Converting UML-Based Ontology Conceptualizations to OWL with Chowlk

Serge Chávez-Feria(✉), Raúl García-Castro, and María Poveda-Villalón

Ontology Engineering Group, Universidad Politécnica de Madrid, Madrid, Spain
serge.chavez.feria@upm.es, {rgarcia,mpoveda}@fi.upm.es

Abstract. During the ontology conceptualization activity, developers usually generate preliminary models of the ontology in the form of diagrams. Such models drive the ontology implementation activity, where the models are encoded using an implementation language, typically by means of ontology editors. The goal of this demo is to take advantage of the developed ontology conceptualizations in order to accelerate the ontology implementation activity. For doing so we present Chowlk, a converter to transform digital UML-based ontology diagrams into OWL. This system aims at supporting users in the generation of the first versions of ontologies by reusing the ontology conceptualization output.

Keywords: Ontology engineering · Ontology conceptualization · Ontology implementation

1 Introduction

One important step in ontology development is the conceptualization one, during which the ontology development team defines a set of concepts and properties to represent the knowledge of an specific domain. Often, this conceptualization is materialized in a diagram that displays the ontology elements and their connections. From this model, the ontology implementation is carried out normally using an ontology editor, encoding the model in OWL. In this process the diagram is in most of the cases only used as a guideline to manually implement the ontology. To address this issue, some tools have been proposed to allow the graphical creation or modification of ontologies [2–5].

The present work aims at leveraging the above-mentioned diagrams in order to allow for a smoother transition from the conceptualization activity to the actual implementation, that is transforming XML diagrams following a UML-based ontology visual notation into OWL. For doing so, rather than building a graphical ontology editor, the process builds on top of a well-adopted system such as diagrams.net which allows collaborative and synchronous edition of the conceptualization models.

This work has been supported by the BIMERR project funded from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 820621.

R. Verborgh et al. (Eds.): ESWC 2021 Satellite Events, LNCS 12739, pp. 44–48, 2021.
https://doi.org/10.1007/978-3-030-80418-3_8

2 Chowlk Features

Chowlk is a web application that takes as input an ontology conceptualization created with diagrams.net and generates the OWL implementation. The service is available online[1] (See Fig. 1) and the source code is shared in a GitHub repository[2] under the Apache 2.0 license.

The conceptualization should follow the Chowlk visual notation[3] which is also provided as a diagrams.net library[4] to allow users to easily reuse the correct shapes to avoid problems during the transformation and save to time during the conceptualization.

The converter is able to identify concepts, object properties, datatype properties, and restrictions between those elements. Also, the converter identifies ontology metadata, namespaces and the prefixes being used in the model, due to specific blocks dedicated to this type of information. Labels to each ontology element are added during the detection process. Once the XML diagram is loaded into the system, Chowlk starts searching for all the ontology elements in the conceptualization ignoring shapes not included in the specification. After the detection and creation of the corresponding associations between the ontological elements, Chowlk proceeds to write the implementation using the OWL language. Finally, the ontology is provided in two downloadable formats: Turtle and RDF/XML (see Fig. 2).

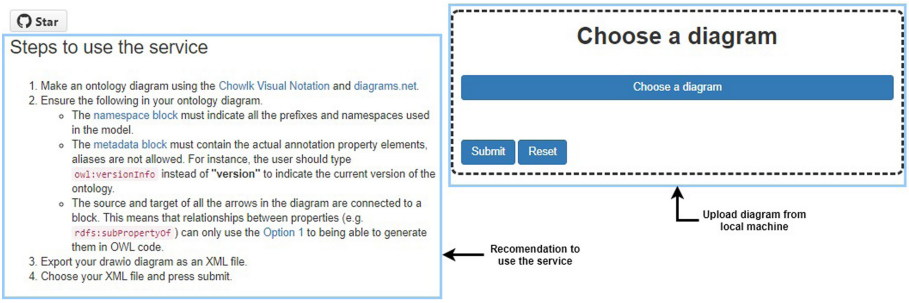

Fig. 1. Screenshot excerpt of Chowlk home page

[1] https://chowlk.linkeddata.es.
[2] https://github.com/oeg-upm/Chowlk.
[3] https://chowlk.linkeddata.es/chowlk_spec.
[4] https://github.com/oeg-upm/chowlk_spec/blob/master/chowlk-drawio-library.xml.

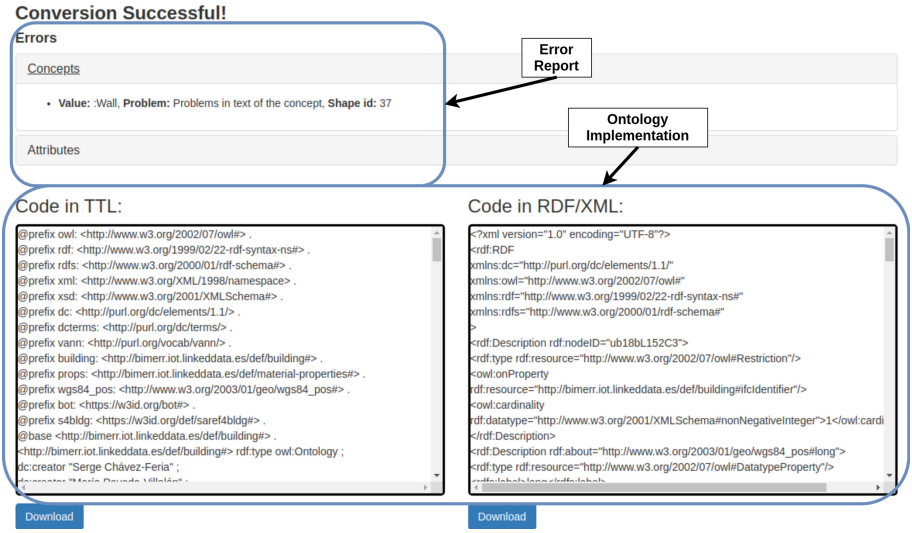

Fig. 2. Screenshot of the web GUI showing the output serializations.

