RepOSE-CTab - A Protégé Plugin for Completing Ontologies

Zlatan Dragisic[1], Ying Li[1], and Patrick Lambrix[1,2,3]

[1] Department of Computer and Information Science, Linköping University, Sweden
[2] The Swedish e-Science Research Centre, Linköping University, Sweden
[3] Department of Building Engineering, Energy Systems and Sustainability Science, University of Gävle, Sweden
patrick.lambrix@liu.se

Abstract. As the quality of ontologies plays an important role in supporting semantically-enabled applications, defining concepts as well as their relations correctly and completely is crucial when developing an ontology. In this paper we introduce a Protégé plugin for extending ontologies, which guides Protégé users through the addition of new concepts as well as their instances and axioms in which they participate in a semi-automatic way. Furthermore, the tool suggests additional subsumption axioms that a user can validate and, if appropriate, add to the ontology to make the ontology more complete.[1]

1 Introduction

Developing ontologies is not an easy task and when ontology developers add new concepts to an ontology they may not be as complete as they could be. When ontologies are not complete, it has an effect on semantically-enabled applications using these ontologies as valid conclusions may be missed [2]. For instance, in [3] an example is given where a missing subsumption relation leads to the fact that ontology-based search in PubMed misses more than half of the query results.

To improve the completeness of ontologies during development, we present a plugin for Protégé [4] that supports a user when adding new concepts to an ontology. The tool requests information to be filled out in terms of concept names, instances and super-concepts and shows the consequences, e.g., in terms of the subsumption hierarchy. Compared to the traditional way of adding concepts and instances in ontology editors, this plugin introduces an extra step, called completion, based on earlier work on the RepOSE tool [3]. In this step the plugin suggests additional subsumption axioms that the user can validate and, if appropriate, add to the ontology to make the ontology more complete. Although it may seem that when knowledge engineers and domain experts add knowledge in the form of subsumption axioms, they add all knowledge they know, this may not always be the case. For instance, [3] describes an experiment with the ontologies

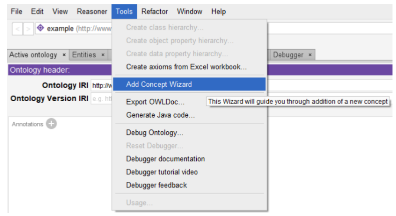


Fig. 1. The RepOSE-CTab as *Add Concept Wizard* in the tools menu.

of the Anatomy track of the Ontology Alignment Evaluation Initiative [1] where 94 subsumption axioms were initially added to the Adult Mouse Anatomy ontology, but using the RepOSE completion tool 47 additional subsumption axioms that could not be derived from the ontology and the 94 axioms, were found. For the NCI Human Anatomy ontology, 58 subsumption axioms were added initially and 10 additional axioms were found using completion.

In terms of the design space dimensions in [6] the **purpose** of the tool is to support a user when adding new concepts to an ontology. It focuses on $\mathcal{EL}$ ontologies, although for ontologies defined in more expressive knowledge representation languages, the tool can be used for the taxonomic part, i.e., subsumption relations between the named concepts representing ontological concepts. The main purpose of the tool is *creating and managing* as defined in [6] as it is essentially part of an ontology development tool. However, there is also a minor aspect of *learning and understanding* as a user needs to understand the concepts and their contexts to be able to make informed decisions about what axioms to add. The **users** of the tool are ontology developers. This usually requires expertise in ontology engineering as well as in the domain that is being modeled in the ontology.

In section 2 we describe our tool and in section 3 we show the use of the tool on a small example.

2 RepOSE-CTab

Our tool is a plugin for Protégé 5.0 (and tested for Protégé 5.0 and 5.1). It is Java based and uses the OWL API[2], the Protégé package for accessing and handling OWL ontologies and the Protégé editor. It is installed by copying a jar file (AddConcept) into the plugin directory of the existing main Protégé folder. Then upon a new start, Protégé will find the tool and create a new tab for it. The tool is then found in the tools menu as *Add Concept Wizard* (Figure 1).

The general process of adding a new concept as well as their subsumption relations and instances with the assistance of this Add Concept Wizard is as follows. In a first step, the user types the new concept's name and selects its super-concepts from the concepts list. When selecting the super-concepts, the

[2] http://owlapi.sourceforge.net/

tool provides information such as super-concepts, sub-concepts and disjoint concepts. This allows the user to get an overview of the possible context of the concept and may be helpful in deciding, e.g., to add disjointness or additional subsumption axioms. Upon providing the super-concepts of the new concept, the corresponding subsumption axioms are generated. Most of this functionality is common in ontology editors.

