

A Workflow for the Networked Ontologies Lifecycle. A Case Study in FAO of the UN

Óscar Muñoz-García (UPM), Asunción Gómez-Pérez (UPM), Marta Iglesias-Sucasas (FAO), and Soonho Kim (FAO)

Ontology Engineering Group - Universidad Politécnica de Madrid (UPM),
Food and Agriculture Organization of the United Nations (FAO)

Abstract. This document shows a preliminary framework for editing *networked ontologies* in the context of the NeOn project. The goal is to manage, in a collaborative way, multiple networked ontologies for large-scale semantic applications. This paper shows the main concepts on the editorial workflow and several lifecycle use cases. The ontologies produced with this framework will be used by the Food and Agriculture Organization of the United Nations (FAO) in many different large applications such the *Fisheries Stock Depletion Assessment System*[4]. Therefore a major goal for FAO is to have a strong and reliable ontology management system for editing the networked ontologies that applications will use as a basis. This framework for editing networked ontologies is being developed in the context of the NeOn Project¹. What we present here is a brief summary of the activities carried out in this project regarding user requirements and subsequent use case analysis.

Key words: networked, ontology, workflow, fisheries, lifecycle, FAO, NeOn

1 Introduction

The Food and Agriculture Organization of the United Nations (FAO) leads international effort to defeat hunger. The Organization acts as a neutral forum where all nations dialogue as equals to debate policy and negotiate agreements. FAO is also a source of knowledge and information to help developing countries and countries in transition modernise and improve Agriculture, Forestry and Fisheries practices and ensure good nutrition for all.

Efficiently managing information and knowledge is extremely important to FAO and that is reflected in Article 1 of its Constitution, which reads that "The Organization must collect, analyse, interpret, and disseminate information relating to nutrition, food Agriculture and development."

¹ For more information see: <http://www.neon-project.org>

In this line, in the Fisheries domain, one of the biggest challenges both at present and in the future is to manage the world's fish stocks for achieving long-term sustainable Fisheries. For this purpose, the Fisheries department of the FAO has several information and knowledge organization systems to facilitate and secure the long-term, sustainable development and utilisation of the world's Fisheries and Aquaculture. However, currently each system has its own community having each of them its own vocabulary, different languages, etc. This constitutes a separate knowledge collective.

Current FAO Fisheries systems manage and disseminate statistical data on fishing, GIS data, information on aquaculture, geographic entities, description of fish stocks, etc. Although much of the data are 'structured', they are not necessarily interoperable because they are expressed in different representation languages and according to different models, developed using different technologies or deployed in different platforms. These data sources could be better exploited by bringing together related and relevant information, along with the use of the Fisheries ontologies, to provide inference-based services, enabling policy makers and national governments to make informed decisions.

The current technical state does not solve the problem in applications where complex ontologies should be created and managed collaboratively and in highly dynamic, multilingual and constantly evolving environments. There are several tools such Protégé² for editing ontologies, *R₂O* [6] for making mappings between ontologies and data bases, RDF-Gravity for visualising³, the ontology alignment API and Server⁴, etc. Despite there are a lot them that solve many problems such ontology learning, ontology upgrade and ontology alignment, these tools are stand alone and make the process of managing ontological information very complex basing the interoperability between them in exporting and importing processes that sometimes degrades the information. With respect to methodologies, Methontology [7] and On-To-Knowledge [5] do not define a workflow for editing ontologies taking into account the roles involved in the ontology development. Also these methodologies are defined for building ontologies from scratch not taking into account the reuse of existing ones. All the aforementioned approaches do not consider collaborative and distributed construction of ontologies when developers are geographically distributed using different languages. In fact the first method that included a proposal for collaborative construction was Co4 [2] and the first tool was Tadzebao and WebOnto[8].

