OWLAx: A Protégé Plugin to Support Ontology Axiomatization through Diagramming

Md. Kamruzzaman Sarker[1], Adila A. Krisnadhi[1,2], and Pascal Hitzler[1]

[1] Wright State University, OH, USA
[2] Universitas Indonesia, Depok, Indonesia

Abstract. Once the conceptual overview, in terms of a somewhat informal class diagram, has been designed in the course of engineering an ontology, the process of adding many of the appropriate logical axioms is mostly a routine task. We provide a Protégé[3] plugin which supports this task, together with a visual user interface, based on established methods for ontology design pattern modeling.

1 Motivation

When modeling with domain experts, particularly when they do not possess intimate knowledge about ontology engineering, it is in our experience best to use a visual approach to first design a conceptual overview of ontology modules (or corresponding content ontology design patterns), in the form of class diagrams [4]. We have found it most effective to use non-electronic means for this, such as whiteboards and flipcharts, as they readily support a natural flow of discussion without assuming any prior knowledge of particular software tools.

The ontology engineers in the modeling team will of course keep track of the precise meaning of each part of the diagram, so that they can convert their insights into exact specifications, i.e., axioms for the ontology. This conversion can then, based on the class diagram and the discussions during the modeling sessions, in our experience mostly be done by the ontology engineers without a lot of required further interaction with the domain experts. For documentation (or publication) purposes, the class diagram will usually also be redrawn using appropriate software tools.

In our experience, based on the class diagram and the discussions with domain experts during its design, it is mostly a routine, albeit somewhat tedious, task to write down appropriate axioms for an ontology module in an ontology editing tool. Most axioms in fact arise out of a systematic exploration of the class diagram. In order to simplify this part of the work, we have cast this systematic exploration into a Protégé plugin, which we describe herein. Of course, some axioms – arguably the more interesting and more critical ones – will not come up as candidates during our systematic exploration, and so will have to be added manually. Nevertheless, our plugin is helpful in making the task of adding many routine axioms much quicker and less error prone.

[3] http://protege.stanford.edu/

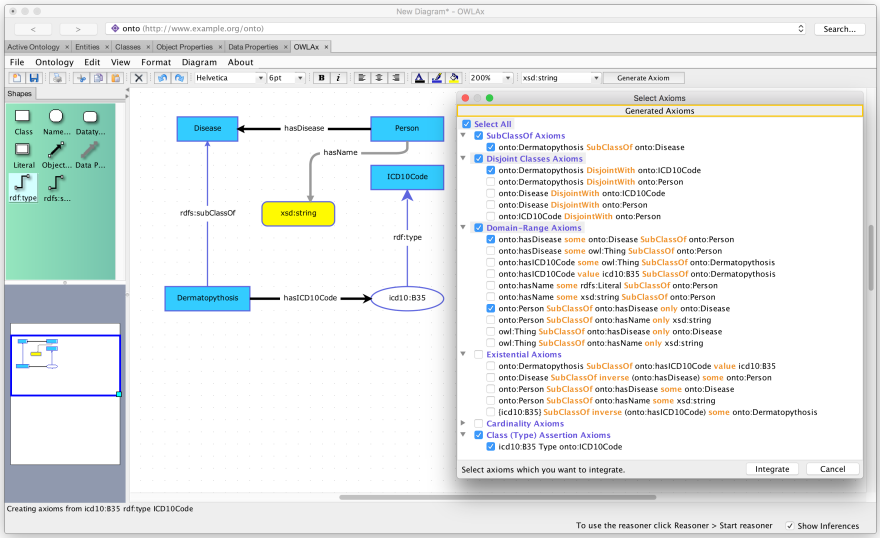

Fig. 1. OWLAx UI when "Generate Axioms" command is executed.

More information about the plugin is located at `http://daselab.org/content/ontology-axiomatization-support`.

2 OWLAx: Description and Main Functionalities

The plugin provides an interface for drawing a *class diagram*, and a command (accessible through an item in the menu or a button in the toolbar) to generate axioms from the given diagram, which are to be *added* into the currently active ontology. As seen in Fig. 1, the class diagram itself is composed of *nodes* and *edges*. A node in the class diagram represents either a *class*, *datatype*, *individual name*, or *literal*. Meanwhile, an edge represents either an *object property*, *data property*, the *typing* relation (i.e., rdf:type), or the *subclass* relation (i.e., rdfs:subClassOf). A pallette on the left side of the interface provides the user with those nodes and edges, which can be dragged and dropped onto the canvas.

In the following, $X \xrightarrow{P} Y$ means that there is a directed edge P from a node X to another node Y in the class diagram. Also, A and B denote class names, M a datatype, c a named individual, ℓ a literal, R an object property, and Q a data property. The plugin enforces the class diagram to contain at least one node, and if there is an edge, it only allows the following node-edge-node configurations: $A \xrightarrow{R} B$, $A \xrightarrow{R} c$, $A \xrightarrow{Q} M$, $A \xrightarrow{Q} \ell$, $c \xrightarrow{\text{rdf:type}} A$, and $A \xrightarrow{\text{rdfs:subClassOf}} B$. We do not aim to represent all possible relationships between components of the class diagram above because in our experience when modeling, the class diagram is usually considered informal, and the aforementioned node-edge-node

configurations are those typically used to describe a conceptual overview when we conduct modeling [4]. In fact, we do not intend to represent all possible OWL 2 constructs in the diagram unlike, e.g., Graphol [2], Graffoo [3], or Ontodia [5].