Furthermore, in a second step a unique feature not found in current ontology editors is implemented. The tool generates suggestions for additional subsumption axioms to add to the ontology. This step is called completing in [2] and is an abductive reasoning problem. The algorithm for generating these suggestions is explained, e.g., in [3]. For the implementation in this plugin the tool generates source and target sets for each axiom $\alpha \sqsubseteq \beta$. The source set contains the super-concepts of α and the target set contains the sub-concepts of β. We know then that adding any $\alpha_s \sqsubseteq \beta_t$ with α_s in the source set and β_t in the target set, to the ontology will make $\alpha \sqsubseteq \beta$ logically derivable from the ontology. However, the set of logical solutions may contain solutions that are not correct according to the domain. It is, therefore, important that a domain expert validates these $\alpha_s \sqsubseteq \beta_t$ to make sure that no wrong knowledge is added to the ontology.[3]

The source and target sets for each axiom are shown to the user who can decide on using such a new axiom by clicking on a concept in the source set and a concept in the target set and validate it or by validating it from a list. Further, when such new axiom is added, new knowledge is added to the ontology and there may be new completions possible. Therefore, this step is iterated.

In [2] a relation *more complete* is defined between different repairs, which, in this case, refers to different sets of axioms that are used to add knowledge to the ontology.[4] A repair R1 is more complete than another repair R2 if every correct statement according to the domain that can be derived from the ontology with the addition of the axioms in R2, can also be derived from the ontology with the addition of the axioms in R1, and there is correct statement according to the domain that can be derived from the ontology with the addition of the axioms in R1, but that cannot be derived from the ontology with the addition of the axioms in R2. Essentially, the new ontology based on R1 contains more correct knowledge than the new ontology based on R2 and thus is 'more complete'. Based on this definition, we can then say that, if an axiom is added in the second step of the process, then the new ontology after the second step is more complete than the ontology after the first step of the process and more complete than the ontology before we started adding a new concept.

[3] A similar issue appears in a dual problem of completing, i.e., debugging. While in completing logical solutions may add axioms that are not correct according to the domain, in debugging logical solutions may remove correct knowledge from the ontologies or mappings [5]. Therefore, in both cases a domain expert needs to validate the logical solutions.

[4] The definition in [2] takes also into account removal of axioms, but we use a simplified version only dealing with addition of axioms as the plugin only deals with adding concepts.

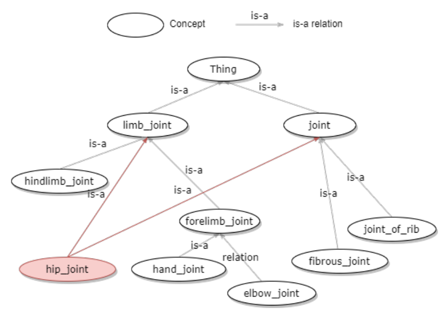


Fig. 2. A piece of the Adult Mouse Anatomy ontology.

Finally, the user can then also add existing instances for the new concept. As a help the user can obtain the instances of super-concepts or search for instances. Once all the relevant information is provided, all axioms and instances are listed in the final step. After that, the new concept as well as their instances and relevant axioms are added into the ontology.

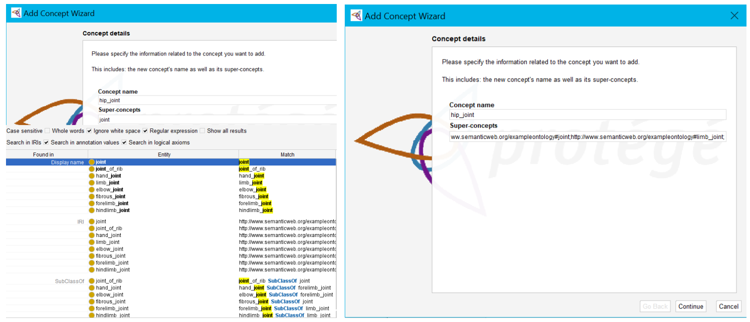
3 Example run

In the following we run through a scenario showing how the plugin guides a user when adding concepts and subsumption relations related to that concept to an ontology. We use an example inspired by the Adult Mouse Anatomy (`http://www.informatics.jax.org/vocab/gxd/ma_ontology/`, part of the Gene Expression Database) that is used in the Anatomy and Interactive tracks of the Ontology Alignment Evaluation Initiative [1].

Fig. 2 shows a small piece of the ontology and it is this piece that we will extend. We assume that the user wants to add the new concept *hip_joint* and the user knows that the existing concepts *joint* and *limb_joint* are super-concepts of the new concept. The user will use the plugin to add this knowledge to the ontology.

