf

[ISSCFG] International Standard Statistical Classification of Fishing Gear (ISSCFG) ftp://ftp.fao.org/FI/DOCUMENT/cwp/handbook/annex/AnnexM1fishinggear.pdf

[ISSCFC] FAO. International Standard Statistical Classification of Fishery Commodities: Divisions and Group. ftp://ftp.fao.org/FI/DOCUMENT/cwp/handbook/annex/ANNEX_RII.pdf

[M49] United Nations. UN Code (M49). http://unstats.un.org/unsd/methods/m49/m49alpha.htm.

[ONEF] One fish topic tree. http://www.onefish.org/global/index.jsp

[ONTO101] N. F. Noy and D. L. McGuinness. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[PER05] C. Perez and S. Conrad. Relational.OWL - A Data and Schema Representation Format Based on OWL. In Proc. of APCCM 2005.

[RT] FAO. Table Selector for Reference Tables. http://www.fao.org/figis/servlet/RefServlet

[SITC3] United Nations Statistics Division. Standard International Trade Classification, Revision 3. http://unstats.un.org/unsd/cr/registry/regcst.asp?Cl=28&Lg=1

[XMLSpy] Altova. XMLSpy. http://www.altova.com/products/xmlspy/xml_editor.html