OnToology, a tool for collaborative development of ontologies
Ahmad Alobaid 1, Daniel Garijo 1*, María Poveda-Villalón 1, Idafen Santana-Perez 1 and Oscar Corcho 1

1Ontology Engineering Group, Universidad Politécnica de Madrid, Spain

ABSTRACT

In this demo we present OnToology, a tool for developing ontologies collaboratively using Github. OnToology addresses several steps of the ontology development lifecycle, including documentation, representation, evaluation and publication in a non-intrusive way.

1 INTRODUCTION

The rise of collaborative technologies has sped up the development of software on the last decade. When working as a team, it is common to use repositories for software development, open discussions and having a ticketing system that warns and keeps track of the main issues to be solved.

This paradigm is slowly moving towards other domains, like ontology development. Ontologies, like software, require a set of requirements to be established and are usually discussed in a group before agreeing on a design decision. Therefore, they benefit heavily from the ticketing system, versioning and decision tracking that collaborative environments offer. However, this is often not enough, as ontologies need to be further documented and published online.

Although some tools cover part of these activities e.g. documentation and evaluation, there are no tools that integrate them with a collaborative environment.

In this demo we present OnToology, a tool for documenting, evaluating, presenting and publishing ontologies developed collaboratively. Section 2 describes the requirements for developing ontologies collaboratively, while Section 3 describes our approach. Finally Section 4 describes related work and Section 5 introduces our efforts for improving the tool.

2 ONTOLOGY DEVELOPMENT LIFE CYCLE

Typically, the ontology development process can be divided in several independent activities:

- Ontology requirements: before committing to implement an ontology, it is advised to write a set of competency questions (CQs) in an ontology requirements specification document as mentioned in NeOn methodology (Suárez-Figueroa et al., 2012), which will be used to test the ontology.
- Ontology Implementation: once agreed on the ontology requirements, one can use an ontology editor such as NeOn-toolkit or Protégé to design the properties and classes of the proposed ontology.
- Ontology evaluation: the resultant ontology can be evaluated in two different ways: by checking whether the requirements (i.e., CQs) are answered properly and by checking whether the ontology follows design patterns and well-established practices for its implementation or not.
- Ontology documentation: an ontology is unlikely to be reused unless it is documented properly with examples. This phase focuses on producing a human-readable documentation that allows users understand the OWL or RDF files produced during the implementation phase.
- Ontology publication: in this phase the ontology has been agreed on and its ready for release. As the aim of the vocabularies and ontologies is normally to share the model for its reuse, the ontology is released with its documentation.

Figure 1 presents an overview of the different phases of the ontology development lifecycle. As shown in the figure, this cycle benefits from a collaborative versioning environment that tracks the changes made to the ontology, requirements, documentation and diagrams; and keeps a log of the group discussions and decisions made.

3 COLLABORATIVE CREATION OF ONTOLOGIES WITH ONTOOLOGY

OnToology is a web-based tool designed to automate part of the ontology development process in collaborative environments. In particular, OnToology is designed to work with Github, one of the most common environments for software development. After registering a repository to OnToology, developers just push their changes to Github and the tool will produce the documentation (with several proposals for diagram representation), evaluation and publication of the ontology in the user’s repository. The phases covered by OnToology are further described below:

*To whom correspondence should be addressed: dgarijo@fi.upm.es

1 http://purl.org/net/OnToology
2 http://neon-toolkit.org/
3 http://protege.stanford.edu/

4 https://github.com/OnToology/OnToology
5 http://www.github.com/
The web interface of the tool are the four external systems as can be seen in Figure 2. The two main parts OnToology is composed of two main parts and is integrated with

3.1 OnToology Architecture

OnToology allows developers to customize which of the integrated systems and submitting them in a pull request to the repository, where the maintainer can review the changes and merge them later on. The integrator also opens an issue in Github including the summary of the pitfalls generated by OOPS! with a link to a full extended explanation.

4 RELATED WORK

Neologism (Cosmin Basca et al., 2008) is a web-based editor for vocabulary editing and publishing. Unlike OnToology, it lacks revision tracking and ontology best practice evaluation. VoCol (Niklas Petersen et al.) is another tool integrated with Github for a collaborative approach. The tool suffers from strictness and lack of freedom on the generated output, and OnToology provides full control over the generated output. Finally, WebProtégé is a web-based ontology editor for the collaborative development of ontologies. WebProtégé is focused on the implementation of the ontologies, which can only be edited online. In OnToology, the creation can be done offline as well. WebProtégé also lacks the evaluation of ontologies like the one provided by OOPS!, which is also used by OnToology.

5 CONCLUSIONS AND FUTURE WORK

In this demo we have introduced OnToology, a tool for improving the ontology development lifecycle in collaborative environments. OnToology helps documenting, depicting, evaluating and publishing. As future work, we are currently working on addressing automatic deployment and archival of the ontology releases under demand.

ACKNOWLEDGEMENTS

This research has been funded by the project “4V: Volumen, Velocidad, Variedad y Validez en la Gestión Innovadora de Datos” (TIN2013-46238-C4-2-R).

REFERENCES

