WIDOCO:
A Wizard for Documenting Ontologies

Daniel Garijo[1]

Information Sciences Institute, University of Southern California,
4676 Admiralty Way, Marina del Rey CA, 90292, USA
`dgarijo@isi.edu`

Abstract. In this paper we describe WIDOCO, a *WIzard for DOCumenting Ontologies* that guides users through the documentation process of their vocabularies. Given an RDF vocabulary, WIDOCO detects missing vocabulary metadata and creates a documentation with diagrams, human readable descriptions of the ontology terms and a summary of changes with respect to previous versions of the ontology. The documentation consists on a set of linked enriched HTML pages that can be further extended by end users. WIDOCO is open source and builds on well established Semantic Web tools. So far, WIDOCO has been used to document more than one hundred ontologies in different domains.

1 Introduction

Ontology engineering methodologies acknowledge *reuse of existing vocabularies* as a crucial step when developing a new ontology [11]. Therefore, ontology authors often provide a human-readable documentation of their vocabularies, in order to facilitate their understanding and adoption by other researchers [9].

There are three main aspects related to ontology documentation. The first one is creating a human-readable representation of the content of the ontology: metadata, definition of classes and properties, visualization (e.g., diagrams relating the different concepts) and versioning (explanation of the difference between versions of the ontologies). The second aspect is creating machine-readable annotations of documentation metadata (e.g., provenance, snippets for facilitating vocabulary discovery by search engines) and the third aspect is preparing the documentation files to be accessed as a web resource (doing content negotiation).

Related work has been proposed to facilitate some of these aspects. For example, ontology editors like Protégé [8], have plugins for automatically creating an HTML documentation with the definition of classes and properties.[1] Similarly, approaches like LODE [9] or Parrot [12] provide drag-and-drop services to

---

[1] https://protegewiki.stanford.edu/wiki/OWLDoc

automatically document ontology terms. However, most approaches are typically designed for Semantic Web experts, presenting some of the following issues:

1. *Lack of guidelines and best practices for ontology documentation*: users developing ontologies may not know which are the common terms used to describe the metadata of their ontologies. These metadata are important, because they are used by existing tools to create human readable descriptions of an ontology.
2. *Lack of ontology metadata completion*: Current efforts do not indicate which key information may be missing when documenting an ontology.
3. *Lack of an ecosystem for ontology documentation and customization*: most existing approaches focus on specific aspects of ontology documentation. On the one hand, approaches like LODE [9] generate a human readable description of the classes and properties of a given ontology, but neglect the generation of diagrams. On the other hand, tools like WebVowl [5] create dynamic visualizations of ontologies, but do not deal with the generation of text. Integrating the outcome of these and other tools and customizing them according to user preferences takes time, especially to non programmers.

In this paper we describe WIDOCO, a wizard for documenting ontologies designed to tackle these issues in an automated way. WIDOCO takes as input an annotated RDF vocabulary (e.g., an OWL file with labels and definitions for its concepts) and generates a set of linked HTML pages containing a human readable description of the ontology.[2] WIDOCO *guides users through the steps to be followed* when documenting an ontology, relying on common best practices and *indicating missing metadata that should be included*. WIDOCO also *facilitates customizing the produced documentation*, enabling users to select which aspects they want to include in their document (e.g., sections, diagrams, provenance information, etc.). WIDOCO integrates and extends well established tools, like LODE [9] for term documentation, WebVowl [5] for interactive diagram creation, Bubastis [6] for adding automated change logs between versions and web services like Licensius[3] for completing ontology metadata. In addition, WIDOCO enriches the documentation with snippets discoverable by search engines, prepares content negotiation files for an ontology using W3C best practices, exposes the documentation in multiple languages and exports provenance information of the creation process. We consider that these features make WIDOCO a useful resource for documenting ontologies, and so far we have received positive feedback from the community.

The rest of the paper is structured as follows: a description of the main features of WIDOCO in shown in Section 2. Section 3 discusses the community adoption of the proposed resource, followed by a brief overview of related work describing approaches for ontology documentation in Section 4. Finally, Section 5 points out future directions of work.