To solve the above problems, the goal of the NeOn project is to create the first ever service-oriented, open infrastructure, and associated methodology to support the development lifecycle for a new generation of semantic applications being the FAO case study a complex use case that will validate the NeOn methodologies and the NeOn Toolkit (which is on development, but includes several parts partially tested). NeOn provides FAO with a great opportunity to develop an appropriate framework to manage the Fisheries ontologies and their

² For more information see: <http://protege.stanford.edu>

³ See: <http://semweb.salzburgresearch.at/apps/rdf-gravity/index.html>

⁴ For more information see: <http://alignapi.gforge.inria.fr>

lifecycles, as well as to implement a semantic web Fisheries Stock Depletion Assessment System that exploits those ontologies.

In this context, one of the most important goals for FAO, is to develop a framework and support tools where ontologies can be modelled, populated, validated and deployed, and at the same time, mechanisms are put in place to facilitate that the existing ontological resources used by applications are maintained and kept up-to-date, and that when applying changes (to single ontologies or networks) all dependencies between systems continue to hold.

While requirements for ontology design, population and validation are common in ontology engineering environments, the FAO case study looks for a more articulated approach paying special attention to an editorial workflow, key to ensure that users can modify and update ontologies in a controlled and coherent manner, especially for those ontologies already deployed on the Internet. At the same time, this controlled environment for the editorial workflow will provide the necessary support to appropriately version ontologies deployed on the Internet, and to ensure semantic web applications reliability on the ontologies exploited.

2 Fisheries ontologies lifecycle

2.1 Users

The Fisheries ontologies lifecycle will be managed by a combination of two major types of users: ontology engineers and subject experts.

Ontology engineers are specialised in ontology modelling techniques and issues; have from basic to advanced knowledge of ontology engineering tools and inference engines, but may know little about the domain to be modelled. Usually, they are in charge of defining the initial skeleton of the ontology, and in so doing, they take into account the purpose of the ontology, possible interactions with legacy systems, and other relevant issues.

Ontology editors are domain experts, although they can also be information management specialists, terminologists or translators. they are in charge of the everyday editing and maintenance work of the networked multilingual ontologies and they can be in charge of developing specific fragments of ontologies, revising work done by others, and developing multilingual versions of ontologies.

2.2 Roles

Users participating in the Fisheries Ontologies Lifecycle will need to be authorised in the system to get access rights by the system Administrators. Authorised users will be assigned roles to various ontology modules as either Ontology engineers, Subject experts, Validators or Viewers, depending on the kind of rights they will have and the kind of tasks they will be assigned to.

Subject expert, validator and viewer correspond to the possible roles of the Ontology editors within the editorial workflow.

- Subject experts are the editors inserting or modifying ontology content.

- Validators revise, approve or reject changes made by subject experts, and they are the only ones who can copy changes into the production environment for external availability.
- Viewers are users authorised to enter in the system and consult approved information about ontologies but they cannot edit the ontologies.

2.3 Major Processes

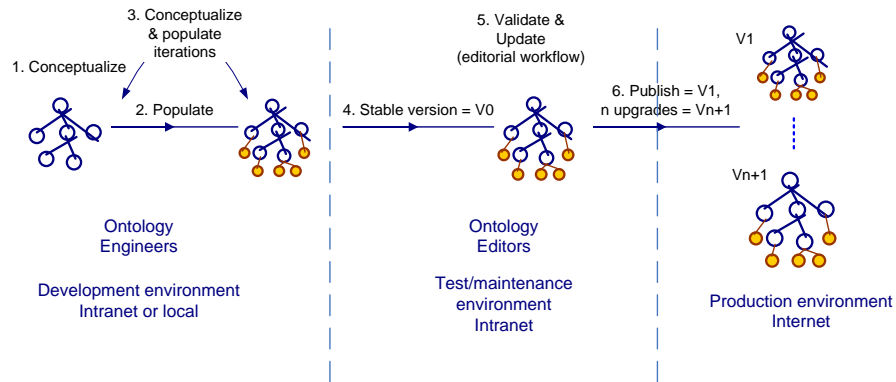


Fig. 1. Fisheries Ontologies Lifecycle

As illustrated in figure 1, the Fisheries ontologies lifecycle consists of the following major processes:

1. Ontology conceptualisation: Ontology engineers organise and structure the domain information into meaningful models at the knowledge level. In the fishery domain, they collect the information from Fisheries databases, information system and documents, and analyse it together with Fisheries domain experts in FAO. The conceptualisation process results in an ontology model with most of the concept level entities, such as, classes, properties and restrictions.

