OnToology Drag&Drop [⋆]

Adrián Mora Carrero[1][0000−0002−0511−2488], Ahmad Alobaid[1][0000−0001−8637−6313], and María Poveda-Villalón[1][0000−0003−3587−0367]

Universidad Politécnica de Madrid, Boadilla del Monte, Madrid, Spain
a.mcarrero@alumnos.upm.es
{aalobaid,mpoveda}@fi.upm.es

Abstract. The goal of this demo paper is to present OnToology Drag& Drop, a web application that seeks to facilitate the integration of several ontology development activities (documentation, validation, evaluation and shapes generation) without the need for installation of any software and without the dependency of git system.

Keywords: Ontology engineering · Ontology documentation · Ontology evaluation · Ontology constraints

1 Introduction

There is a variety of ontology development activities that are normally carried out in any development project, for example generating adequate documentation, designing and executing tests, or identifying the errors of the ontologies, etc. In addition, such activities might be repeated several times within a project lifetime when following iterative or incremental methodologies. To support these scenarios and align the ontology development with software practices, several tools have been developed in order to orchestrate the ontology development activities like OnToology [1] or VoCol [5]. These systems allows for continuous integration of ontology development artefacts integrating support for several activities by implementing new features or integrating existing tools. In both cases, the functionality is built on the basis of git systems. While this solution is robust for ontology maintenance and traceability of an ontology project, in some cases it imposes some requirements or overloads not desired by users looking for a more lightweight platform. For example, some users that are not handling the ontology in git repositories, or do not desire to make them public.

For this reason, in this demo we present OnToology Drag&Drop[1], a web application to generate ontology documentation, evaluate ontologies, run validation tests and generate SHACL Shapes[2] from ontologies. OnToology Drag&Drop is available as a web interface where the user can drag and drop the ontology files and easily select which services to apply over each ontology without having to install any application or be tight to git systems.

[⋆] This work has been supported by the Spanish project DATOS 4.0: RETOS Y SOLUCIONES (TIN2016-78011-C4-4-R).

[1] https://ontoologydd.linkeddata.es/

[2] https://www.w3.org/TR/shacl/

2 Features

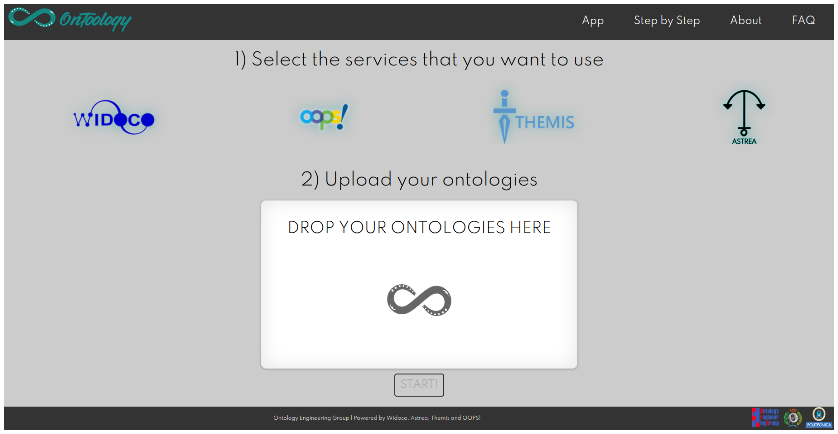
As we have mentioned in the introduction, the power of this tool lies in being able to easily run a set of utilities on our ontologies, simply by dragging them into the web application. Specifically, OnToologyD&D integrates the following features provided by existing tools:

- Generate ontology HTML documentation: through Widoco [4], the system allows users to produce HTML documentation for the ontologies that can be published using the htaccess file for content negotiation provided within Widoco output.
- Ontology evaluation: this functionality is provided by integrating OOPS! [7] to generate an error report of the given ontology provided as an HTML document containing the potential errors that have been found in the ontology.
- Ontology validation: this feature is possible integrating Themis [3] to allow users to run tests in order to check that the ontology meets the requirements represented in the tests. It is also possible to generate basic template tests from the ontology to be later adapted by users to their needs.
- Ontology shapes generation: this capability is provided by integrating Astrea [2] to generate SHACL shapes [6] (Shapes Constraint Language) from the uploaded ontologies.

The presented application it is intended to facilitate the user the use of all these technologies, having a common point to access them and avoiding installation processes or handling several applications in different environments and without dependencies with git systems. It also saves valuable time since the application is designed to minimize the number of steps needed to select the features and generate the output.

The application allows the user to select any combination of tools to be run for one or more ontologies. When all the ontologies are uploaded, the system orchestrates the calls to all the applications and the output is generated as a zip file that contains a folder for each ontology and the services selected for them, in a short time (depending on the ontology number and sizes).

The flow of the application is simple, the user can choose an initial configuration of services that it want to run by enabling/disabling the application logos in the first screen (Figure 1). After this, it can be dragged to the application the ontologies. Later on, the user can modify the configuration for each ontology and the change will be applied to all the ontologies uploaded after the change. The selection can be reverted any time before running the application for all the previously uploaded ontologies (Figure 2). If the user has selected Themis, (s)he is asked to upload a file with the tests, otherwise, template test will be generated. After this, simply click on the start button and the files will be generated.