---

[2] An overview with examples can be accessed online: https://w3id.org/widoco/gallery
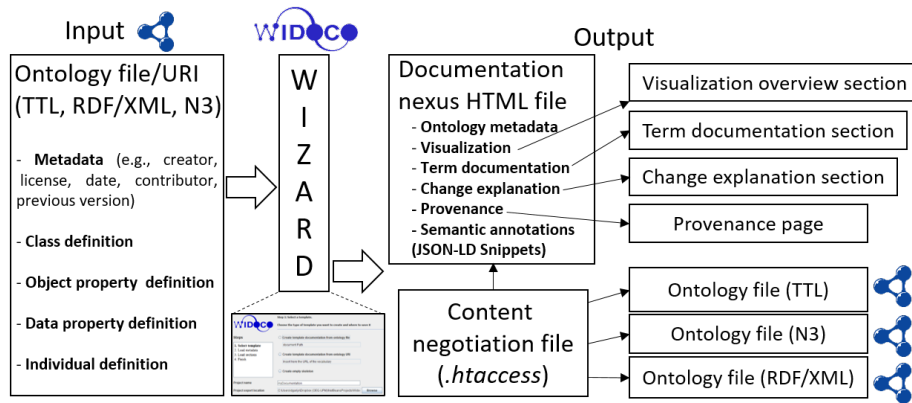[3] http://licensius.com/

Fig. 1. Overview of WIDOCO: Given an ontology file with metadata and definitions for its ontology terms, WIDOCO generates a set of linked HTML files (linked through a nexus file) with definitions of terms, an interactive diagram, an explanation of changes from previous versions and annotations of the ontology document itself (provenance file). In addition, the system generates a file for facilitating documentation publication through content negotiation (*.htaccess*).

2 WIDOCO: A Wizard for Documenting Ontologies

WIDOCO is a standalone java application developed to help users documenting their ontologies. Given an ontology file or URL as input, WIDOCO guides the user through a wizard that generates a customized enriched HTML documentation of the ontology. In this section we describe the main features of WIDOCO in Section 2.1, explaining the steps to be followed by users on WIDOCO's wizard in Section 2.2 and how users may extend the generated documentation on Section 2.3.

2.1 WIDOCO Overview

Figure 1 shows an overview of the different aspects of the documentation process tackled by WIDOCO. Users provide as input to the system a URI or file with metadata annotations (e.g., creators, title, contributors, license, etc.) and definitions for the different ontology terms. After following the steps of the wizard, WIDOCO generates a customized documentation with the most relevant metadata of the ontology. The documentation is composed by a set of linked HTML files, including a main *nexus* file and several *section* files. Each section file describes the content of part of the documentation (e.g., abstract, definition of ontology terms, overview diagrams, etc.) while the nexus file links all the sections together. In addition, WIDOCO generates separate provenance records about the ontology documentation and annotates the nexus file with JSON-LD snippets,[4] which help search engines discovering the ontology metadata. Finally,

---

[4] https://www.w3.org/TR/json-ld/

WIDOCO prepares a content negotiation file and serializations of the ontology to facilitate serving the documentation and ontology in different formats. WIDOCO's features are further described below:

Documenting Relevant Ontology Metadata: WIDOCO uses the OWL API [4] to process and recognize over forty properties for ontology metadata description from common vocabularies and standards. These terms have been grouped in metadata categories and collected in a best practices document,[5] including a rationale of their importance when describing a vocabulary. Some examples of metadata categories are *vocabulary access* (e.g., namespace URI that should be used to dereference the vocabulary), *attribution*(e.g., creators, contributors or publishers), *provenance* (e.g., creation date, sources, previous versions) or *citation* (e.g., DOI of the vocabulary, how to cite it) among others.

Ontology Visualization: We have reused WebVowl [5] to add an interactive diagram to the documentation. The diagram is a useful aid, as it simplifies the visualization in bigger ontologies (automatically filtering loosely connected terms) and helps having an overview of the main properties and classes. WIDOCO transforms the ontology into the format required by WebVowl and saves the result as a separated HTML file.

Ontology Terms Documentation: WIDOCO builds on top of LODE [9], an open source tool designed to generate an HTML file with the definition of classes, properties, data properties and individuals of an ontology, based on the annotations made by the user. WIDOCO extends LODE by expanding the properties a user may use to qualify a term in the ontology. For example, LODE uses *rdfs:label* to represent the names of ontology terms, and *rdfs:comment* to describe their definition in the HTML documentation. However, users may use other similar properties for this purpose, such as *skos:label* and *skos:definition*. WIDOCO includes these properties and recognizes new properties to qualify ontology terms, such as examples of concepts or their rationale for inclusion in the ontology. These properties are also part of the list of best practices for ontology metadata description mentioned above.[5]

Explaining Changes from Previous Versions: Ontologies are likely to evolve, being released in different versions. In these cases, part of the documentation is targeted towards explaining the differences from the last version of the ontology. WIDOCO expands Bubastis [6], a software for capturing differences between classes in ontologies automatically, adding also which are the object properties, data properties and annotations that have changed from version to version.