2. Ontology population: Ontology engineers perform the knowledge acquisition activities with various manual or (semi)automatic methods various methods to transform unstructured, semi-structured and/or structured data sources into ontology instances. In the Fisheries domain, this process consist mainly in converting semi-structured data sources (fishery fact sheets in XML format) and structured data source (from relational databases) into corresponding instances in the conceptualised Fisheries ontology. Figure 2 shows the possible population sources.

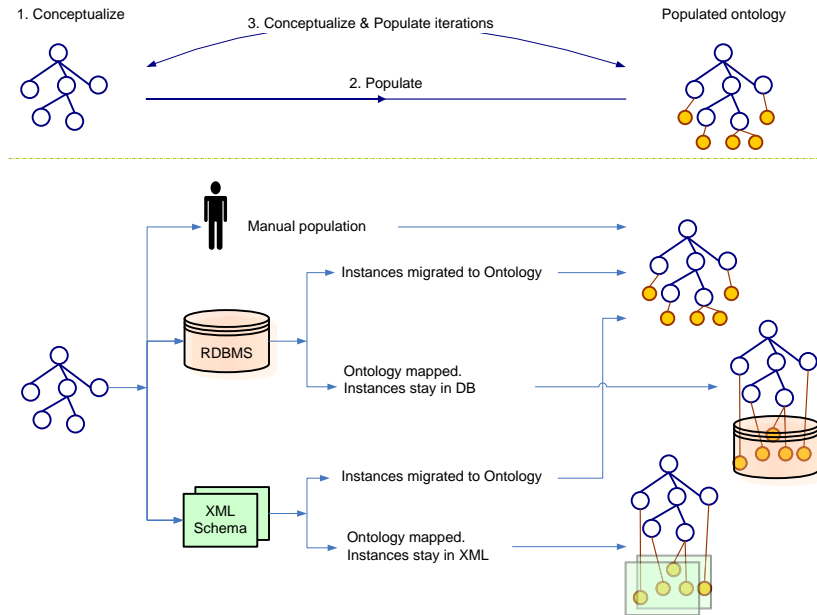


Fig. 2. Ontology Population

3. and 4. Iteration of conceptualisation and population process until getting a stable version: Ontology engineers will iterate the conceptualisation and population processes until getting a populated ontology that satisfies all requirements and it is considered stable. Once achieved, the ontology will enter into the test and maintenance environment, implemented through the editorial workflow.

5. Ontology validation and update through editorial workflow: The editorial workflow will allow Ontology editors to consult, validate and modify the ontology keeping track of all changes in a controlled and coherent manner. Any ontology to be released on the production environment needs to pass through the editorial workflow being it the first time for version 1 or for any subsequent upgrade. The editorial workflow is explained in detail in the following section.

6. Ontology publication Once ontology editors in charge of validation consider the ontology final, they are authorised to release it on the Internet and make it available to end users and systems. A release will consist in making a copy of the ontology in the maintenance environment into the production environment, which in the case of FAO will be the Internet. Ontologies published on the Internet will be always versioned, from 1 for the first published version to N+1 to the N upgrade of the ontology. All versions will be available all the

time with necessary metadata in order to ensure that semantic third party semantic web applications relying on a particular version will keep working relying on a previous version independently of the new one until a decision is made to upgrade the application, if required, to use the new ontology version.

2.4 Editorial workflow

The Fisheries editorial workflow will implement the necessary mechanisms to allow Ontology editors to consult and if authorised, validate and/or modify the ontology in a controlled and coherent manner, ensuring that only fully validated ontologies will be released on the Internet.

