Improved explanations in the Protégé OWL ontology editor

Cilliers Pretorius[1] prtpie003@myuct.ac.za and
Thomas Meyer[1,2] tmeyer@cs.uct.ac.za

[1] University of Cape Town, South Africa
[2] Centre for Artificial Intelligence Research

A logic-based reasoning system is a software system that generates conclusions that are consistent with a knowledge base (KB) or ontology. However, because the steps taken to generate a conclusion are usually hidden from the user, it cannot be guaranteed that the user accepts and acts upon the conclusion[3]. This leads to systems that provide explanations and justifications as a key part of the system's design[5]. Novice and expert users greatly benefit from explanations[1].

Protégé is an ontology development tool that allows users to create ontologies according to the Web Ontology Language (OWL). OWL is a Description Logic (DL) that allows for precise and unambiguous definitions, allowing for reasoners to infer conclusions[2]. The Explanation Workbench plugin developed by Horridge et. al. [2] is bundled with Protégé. The Explanation Workbench allows users to generate explanations for an inferred conclusion using the same reasoner. It generates justifications and outputs the axioms contained in the justification. However, these axioms can be difficult to understand if the user did not create the ontology themselves. Despite the use of keywords, both expert and novice users might struggle to understand the explanations unless they have enough knowledge of description logics. This is unlikely given that many ontologies are created for specific knowledge bases not related to description logics[4].

This paper attempts to provide more readable and more convincing explanations, built on the Explanation Workbench's explanations. The current tool's explanations have some natural language as a side-benefit of the Manchester Syntax used by Protégé. It does not help users who are not familiar with description logics or ontologies in general. With the overarching goal of improved readability and more effective explanations, two methods are considered. One method is to allow the creator of an ontology to define an explanation for an axiom, which would be displayed as the explanation for that particular axiom. The second method is to expand the keywords that are used in the axiom to use more natural language. Thus, the explanations should be more readable and easily understood by users, even if they are unfamiliar with description logics.

We define an annotation property exp:Explanation with the typing triple
```
<owl:AnnotationProperty rdf:about="exp:Explanation"/>
```

If a formal definition or AnnotationProperty is added to the OWL standards, then the definition used in this paper will be changed to reflect the OWL standards. Explanations are still invoked when the user clicks on the "Explain inference" button that appears next to each axiom. Checkboxes are added to allow the user to decide if annotated explanations should be displayed or keywords expanded. Users are allowed to have the checkboxes checked in any combination. If neither is checked, the output is exactly as the original Explanation Workbench would produce.

Keyword expansion is implemented as a single function. The function iterates through the axiom and replaces every keyword with the equivalent expansion. It returns the expanded axiom to the renderer to be displayed on the screen. If the checkbox for explanatory annotations is checked, the method checks if the axiom has any annotations attached to it. If there is at least one annotation, it iterates over all annotations and checks if any annotation has the annotation property "exp:Explanation". If this is true, the renderer will display the axiom (in its original Manchester Syntax form) and append the explanatory annotation after it. Note that this functionality requires the ontology creator to have defined the explanatory annotations beforehand.

The source code of this extension of the Explanation Workbench can be found and freely downloaded from the paper's GitHub repository[3]. It will require Protégé to be of use, which can be downloaded from `protege.stanford.edu`. The More Readable Extension to the Explanation Workbench (MRE) is of significant value to novice users since it allows them to more easily understand description logics and the knowledge held by the ontology. It does not prohibit the more rigid and formal notation, thereby benefiting expert users who might prefer the more formal notation. It can allow all users to receive terminological knowledge regarding the ontology with significantly greater ease and is therefore of great benefit to users wanting to familiarise themselves with a new ontology.

The next step in this research would be to create a tool to automatically generate the explanatory annotations for axioms. This research could draw inspiration from OWL Simplified English (OWLSE) and Attempto Controlled English (ACE) to formulate the actual annotations. It might also look at Horridge's work on justifications to determine what axioms should get annotations if annotations are prioritised to the most important axioms. Further research can also be done to integrate the keyword expansion and the OWLSE and ACE syntaxes. This can be integrated with user testing to evaluate the various attempts at natural language expressions for OWL and the associated explanations.

References

1. Dhaliwal, J.S.: An experimental investigation of the use of explanations provided by knowledge-based systems. Ph.D. thesis, University of British Columbia (1993)

[3] `https://github.com/Pietersielie/Explanation-Workbench-More-Readable-Extension`

2. Horridge, M., Parsia, B., Sattler, U.: The owl explanation workbench: A toolkit for working with justifications for entailments in owl ontologies (2009)
3. McGuinness, D.L., Patel-Schneider, P.F.: Usability issues in knowledge representation systems. In: AAAI/IAAI (1998)
4. Musen, M.A., et al.: The protégé project: a look back and a look forward. AI matters **1**(4), 4 (2015)
5. Teach, R.L., Shortliffe, E.H.: An analysis of physician attitudes regarding computer-based clinical consultation systems. Computers and Biomedical Research **14**(6), 542–558 (1981)