3 Architecture

The architecture of the Chowlk converter is shown in Fig. 3. It has been developed as a single page web application where the user can upload its ontology conceptualization and the converter returns the corresponding OWL ontology. The conceptualization diagram should be provided as an XML file including only the parts of the diagram intended to be translated. Once the conceptualization is parsed, using standard XML Python libraries, the converter proceeds with the detection of the ontology elements inside the diagram. The detection module analyzes the fields inside the XML structure of each shape in order to make the appropriate mapping with the OWL elements. The information inside the XML should follow the Chowlk visual notation that provides the specification with respect to the type and style of the shapes to be used. Shapes that do not follow the visual notation are discarded during the transformation process and passed to the writing module as an error report.

The association module performs the connection between the ontology elements identified by the detection module. For each concept, the module iterates over the object properties and datatype properties in the conceptualization and identifies which ones are connected physically in the diagram. Afterwards, OWL restrictions are detected between the concepts and the properties associated previously by identifying keywords such as “some” or “all” in the properties text. If no restriction is specified then the association is discarded even if the elements are connected. The reasoning behind this is that in the diagram a modeller can make recommendations about which properties are expected to be used for a concept but if it is not formalized, by means of a class restriction (universal,

existential or cardinality), it has no effect on the OWL implementation further than the declaration of the class and the property.

Finally, the writing module receives the information associated from the previous module and performs the serialization of the model into two different formats (Turtle and RDF/OWL) which are presented to the user through the web interface. The output interface also displays the possible errors that the diagram could contain, indicating not only the problematic shapes but also the possible cause of the problem.

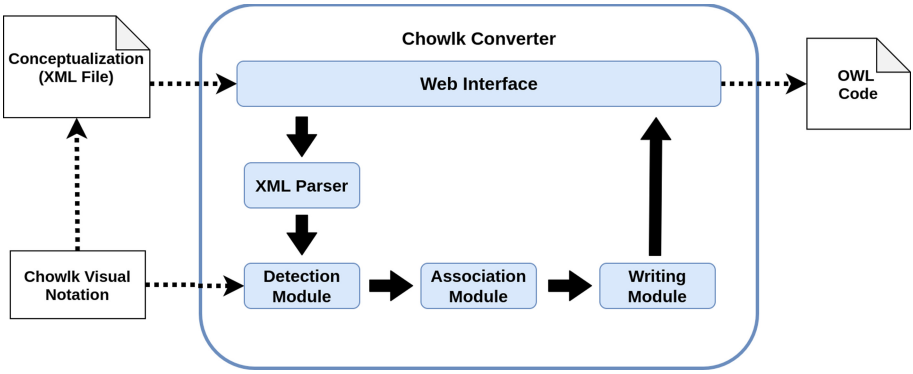

Fig. 3. Chowlk architecture.

4 Demonstration

During the demonstration,[5] some examples will be developed to show the participants how to use Chowlk. First, the conceptualization will be developed using diagrams.net following the Chowlk visual notation. The shapes for the construction of the model will be provided as a diagrams.net library that can be loaded into the diagramming tool. The final conceptualization will be exported an uploaded into the Chowlk web app that will generate the corresponding OWL implementation. The GUI will return the ontology in two formats: RDF/XML and Turtle. As a final step, the ontology will be uploaded into Protégé to demonstrate the syntax correctness of the exported implementation.

5 Conclusions and Future Work

This paper presents the Chowlk converter that facilitates the ontology development process by leveraging the ontology conceptualization diagrams to transform them into OWL. The system is currently being used within the authors research group[6] and also by researchers from other institutions[7]. Next steps involve the

[5] A demo video of Chowlk is available at https://youtu.be/1oF0qDhParA.

[6] See the BIMERR European project https://bimerr.iot.linkeddata.es/ or contributions to the W3C WoT-discovery task https://tinyurl.com/w653me9f.

[7] http://mired.uspceu.es/microrrelatos/.

announcement and promotion of the system to get feedback from users. Additionally, we plan to develop a REST API so that Chowlk can be easily integrated within third-party software such as OnToology [1]. It is also planned to develop a plugin for diagrams.net in order to embed the ontology generation feature into the diagramming tool. Besides, we plan to integrate the prefix.cc[8] service to suggest an URI for prefixes not declared. Including support for including additional metadata, such as comments and labels, is also being explored.

References

1. Alobaid, A., et al.: Automating ontology engineering support activities with OnTology. J. Web Semant. **57** (2019)
2. Barzdins, J., Barzdins, G., Cerans, K., Liepins, R., Sprogis, A.: UML style graphical notation and editor for OWL 2. Perspect. Bus. Inform. Res. **64**, 102–114 (2010). https://doi.org/10.1007/978-3-642-16101-8_9
3. Dudás, M., Lohmann, S., Svátek, V., Pavlov, D.: Ontology visualization methods and tools: a survey of the state of the art. Knowl. Eng. Rev. **33**, e10:1–e10:39 (2018)
4. Weiten, M.: Ontostudio as a ontology engineering environment. Semant. Knowl. Manage. 51–60 (2009). https://doi.org/10.1007/978-3-540-88845-1_5
5. Wiens, V., Lohmann, S., Auer, S.: Webvowl editor: device-independent visual ontology modeling. In: International Semantic Web Conference 2180 (2018)

[8] http://prefix.cc/.