From the class diagram, a user can generate several types of candidate axioms based on the relationships depicted in the class diagram. They are only *candidates* since the class diagram is informal; each candidate axiom captures one way to read a relationship, and the actual intent should typically be inquired to the domain experts while conducting the modeling. Note that from one type relationship, more than one actual intents need to be formalized, i.e., the candidate axioms are not mutually exclusive. On the other hand, the list of candidate axioms is not exhaustive to keep it sufficiently simple: there are obviously axioms that will not be directly generated from the class diagram, especially if it is too complex. For such axioms, one has to simply directly input them in Protégé.

The plugin facilitates the creation of candidate axioms through a dialog box (accessible through "Generate Axiom" command from the menu or toolbar) that contains a checkbox of the candidate axioms presented in the Manchester syntax. After clicking "Integrate", the plugin will integrate the axioms with a check-mark to the ontology. We explain some of the candidate axioms below, though we use mainly description logic notation [1].

Every $c \xrightarrow{\mathsf{rdf:type}} A$ leads to a *class assertion* $A(c)$, and $A \xrightarrow{\mathsf{rdfs:subClassOf}} B$ to a *subclass axiom* $A \sqsubseteq B$. Next, for every $A \xrightarrow{R} B$, the plugin generates several types of candidate axioms. First, it generates *(unscoped) domain restriction* $\exists R.\top \sqsubseteq A$ — equivalent to R rdfs:domain A — and *scoped domain restriction* $\exists R.B \sqsubseteq A$. The former would be later integrated if the domain experts involved in modeling agrees that for every pair of instances x, y, if $x\ R\ y$ holds, then x belongs to A (regardless whether or not y belongs to B), while the latter is chosen if the domain experts agrees that if $x\ R\ y$ holds and y is known to belong to B, then x belongs to A. Such agreements will be solicited from domain experts involved in the modeling for every candidate axiom. Besides domain restrictions, the plugin also generates *scoped* and *unscoped range restrictions* $A \sqsubseteq \forall R.B$, $\top \sqsubseteq \forall R.B$ — equivalent to R rdfs:range B; several *existential axioms*, e.g., $A \sqsubseteq \exists R.B$, etc.; and several *functionality restrictions*, e.g., $A \sqsubseteq (\leq 1\ R.B)$, etc.

Similar types of candidate axioms are generated for every $A \xrightarrow{Q} M$, $A \xrightarrow{R} c$, and $A \xrightarrow{Q} \ell$ relationships. Finally, *class disjointness axioms* are generated as candidate axioms for every pair of different classes, unless there is a path of rdfs:subClassOf edges in the diagram connecting one class to the other.

3 Implementation Information and Other Features

The plugin is implemented on top of the OWL API provided by Protégé. The visual components are built using mxGraph.[4] The plugin allows users to save the diagram as XML-annotated PNG, which can then be loaded again. This plugin

[4] `http://jgraph.github.io/mxgraph/`

is *not* for visualizing an ontology for which there are a number of Protégé plugins already existing, but rather, it facilitates creating graphical class diagrams *inside* Protégé and provides a way to generate axioms from it. It eliminates the need to use separate tools for creating the class diagram and writing down the axioms. In addition, the user can customize various aspects of the class diagram, e.g., coloring, size of nodes and edges, text formatting, etc., through the provided menu or by right-clicking the corresponding graphical components.

One could use this plugin for modeling from scratch, or starting from an already created ontology. In the latter case, the plugin will not attempt create a class diagram from the ontology, and rather, start with an empty canvas. Nevertheless, when the user wishes to generate axioms through the plugin, existing axioms that are already in the ontology will be included as part of the list of candidate axioms, and the user can confirm whether to keep them. Finally, we hope to continue improving this plugin, particularly to support quick modeling of modular ontologies and ontology design patterns, and furthermore, evaluate the usability of our plugin via a comprehensive user study.

Acknowledgements. This work was supported by the National Science Foundation award 1017225 *III: Small: TROn – Tractable Reasoning with Ontologies.*

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press (2007)
2. Console, M., Lembo, D., Santarelli, V., Savo, D.F.: Graphical representation of OWL 2 ontologies through Graphol. In: Horridge, M., Rospocher, M., van Ossenbruggen, J. (eds.) Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference, ISWC 2014, Riva del Garda, Italy, October 21, 2014. CEUR Workshop Proceedings, vol. 1272, pp. 73–76. CEUR-WS.org (2014)
3. Falco, R., Gangemi, A., Peroni, S., Shotton, D., Vitali, F.: Modelling OWL ontologies with Graffoo. In: Presutti, V., Blomqvist, E., Troncy, R., Sack, H., Papadakis, I., Tordai, A. (eds.) The Semantic Web: ESWC 2014 Satellite Events - ESWC 2014 Satellite Events, Anissaras, Crete, Greece, May 25-29, 2014, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8798, pp. 320–325. Springer (2014)
4. Hitzler, P., Janowicz, K., Krisnadhi, A.A.: Ontology modeling with domain experts: The GeoVocamp experience. In: d'Amato, C., Lécué, F., Mutharaju, R., Narock, T., Wirth, F. (eds.) Proceedings of the 1st International Diversity++ Workshop co-located with the 14th International Semantic Web Conference (ISWC 2015), Bethlehem, Pennsylvania, USA, October 12, 2015. CEUR Workshop Proceedings, vol. 1501, pp. 31–36. CEUR-WS.org (2015)
5. Mouromtsev, D., Pavlov, D., Emelyanov, Y., Morozov, A., Razdyakonov, D., Galkin, M.: The simple web-based tool for visualization and sharing of semantic data and ontologies. In: Villata, S., Pan, J.Z., Dragoni, M. (eds.) Proceedings of the ISWC 2015 Posters & Demonstrations Track co-located with the 14th International Semantic Web Conference (ISWC-2015), Bethlehem, PA, USA, October 11, 2015. CEUR Workshop Proceedings, vol. 1486. CEUR-WS.org (2015)