After loading the target ontology into Protégé and starting the built-in reasoner, the user can click on *Add Concept Wizard* in the tools menu to start the guide to concept and axiom addition. First, the user is asked to enter the concept name and select super-concepts from the list of concepts in the ontology. As an additional help, when typing the name of a super-concept, information about these concepts, such as disjoint concepts, super-concepts and sub-concepts, is shown (Figure 3a). In this example, the user uses *hip_joint* as the concept name, and *joint* and *limb_joint* as its super-concepts (Figure 3b). This means that the axioms $hip\_joint \sqsubseteq joint$ and $hip\_joint \sqsubseteq limb\_joint$ should be derivable from the extended ontology.

In the next step completion is performed. The tool proposes other axioms to add to the ontology that make $hip\_joint \sqsubseteq joint$ and $hip\_joint \sqsubseteq limb\_joint$ derivable, and would add more new knowledge to the ontology. As explained



(a) (b)

Fig. 3. Concept name and super-concepts.

earlier, it does so by creating source and target sets for these axioms and asks the user to validate axioms $\alpha_s \sqsubseteq \beta_t$ with α_s in the source set and β_t in the target set. In our example, for $hip\_joint \sqsubseteq limb\_joint$, the source set is $\{hip\_joint\}$ and the target set is $\{limb\_joint$, $hindlimb\_joint$, $forelimb\_joint$, $hand\_joint$, $elbow\_joint\}$ (Fig. 4a). The user recognizes that $hip\_joint \sqsubseteq hindlimb\_joint$ and chooses and validates this relation. When adding the $hip\_joint \sqsubseteq hindlimb\_joint$ into the ontology, $hip\_joint \sqsubseteq limb\_joint$ can be derived from the ontology and thus is redundant. The user can remove this redundant axiom by clicking the validate relation button, uncheck the relation $hip\_joint \sqsubseteq limb\_joint$ in the validated relation list and then click the validate button to finish.

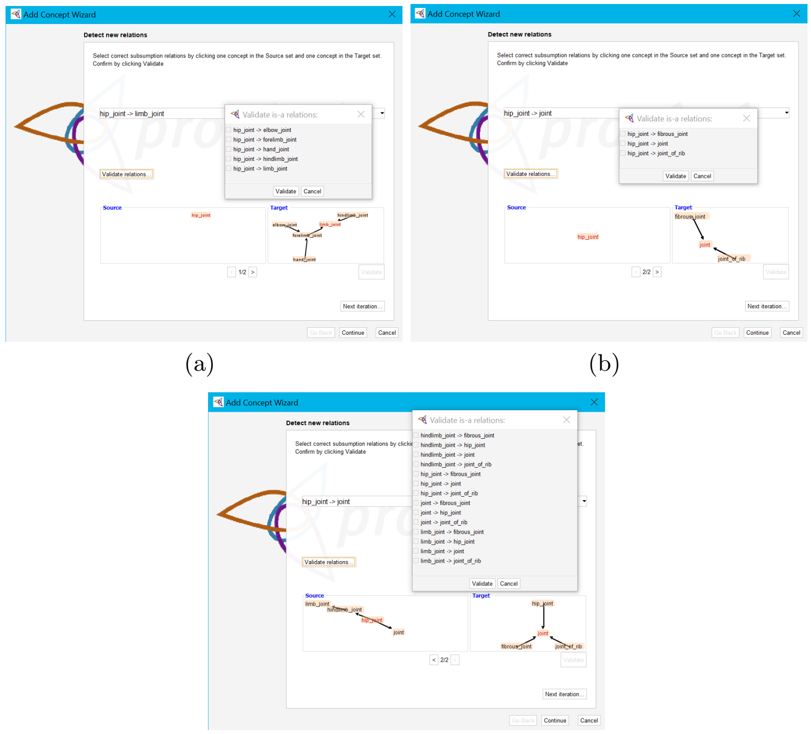
Similarly, the tool computes the source and target sets for $hip\_joint \sqsubseteq joint$. The source set is $\{hip\_joint\}$ and the target set is $\{joint$, $fibrous\_joint$, $joint\_of\_rib\}$ (Fig. 4b). The additional suggestions from the tool are not retained. After this first iteration we thus have the original axiom to add $hip\_joint \sqsubseteq joint$ and the new $hip\_joint \sqsubseteq hindlimb\_joint$. As we have added a new axiom we start a new iteration.

Fig. 4c shows the new source ($\{hip\_joint$, $limb\_joint$, $hindlimb\_joint$,$joint\}$) and target ($\{joint$, $fibrous\_joint$, $joint\_of\_rib$, $hip\_joint\}$) sets of $hip\_joint \sqsubseteq joint$. The user may note that $limb\_joint \sqsubseteq joint$ and choose and validate this relation. Then, the iteration is ended by the user. As before, the user can uncheck redundant relations in the validated relation list.