Provenance: WIDOCO produces a separate page, linked to the nexus file of the documentation, with the statements that refer to the provenance of the documentation itself (sources, previous versions, authors, etc.). The page is captured both in a human readable and machine readable way, following the PROV-O standard [7].
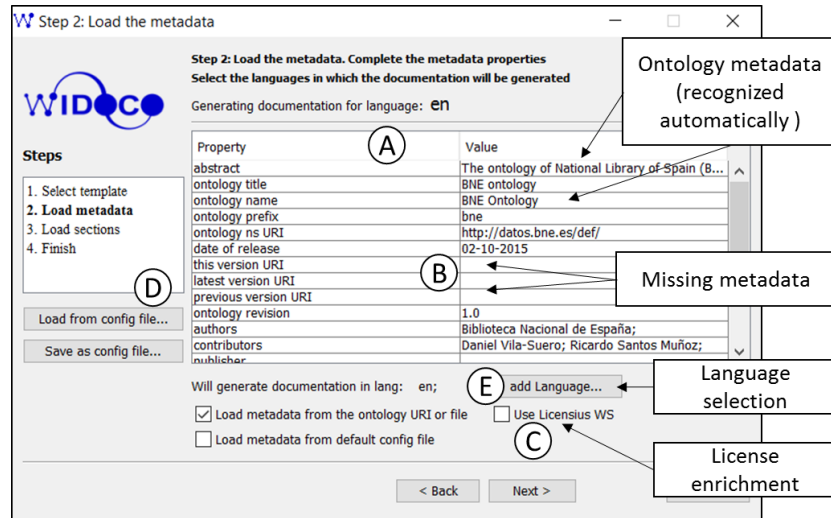
---

[5] https://w3id.org/widoco/bestPractices

Fig. 2. Snapshot of the wizard for gathering ontology metadata: WIDOCO will automatically extract all available metadata from the ontology, showing missing properties that may be included in the documentation.

Semantic annotations: WIDOCO includes JSON-LD snippets on the nexus HTML file with a description of the ontology metadata annotated according to Schema.org.[6] These snippets are useful for search engines to find and explore the metadata of the ontology documentation automatically.

Ontology Serialization and Content Negotiation: WIDOCO automatically creates an *.htaccess* file to access the documentation through its URI once it is published online. We have adopted the W3C best practices for vocabulary publishing on the Web,[7] adapting content negotiation to the vocabulary URI (hash versus slash vocabularies) and preparing the *.htaccess* file to serve (by default) the RDF/XML, TTL and N3 serializations.

2.2 Guiding Users Through the Documentation Process

WIDOCO consists on a wizard that helps users create, customize and enrich a documentation for their ontologies automatically. The main steps are:

1. *Metadata collection:* After selecting an ontology, WIDOCO will load it and fill a table of recommended metadata values, as shown in Figure 2 (A). If a metadata field is not found in the ontology, the corresponding value will appear blank in the table (B), so users can complete it if desired. Users may

---

[6] http://schema.org/
[7] https://www.w3.org/TR/swbp-vocab-pub/

Fig. 3. An overview of the options for customizing the documentation of a vocabulary.

also choose to let WIDOCO look for the license name and URL used in the ontology (C) by querying the Licensius web service and license dataset.[8] Alternatively, ontology metadata may be loaded from a file using a key value pair (D). This option is useful when several ontologies share common metadata but do not have their ontology files annotated.

Finally, this step of the wizard also lets the user select the languages in which the documentation will be generated (E). Metadata loaded in the table is customized to the selected language, which is useful when generating a documentation in multiple languages.

2. *Customization:* In this step a user can select whether to include or not each of the features of WIDOCO in the final documentation. Figure 3 shows an overview of the different options: include default sections (i.e., introduction, overview, description and references) or load them from existing files, include annotation properties or individuals as part of the ontology documentation; export the provenance of the vocabulary as a separated page, create an .htaccess file to handle content negotiation for the ontology, include an interactive diagram and show the changes with respect to the last version of the ontology. In addition, users may choose between two different CSS styles for the documentation.

3. *Ontology browsing:* the final step of the wizard shows up after the documentation has been generated successfully, allowing opening it on a local web browser. This step also enables users to produce evaluation reports of the ontology, facilitating checking whether the ontology has any design flaws or not. WIDOCO produces these reports by using the OOPS! web service [10], which evaluates ontologies against a catalog of pitfalls.

2.3 Extending the generated documentation

WIDOCO organizes its output to be easily modified, allowing users to expand their documentation with additional narratives and diagrams. Each of the sections included in the documentation (e.g, abstract, introduction, description, cross reference, etc.) are separated in individual HTML files, which can be edited

---

[8] http://licensius.com/apidoc/index.html

individually. If new sections or subsections are added to any existing page, all section numbers and table of contents will be updated consistently. Furthermore, WIDOCO supports Markdown,[9] which is generally easier to edit than HTML.