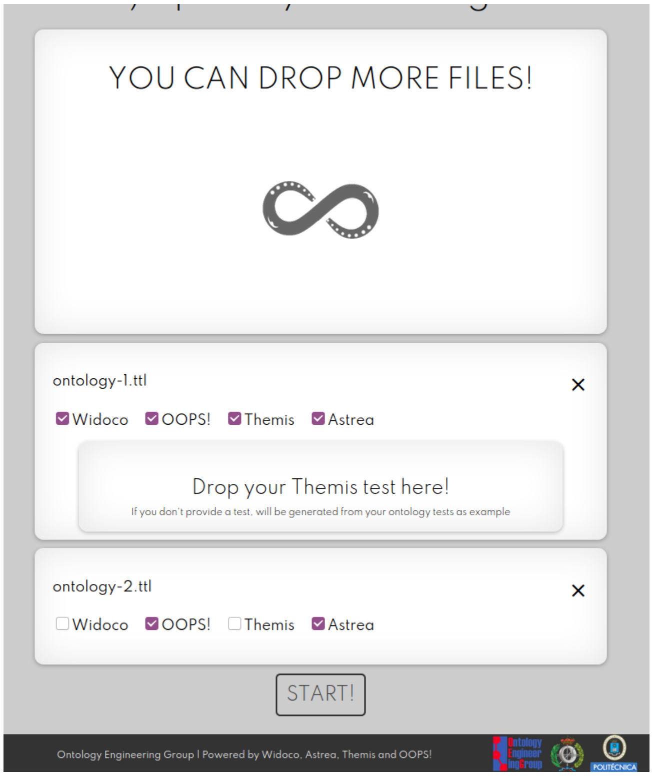
1) Select the services that you want to use

2) Upload your ontologies

DROP YOUR ONTOLOGIES HERE

Ontology Engineering Group | Powered by Widoco, Astrea, Themis and OOPS!

Fig. 1. OnToology Drag&Drop app



YOU CAN DROP MORE FILES!

ontology-1.ttl

☑ Widoco ☑ OOPS! ☑ Themis ☑ Astrea

Drop your Themis test here!

If you don't provide a test, will be generated from your ontology tests as example

ontology-2.ttl

□ Widoco ☑ OOPS! □ Themis ☑ Astrea

Ontology Engineering Group | Powered by Widoco, Astrea, Themis and OOPS!

Fig. 2. OnToology Drag&Drop app. Ontologies uploaded

3 Architecture

OnToology Drag&Drop is a web application, developed with JAVA in its backend and HTML, CSS and JavaScript in its frontend.

The application works, in summary, as follows: when all the ontologies have been sent, the frontend, which has stored the services configuration selected by the user, makes a set of calls to the server to generate all the files. When all the tools have been executed, a ZIP file is generated.

On a technical level, the application works this way, the ontology, when it arrives at the server, is checked by a parser, if it is correct it is stored and the tools are later executed on it. The server executes Widoco, OOPS!, Themis or Astrea, as applicable, making calls to the REST APIs that provide those services or using Java programs, with an appropriate configuration to deal with the ontologies without the need for further worries by the user. After having generated all the files in a suitable folder structure, this is compressed and automatically downloaded to the user's computer. Widoco is executed through a Java ARchive (JAR), on the other hand OOPS!, Themis and Astrea are executed by web services.

4 Demonstration

During the demonstration of the application two ontologies will be uploaded, one in XML/RDF and another in TURTLE, all the available services will be selected for one of them, and for the remaining one a subset of them. In both cases, Themis will be selected in order to show the two facets of the service, the first by uploading a file with our own tests and verifying the results; and the second by letting themis provide some sample tests.

In any case, attendants will be able to see from the beginning how to select the services to be run, how to upload the ontologies, how to modify the services independently for each ontology, how to upload Themis tests and, finally, they will be able to observe all the files generated thanks to the application.

The aforementioned demo can be seen in brief in the following video.[3]

5 Conclusions

With this alternative of OnToology, a lighter version of the app is provided with the focus on developments that are carried out by small teams or by single persons, and therefore do not use Git systems. In addition, the system is designed to be expandable, that is, it is implemented in such a way that adding a new service to the server is simple, since they are unconnected and without need to modify the code already written. To include a new system it is required to adapt the web interface in order to add the corresponding calls to that new application

[3] https://www.youtube.com/watch?v=rKVeGFGRYhk

or web service which implements the new functionality. The code is available for the community to use, modify or contribute on GitHub under an open license.[4]

As future lines of work it should be explore how to incorporate Widoco configuration options to the current features of OnToology D&D as well as the inclusion of other tools.

References

1. Alobaid, A., Garijo, D., Poveda-Villalón, M., Santana-Perez, I., Fernández-Izquierdo, A., Corcho, O.: Automating ontology engineering support activities with ontoology. Journal of Web Semantics **57**, 100472 (2019)
2. Cimmino, A., Fernández-Izquierdo, A., García-Castro, R.: Astrea: Automatic generation of shacl shapes from ontologies. In: European Semantic Web Conference. pp. 497–513. Springer (2020)
3. Fernández-Izquierdo, A., García-Castro, R.: Themis: a tool for validating ontologies through requirements. In: SEKE. pp. 573–753 (2019)
4. Garijo, D.: Widoco: a wizard for documenting ontologies. In: International Semantic Web Conference. pp. 94–102. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68204-4_9
5. Halilaj, L., Petersen, N., Grangel-González, I., Lange, C., Auer, S., Coskun, G., Lohmann, S.: Vocol: An integrated environment to support version-controlled vocabulary development. In: European Knowledge Acquisition Workshop. pp. 303–319. Springer (2016)
6. Knublauch, H., Kontokostas, D.: Shapes constraint language (shacl). W3C Candidate Recommendation **11**(8) (2017)
7. Poveda-Villalón, M., Gómez-Pérez, A., Suárez-Figueroa, M.C.: OOPS! (OntOlogy Pitfall Scanner!): An On-line Tool for Ontology Evaluation. International Journal on Semantic Web and Information Systems (IJSWIS) **10**(2), 7–34 (2014)

[4] https://github.com/oeg-upm/OnToologyDnD