The plugin also helps adding instances to the ontology, but in this example we do not have any.

In the final step, all axioms that we validated are shown to the user again (Fig. 5a). Upon finishing, the new concept hip_joint and the remaining axioms $hip\_joint \sqsubseteq hindlimb\_joint$ and $limb\_joint \sqsubseteq joint$ are added to the ontology (original ontology in Fig. 5b and new ontology in Fig. 5c).

The new ontology contains the new concept and the axioms based on the super-concepts are derivable. However, we also added additional knowledge during the completion step. For instance, $elbow\_joint \sqsubseteq joint$ is now also derivable



(a) (b)

(c)

Fig. 4. Source and Target sets for different axioms and iterations.

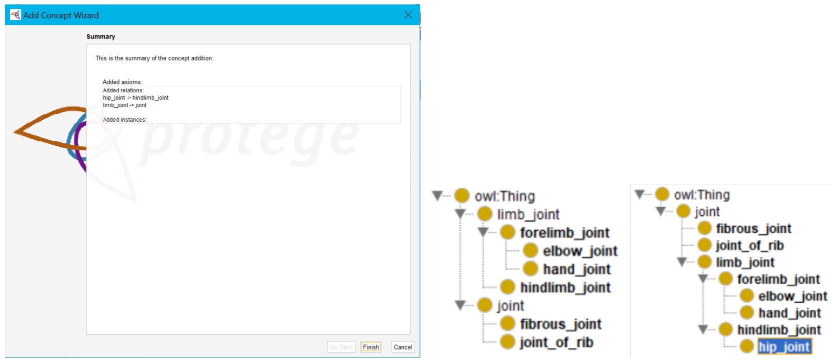
from the ontology. The extension of the ontology using the axioms in the completing step is therefore more complete than the extension with the axioms based on the original super-concepts assigned by the user in the first step.

4 Conclusion

In this paper, we introduced a plugin for the ontology editor Protégé, which guides the Protégé user through extending an ontology by adding new concepts as well as their relations and instances in a semi-automatic way. The use of the plugin leads to more complete ways to extend the ontology than just adding the concepts.

5 Availability

The plugin is available at:
`https://www.ida.liu.se/~patla00/research/RepOSE/downloads.html`.



(a) The validated subsumption axioms. (b) (c)

Fig. 5. Adding subsumption axioms.

Acknowledgements. This work has been financially supported by the Swedish e-Science Research Centre (SeRC), the Swedish National Graduate School in Computer Science (CUGS), the Swedish Research Council (Vetenskapsrådet, dnr 2018-04147), the Horizon 2020 project SPIRIT (grant agreement No 786993), and the Swedish Agency for Economic and Regional Growth (Tillväxtverket, 20201438).

References

1. Dragisic, Z., Ivanova, V., Li, H., Lambrix, P.: Experiences from the anatomy track in the ontology alignment evaluation initiative. Journal of Biomedical Semantics **8**(1), 56:1–56:28 (2017). https://doi.org/10.1186/s13326-017-0166-5
2. Lambrix, P.: Completing and debugging ontologies: state of the art and challenges (2020), arXiv:1908.03171
3. Lambrix, P., Wei-Kleiner, F., Dragisic, Z.: Completing the is-a structure in light-weight ontologies. Journal of Biomedical Semantics **6**, 12:1–26 (2015). https://doi.org/10.1186/s13326-015-0002-8
4. Musen, M.A.: The protégé project: a look back and a look forward. AI Matters **1**(4), 4–12 (2015). https://doi.org/10.1145/2757001.2757003, `https://doi.org/10.1145/2757001.2757003`
5. Pesquita, C., Faria, D., Santos, E., Couto, F.M.: To repair or not to repair: reconciling correctness and coherence in ontology reference alignments. In: Shvaiko, P., Euzenat, J., Srinivas, K., Mao, M., Jiménez-Ruiz, E. (eds.) Proceedings of the 8th International Workshop on Ontology Matching. CEUR Workshop Proceedings, vol. 1111, pp. 13–24 (2013), `http://ceur-ws.org/Vol-1111/om2013_Tpaper2.pdf`
6. Pesquita, C., Ivanova, V., Lohmann, S., Lambrix, P.: A framework to conduct and report on empirical user studies in semantic web contexts. In: Faron-Zucker, C., Ghidini, C., Napoli, A., Toussaint, Y. (eds.) Knowledge Engineering and Knowledge Management - 21st International Conference, EKAW 2018, Nancy, France, November 12-16, 2018, Proceedings. LNCS, vol. 11313, pp. 567–583 (2018). https://doi.org/10.1007/978-3-030-03667-6_36