3 Usage and Community Adoption

WIDOCO started as a simple wizard to help non-programmers documenting their ontologies. Due to the uptake and suggestions by users, we have progressively added new functionality (e.g., supporting new types of metadata, creating diagrams, etc.) and expanded the user base. To date, WIDOCO has been used to document more than a hundred ontologies in different domains,[10] ranging from earth sciences to bio-informatics. WIDOCO has also been adopted by tools for supporting ontology engineering such as OnToology [1] and VoCol [3].

Thanks to our interactions with the community, we have developed a benchmark of vocabularies to test and validate WIDOCO.[11] The benchmark aggregates 35 real-world ontologies with different characteristics for generating documentation, such as incomplete metadata, availability in different formats, sizes or languages, unavailability of imported ontologies, etc.

WIDOCO is available in GitHub,[12] where users can download it, open issues or ask for help. WIDOCO is released under an Apache-2.0 license.[13]

4 Related Work

Many approaches have been developed to help create human readable description of ontologies. These often focus on concrete aspects of ontology documentation, including human-readable definitions (e.g, LODE [9], Parrot [12], etc.), differences between versions of ontologies (e.g., Bubastis [6]) or diagram creation (e.g., WebVowl [5]). Other efforts describe end-to-end frameworks for publishing and versioning ontologies, such as Neologism [2], VoCol [3] or OnToology [1]. To our knowledge, WIDOCO is the only approach that includes guidance for users during the documentation process, helping them to complete and enrich their metadata while customizing their final documentation.

5 Conclusions and Future Work

In this paper we have described WIDOCO, a wizard for documenting and customizing ontologies which a) guides users through the documentation process; b) helps identifying missing metadata in the ontology and c) extends and integrates existing work for documenting ontology terms, diagram visualizations and

---

[9] https://learn.getgrav.org/content/markdown
[10] https://w3id.org/widoco/usage
[11] https://w3id.org/widoco/benchmark
[12] https://github.com/dgarijo/Widoco/
[13] https://www.apache.org/licenses/LICENSE-2.0

ontology revisions. WIDOCO has been used to document more than one hundred ontologies across different domains, and has been adopted by other existing efforts like OnToology and VoCol.

WIDOCO is an ongoing effort, and we are open to suggestions proposed by the community. In fact, we have already addressed several issues raised by adopters of the tool. Our future work aims to facilitate filtering and enrichment of ontology terms to be included in the documentation with external metadata.

Acknowledgements

References

1. Alobaid, A., Garijo, D., Poveda-Villalón, M., Pérez, I.S., Corcho, O.: Ontology, a tool for collaborative development of ontologies. In: ICBO (2015)
2. Basca, C., Corlosquet, S., Cyganiak, R., Fernndez, S., Sch, T.: Neologism: Easy Vocabulary Publishing (2012)
3. 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: Proceedings of the 20th Knowledge Engineering and Knowledge Management - EKAW 2016, Bologna, Italy. pp. 303–319 (2016)
4. Horridge, M., Bechhofer, S.: The OWL API: A java api for owl ontologies. Semant. web 2(1), 11–21 (2011)
5. Lohmann, S., Link, V., Marbach, E., Negru, S.: WebVOWL: Web-based Visualization of Ontologies, pp. 154–158. Springer International Publishing, Cham (2015)
6. Malone, J., Holloway, E., Adamusiak, T., Kapushesky, M., Zheng, J., Kolesnikov, N., Zhukova, A., Brazma, A., Parkinson, H.: Modeling sample variables with an experimental factor ontology. Bioinformatics 26(8), 1112 (2010)
7. McGuinness, D., Lebo, T., Sahoo, S., Belhajjame, K., Cheney, J., Corsar, D., Garijo, D., Soiland-Reyes, S., Zednick, S., Zhao, J.: PROV-o: The PROV ontology. W3C recommendation, W3C (2013)
8. Musen, M.A.: The Protégé project: A look back and a look forward. AI Matters 1(4), 4–12 (2015), http://doi.acm.org/10.1145/2757001.2757003
9. Peroni, S., Shotton, D., Vitali, F.: The Live OWL Documentation Environment: A Tool for the Automatic Generation of Ontology Documentation, pp. 398–412. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
10. Poveda-Villalón, M., Suárez-Figueroa, M.C., Gómez-Pérez, A.: OOPS! (OntOlogy Pitfall Scanner!): An on-line tool for ontology evaluation,. International Journal on Semantic Web and Information Systems (IJSWIS) 10(2), 7–3–4 (2014)
11. Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernández-López, M.: The NeOn Methodology framework: A scenario-based methodology for ontology development. Applied Ontology 10(2), 107–145 (2015)
12. Tejo-Alonso, C., Berrueta, D., Polo, L., Fernández, S.: Metadata for Web Ontologies and Rules: Current Practices and Perspectives, pp. 56–67. Springer Berlin Heidelberg, Berlin, Heidelberg (2011)