ONE Record Ontology

Development & Maintenance process

Scope

This document provides a brief overview of the ONE Record process for developing ontologies. It contains a minimal amount of technical detail sufficient to explain our approach to ontology development.

ONE Record Ontology overview

- ONE Record provides two ontologies:
  - A core air cargo ontology for describing the core information to be used in ONE Record ecosystem. This core ontology can be extended for more specific domains and applications (e.g., pharma, IoT, dangerous goods, etc.). The documentation of the Air Cargo ontology (often referred to as ONE Record Data Model) can be found on the ONE Record Developer Portal.
  - A second ontology for describing models purely related to the ONE Record API functionalities, such as error, publisher/subscriber models, company information, etc. The documentation of the API ontology can be found on the ONE Record Developer Portal.
- The ontologies are stored in the IATA ONE Record GitHub repository.
- Improvements and issues for each domain are discussed in the repository's GitHub issue tracker, as well as in the ONE Record Data Model Working Group.
- The requirements of each domains are stored in a spreadsheet in the GitHub repository. When the ONE Record Data Model Working Group creates a new version of the spreadsheet, a new release of the ontology is created in Turtle format by IATA ONE Record technical experts and added to the repository.
- The tool used for the ontology creation/update is Protégé.
- The ontology developers have to evaluate the ontology and generate its documentation before publishing it. The documentation is generated automatically by copying the Turtle files from the main ONE Record repository to the one-record-ontologies-widoco repository.
- The releases versions of the domains are published in the ONE Record Developer Portal, by copying the generated files in the one-record-ontologies-widoco repositories to the Developer Portal source code.
- OnToology will generate the documentation and the evaluation of the ontology.

Ontology Versioning

Overview

Versioning is needed in order to manage complexity and breaking changes within the Data Model and API specifications, and to iterate faster when breaking changes are identified.

Some possible changes in the ONE Record Data Model (ontology):

- Addition/Removal of Classes/Properties;
- Updates of cardinality/data types;
- Updates of comments/labels.

The solution should:

- Reference the correct documentation;
- Be preserved over time;
- Have the ONE Record API Version in line with the ONE Record Data Model Version.

ONE Record Data Model Versioning

The ONE Record ontology is developed incrementally. The ontology has several releases, which are stored in the GitHub repository.

Data Model Versioning:

- Each release (e.g. 1.0) should have a name that is assigned upon publication and endorsement by the competent IATA governing board, in this case the COTB, comprised of members of the industry ;
- The content of each release is based on the discussions with the domain experts and the discussions hosted on the GitHub site;
- No need for versioning at class/property level;
- Use the **owl:versionInfo** annotation at the ontology level for expressing the version;
- Makes sure that the release notes of every version of the ontology contain enough information about what has changed between the last version and the current one;
- First version of the standard (Aug 2020) - Endorsed version in Mar2020 = 1.0;
- Use standard approach for minor/major versions;
- At the moment, Deprecated Property/Classes annotations are not used. To be used at a later stage if needed;
- Classes and Properties to be removed are labelled with “Deprecated” text and made Optional (meaning the minimum cardinality is set to 0).

Actions on classes and data properties in comparison with Version N – 1		
New	Directly added	
To be removed	Labeled as “Deprecated” and Optional	“Deprecated” items are removed
Modification of Name or datatype	• Addition of new classes/data properties with the right name/datatype • Previous are labeled as “Deprecated” and Optional	“Deprecated” items are removed
Modification of cardinality to Mandatory	Remain Optional with a comment for version N+1	Cardinality is changed to Mandatory

ONE Record Ontology Versioning

ONE Record GitHub repositories

ONE Record standard repository

The tool selected to store the ONE Record ontologies is GitHub. The ONE Record standard documents such as the Data Model documents, ontologies, JSON-LD samples, API & Security specifications can be found under: https://github.com/IATA-Cargo/ONE-Record.

All the new versions of the documents and ontologies should be added under the working_draft folder.

GitHub flow for ontology development

The ontologies are developed incrementally. For each ontology we may have several releases, which are stored in the GitHub repositories. Each release has a name that is assigned upon publication (e.g. v0.0.1). The content of each release is based on the discussions with the domain experts and the discussions hosted on the GitHub site.

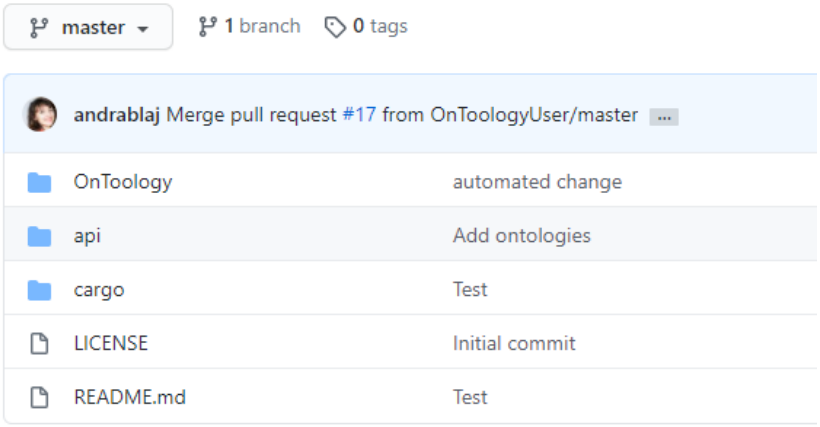
For developing or updating an ontology using GitHub, we propose to follow some steps:

- The central repository holds a main branch called *master* where the source code reflects the production-ready state. To work on something new, the ontology developers have to create a descriptively named branch off the *master*, so that the rest of the developers can see what is being worked on.
- Once the branch is created, the developers add changes to the ontology and commit them. Each commit has to be associated with a commit message, which is a description explaining why a particular change was made, so that the developers can roll back changes if a bug is found.
- After adding commits, the ontology developers have to open a pull request to discuss the modifications done to the ontology. After creating the pull request, some tests are executed (see next section) to check if the ontology meets all the requirements identified.
- If everyone agrees and the ontology passes the tests, the pull request is accepted and the changes are merged to the *master* branch.

OnToology documentation repository

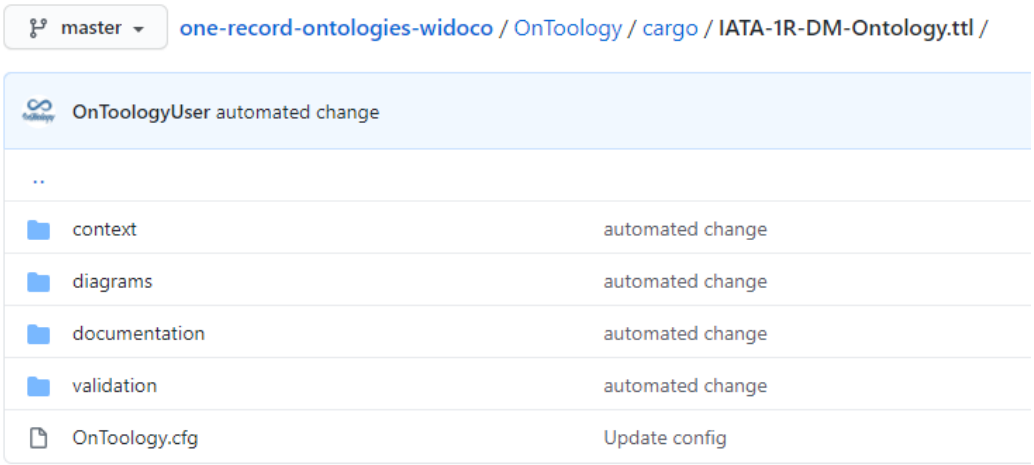
Once the ontology is implemented (for this step the developers can use an ontology editor like Protégé), it has to be evaluated and documented before its publication.

In order to automatically generate the documentation for the ONE Record ontologies, the OnToology tool is used via the https://github.com/IATA-Cargo/one-record-ontologies-widoco GitHub repository.



Structure of the repository

- At each release, a new branch should be created from the *master* branch;
- After the new branch is created, the ONE Record ontologies should be copied from main ONE-Record repository to this one in order to automatically get the updated documentation as follows:
  - The Air Cargo ontology should be copied under the /cargo folder;
  - The API ontology should be copied under the /api folder;
- After the files are copied, the developers should commit the change and create a pull request;
- After creating the pull request, some tests are automatically executed by OnToology to check if the ontology meets all the requirements identified;
- If all tests pass, the documentation is generated under the /OnToology folder.
- The structure if the generated documentation is as in the image below:



Structure of the generated documentation

Step-by-step documentation generation with OnToology

In order to connect the one-record-ontologies-widoco repository to OnToology, the IATA technical developer needs to connect via IATA ONE Record team's GitHub identifier and follow the steps here.

Ontology Evaluation

The ontology has to be evaluated according to syntactic, model and semantic errors. OnToology will evaluate the ontology using Oops! and generate an evaluation report every time there is a push in the repository. The ontology developers must also guarantee that the ontology meets all the requirements identified.

Ontology Documentation

OnToology will generate the documentation using Widoco every time there is a push in the repository. This documentation includes an HTML description of the ontology which describes the classes, properties and data properties of the ontology, and the license URI and title being used. The ontology development team in collaboration with the domain experts can complete this HTML documentation. OnToology will also generate the diagrams using AR2DTool. It will generate two kinds of diagrams:

- Class diagram
- Taxonomy diagram

Note: These diagrams are currently hidden in the ONE Record ontologies documentations, as they are difficult to read, due to the large number of classes and properties.

Developer Portal repository

The generated documentation is copied from the one-record-ontologies-widoco repository to the private IATA Developer Portal repository: https://github.com/IATA-Cargo/one-record-server-java-test.

Ontology Publication

Releases are published on the official site – the ONE Record Developer Portal by the IATA responsible technical expert, so the ontology and its associated documentation will be accessible to all the users. The site will publish additionally for each ontology the repository where is stored, the releases created, and the requirements associated to each ontology.

Ontology Maintenance

If the ontology developers or domain experts want to update or add new requirements to the ontology they have to create a new issue in the GitHub issue tracker. This issue will let the technical experts to start a discussion and accept or reject the modifications. GitHub also provide a good practices guide for creating issues .