The workflow is based on the assignation of a status to each element of the ontology. Only if all the elements have "Approved" status, the ontology can be published or upgraded.

The possible statuses for each element are:

- **Draft:** this is the status assigned to any element when it pass first into the editorial workflow, or it is assigned to a single element when it was approved and then updated by a subject expert.
- **To be approved:** once a subject expert is confident with a change in draft status and wants it to be validated, the element is passed to the to be approved status, and remains there until a validator accepts it.
- **Approved:** if a validator accepts a change in an element in the to be approved status, this passes to the approved status.
- **Published:** this is the status of all elements in an ontology released to the Internet.
- **To be deleted:** if a subject expert considers that an element needs to be deleted, the item will be flagged with the "to be deleted status" and removed from the ontology, although only a validator would be able to definitively delete it.

The workflow then will allow to set up who (depending on the user role) can do what (actions as explained below) and when (depending on the status of the element and the role of the user).

Subject experts will be able to:

- **Insert** a new element, or **Update** an approved element. In both cases the system will automatically assign a Draft status to the element. These two actions triggers the start of the workflow.
- **Send to be approved:** the subject expert changes the status of an element from Draft to To be approved. This automatically moves the responsibility on the item from the subject expert to the validator.
- Delete an approved element, which will be sent to **To be deleted** status; or delete an item in Draft status, which will be automatically deleted.

Validators will be able to:

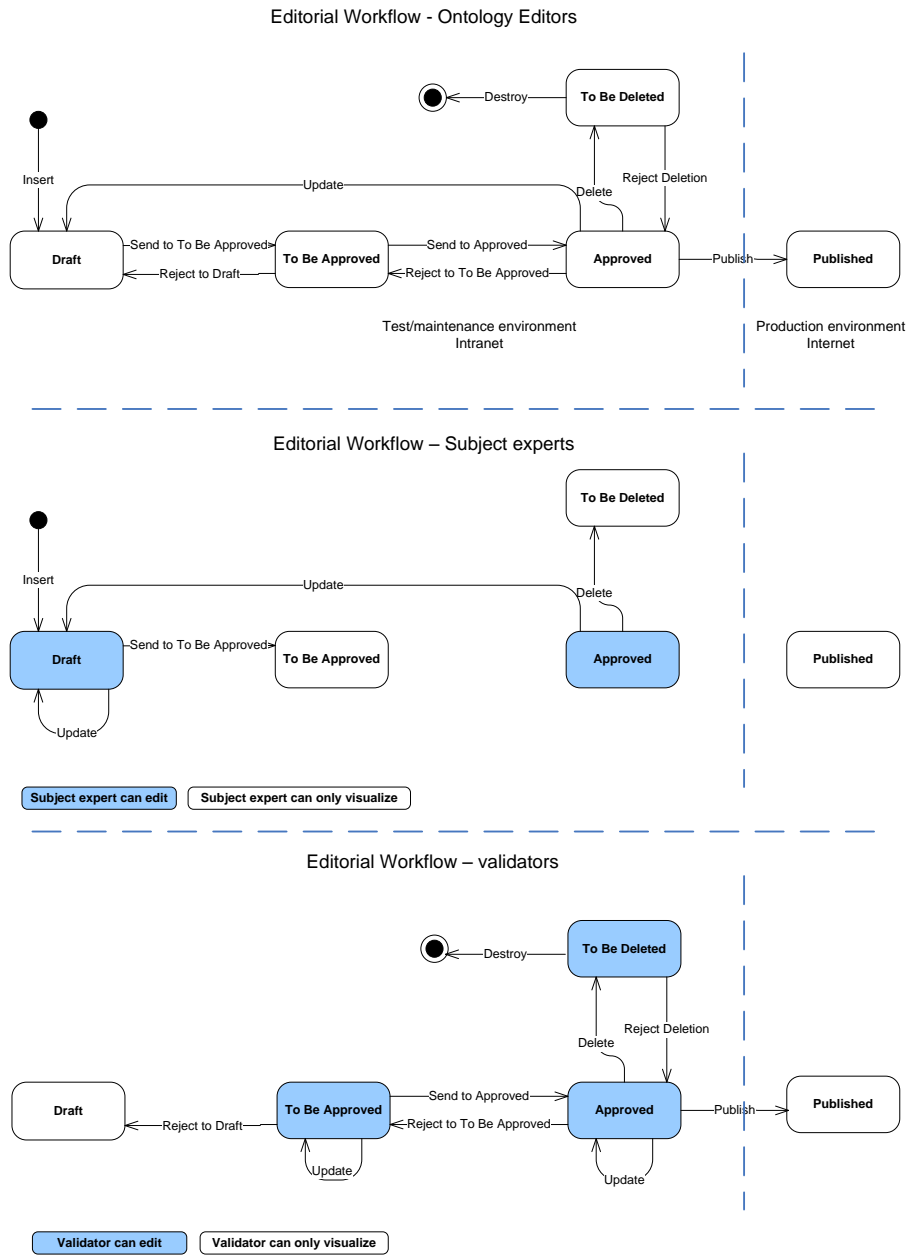


Fig. 3. Editorial Workflow

- **Update** an approved or a to be approved element. Being the validator doing the modification, and not needing to be double checked by other validators, the element will remain in the same status as it was.
- If an element is in the To be approved status, the validator can either accept it, so it will be **Send to the Approved** status; it can be not accepted, then it will be **Rejected to draft** status, or the validator can modify it.
- If an element is in the Approved status, the validator can either send it back to To be approved, so it will be **Rejected to To be approved** status, can delete it and send it to the bin or the **To be deleted** status or the validator can modify it.
- **Delete** an element in the Approved and **Destroy** an element in the To be deleted status.
- If the validator does not agree with an element proposed To be deleted by a subject expert, and thus in the To be deleted status, the validator can **Reject the deletion**, and pass back the element to the Approved status.
- When all the elements of the ontology are approved the validator can decide to **Publish** it. This action will copy the Approved ontology into the production environment assigning it the right version, V_1 for the first release and V_{N+1} for N subsequent releases.

3 Use Cases

A model of the system’s functionality and environment has been developed following the Unified Process methodology [1] coming from software engineering. This section makes a brief summary of the Use-Case Model obtained.

Next we put a description of the most relevant use cases. These use cases take the NeOn metamodel as a basis. This networked ontology model has been designed in the NeOn project and is derived from the modeling primitives offered by OWL[3].

1. **Search:** While editing an ontology, the Ontology Editor is able to perform searches across the whole ontologies being edited, independently of whether the text appears in a concept label, annotation, property name, etc.
2. **Answer Query:** While editing an ontology, the Ontology Editor is able to perform queries within the ontologies being edited. The queries could be using and standard query language (e.g. SPARQL), a natural language query or a predefined query from a template . As an example, these constraints or predefined queries could be:
 - For concepts: "having parent such that ...", "having child such that ..."
 - For instances: "being an instance of ..."
 - For properties: "attached to ...", "linking..."
3. **Manage Multilinguality:** The Ontology Editor deals with the multilingual aspect of the ontologies adding languages to the ontology; doing spell-checking, managing the multilingual labels, selecting the working language, and coping with specificities of translation (i.e., no lexicalization available for concepts, available lexicalization correspond to more than once concept or conversely, several lexicalizations are possible).

