Graphical Representation of OWL 2 Ontologies through Graphol

Marco Console, Domenico Lembo, Valerio Santarelli, and Domenico Fabio Savo

Dipartimento di Ingegneria Informatica, Automatica e Gestionale "Antonio Ruberti"
SAPIENZA Università di Roma
{console,lembo,santarelli,savo}@dis.uniroma1.it

Abstract. We present Graphol, a novel language for the diagrammatic representation of ontologies. Graphol is designed to offer a completely visual representation to the users, thus helping the understanding of people not skilled in logic. At the same time, it provides designers with simple mechanisms for ontology editing, which free them from having to write down complex textual syntax. Through Graphol we can specify $\mathcal{SROIQ}(D)$ ontologies, thus our language essentially captures the OWL 2 standard. In this respect, we developed a basic software tool to translate Graphol ontologies realized with the yEd graph editor into OWL 2 functional syntax specifications.

1 Introduction

Ontologies have become popular in recent years in several contexts, such as biomedicine, life sciences, e-commerce, enterprise applications [9]. Obviously, it is very likely that people operating in such contexts are not experts in logic and generally do not possess the necessary skills to interpret formulas through which ontologies are typically expressed. This turns out to be a serious problem also in the development of an ontology. Indeed, ontologists usually work together with domain experts, the former providing their knowledge about ontology modelling and languages, the latter providing their expertise on the domain of interest. During this phase, communication between these actors is fundamental to produce a correct specification.

The use of a graphical representation for ontologies is widely recognized as a means to mitigate this communication problem. At the same time, the possibility of specifying ontologies in a graphical way might bring software analysts and experts in conceptual modelling to approach ontology modelling, since they would be provided with mechanisms that are close in spirit to those they usually adopt for software design.

Various proposals in this direction exist in the literature, but to date graphical languages for ontology have not become very popular, especially for the editing task. Among various reasons, we single out the following points: (i) many languages for graphical representation of ontologies do not capture the current standard OWL 2, and their extension to it is not straightforward (see, e.g., [3,8,7,6,1]); (ii) other proposals require the use of formulas mixed with the graphical representation (see, e.g., [4,1]); (iii) popular ontology management tools, such us Protégé[1] or TopBraid Composer[2], offer visualization functionalities, but do not support a completely graphical editing.

[1] http://protege.stanford.edu
[2] http://www.topquadrant.com/tools

To meet the main disadvantages mentioned above, in this paper we present our proposal for graphical specification and visualization of ontologies, and introduce the novel *Graphol* language, whose main characteristics can be summarized as follows:

- Graphol is completely graphical (no formulae need to be used in our diagrams) and adopts a limited number of symbols. In Graphol, an ontology is a graph, whose nodes represent either predicates from the ontology alphabet or constructors used to build complex expressions from named predicates. Then, two kinds of edges are adopted: input edges, used to specify arguments of constructors, and inclusion edges, used to denote inclusion axioms between (complex) expressions.
- Graphol has a precise syntax and semantics, which is given through a natural encoding in Description Logics.
- Such enconding shows that Graphol subsumes $\mathcal{SROIQ}(D)$, the logical underpinning of OWL 2.
- Graphol is rooted in a standard language for conceptual modeling: the basic components of Graphol are taken from the Entity-Relationship (ER) model. Notably, simple ontologies that correspond to classical ER diagrams (e.g., some OWL 2 QL ontologies) have in Graphol a representation that is isomorphic to the ER one.
- Graphol comes with some basic tools that support both the graphical editing and the automatic translation of the diagrams into a corresponding OWL 2 specification, to foster the interoperation with standard OWL reasoners and development environments.

We have adopted Graphol in various industrial projects, where we have produced large ontologies with hundreds of predicates and axioms. In such projects we could verify the effectiveness of the language for communicating with domain experts. At the same time, we exploited Graphol in the editing phase: all ontologies realized in these projects have been indeed completely specified in our graphical language, whereas an OWL functional syntax encoding thereof has been obtained automatically through the use of our translator tool. One of these experiences is described in [2], where the impact of the use of Graphol on the quality of the realized ontology is widely discussed.