4. **Export:** exporting an ontology to other formats. In example exporting ontologies into thesaurus format, which implies conversion to: TagText, RDBMS, ISO2709, SKOS and TBX.
5. **Convert:** convert an ontology from other formats, including population from databases using *R₂O* [6] and from existing resources with implicit schema (XML).
6. **Manage Mappings:** creation of alignments between ontologies in a manual way and an semi-automatic way. Mappings between concepts or modules in different ontologies are created. For the creation of an automatic alignment the Ontology Editor gives the System two ontologies. The System returns the Ontology Editor the candidate mappings. The Ontology Editor inspects the proposed candidates one by one selecting the appropriate candidate and confirming the proposed mapping. Finally the System creates the alignment taking into account the mappings chosen.
7. **Visualize:** visualisation of ontologies and fragments of them in different ways, depending on the task to be performed. Mappings and relations between the concepts and modules in networked ontologies are visualised. Browsing an ontology and printing the visualisation is included.
8. **Modularize:** working with ontology modules; creation of modules manually and semi-automatically and merging modules. For more information about an ontology module is please see [3].
9. **Manage Provenance and Statistics:** the System captures ontology changes. The users can see the changes history, view use statistics (provenance, which system they are used by, by whom they are edited, frequency of changes, fragment/domain of the ontology changed at fastest rate) and view ontology statistics (depth, number of child nodes, number of relations and properties, number of concepts per "branch").
10. **Populate from text:** the Ontology Editor chooses the textual corpora. The System provides Ontology Editor with a list of candidate elements of the ontology (classes, instances and relations between concepts). The System shows the documents and excerpts supporting the extracted terminology, including the document metadata such as title of the document, author, data, owner, publication date. The Ontology Editor inspects and selects the appropriate candidate, and adds the selected ones to the ontology. The System populates the ontology doing previously a consistency checking of the ontology with the newly added elements.
11. **Evaluate and Validate Ontology:** the Ontology Editor can check the quality of the development of the ontology, checking for duplicates within the ontology, making comparisons with other ontologies and evaluating structural properties of the ontology.
12. **Obtain Documentation:** automatic creation of relevant metadata concerning the ontology design, such as UML-like diagrams, and documentation concerning the relations and properties used.

4 Conclusions

In this paper we have described the lifecycle needed for managing the networked ontologies that are used by the Food and Agriculture Organization of the United Nations. We have focussed the description in the editorial workflow and also we have enumerated some relevant use cases that describe the features demanded by FAO in order to create and maintain the ontologies.

The current technical state is not enough to cover the needs because there is not an integrated tool that provides all the features needed. So we have introduced the NeOn Toolkit that is been developed in the context of the NeOn project where FAO takes part as a case study partner.

Acknowledgements

This work has been supported by the NeOn project (IST-2005-027595). We are very grateful to our NeOn partners for their revisions and comments.

References

1. Applying UML and patterns: an introduction to object-oriented analysis and design and iterative development. Craig Larman. [ed.] Don O'Hagan. Third Edition. Upper Saddle River : Prentice Hall, 2005. ISBN 0-13-148906-2.
2. Building Consensual Knowledge Bases: Context and Architecture. 1995. Euzenat J. In: Mars N (ed) Second International Conference on Building and Sharing of Very Large-Scale Knowledge Bases (KBKS '95). University of Twente, Enschede, The Netherlands. IOS Press, Amsterdam, The Netherlands, pp 143155
3. D1.1.1 Networked Ontology Model v1. Technical report. NeOn. 2006.
4. D7.1.1 Specification of users and user requirements. Technical report. NeOn. 2006.
5. Knowledge Processes and Ontologies. 2001. Staab S, Schnurr HP, Studer R, Sure Y. IEEE Intelligent Systems 16(1):2634
6. Modelo para la definición automática de correspondencias semánticas entre ontologías y modelos relacionales. December 2006. Jesús Barrasa Rodríguez. PHD Thesis.
7. Ontological Engineering. 2003. Gómez-Pérez A, Fernández-López M, Corcho O, Springer-Verlag.
8. Tadzebao and WebOnto: Discussing, Browsing, and Editing Ontologies on the Web. 1998. Domingue J. In: Gaines BR, Musen MA (eds) 11th International Workshop on Knowledge Acquisition, Modeling and Management (KAW'98). Banff, Canada, KM4:120