We also conducted some user evaluation tests, where both designers skilled in conceptual modelling (but with no or limited experience in ontology modelling) and users without specific logic background were involved. From these tests, we obtained promising results about the effectiveness of our language for both visualizing and editing ontologies. A complete description of our evaluation study is given in [5].

For a complete description of both the syntax and the semantics of Graphol we refer the reader to [5] and to the Graphol web site[3], where it is also possible to download currently available software tools for our language. In the rest of the paper we instead discuss how the Graphol demonstration will be carried out.

2 The Graphol demonstration

In this demo we will show the process we devised to obtain an OWL 2 ontology starting from the specification of a Graphol diagram. Such a process relies on both existing

[3] http://www.dis.uniroma1.it/~graphol

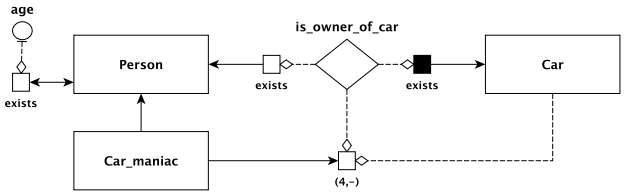

Fig. 1: A simple Graphol ontology

open source tools and original software components. More in detail, to draw a Graphol ontology we make use of the yEd editor for graphs[4], which we equip with a palette containing all and only the symbols needed for Graphol. yEd allows us to save the ontology in GraphML[5], a popular XML-based file format for encoding graphs.

An example of a Graphol ontology obtained through yEd is given in Figure 1. In the figure, the reader can see that in Graphol classes (i.e., Person, Car_maniac, Car), object properties (i.e., is_owner_of_car), and data properties (i.e., age) are modeled by labeled rectangles, diamonds, and circles, respectively, similarly to ER diagrams. The white (resp. black) square labeled with exists is a graphical constructor that takes as input a property, through a dashed arrow whose end node is a small diamond, and returns the domain (resp. the range) of the property. Such squares can have also different labels, to denote different constructs. In the example, the label `(4,–)` on the white square taking as input the is_owner_of_car property specifies a cardinality restriction on the domain of such property, i.e., it denotes all individuals participating at least 4 times to is_owner_of_car. The solid arrow always indicates a subsumption relation. This means that Car_maniac is a subclass of Person, and also of the complex class obtained through the cardinality restriction, which implies that a car maniac owns at least four cars.

Furthermore, the ontology in the example says that the domain of is_owner_of_car is Person, its range is Car, and also that each Person has an age, and that the domain of age is Person. Also, the additional dash orthogonal to the edge connecting age to its domain specifies that this property is functional.

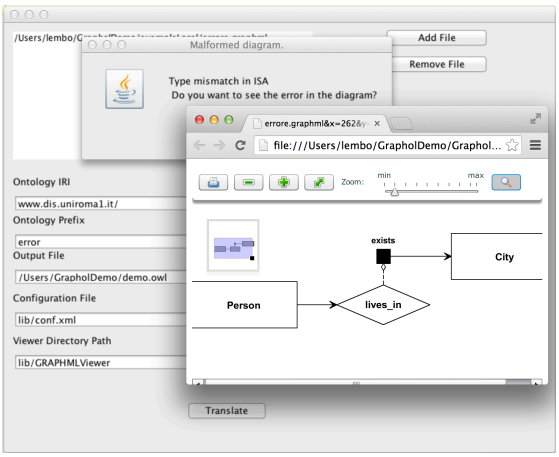

Fig. 2: The Graphol2OWL tool

The above example uses only a limited sets of constructors available in Graphol. Participants to the demo will be provided with the yEd editor and the Graphol palette to draw their own ontologies, experiencing the entire expressive power of the language.

[4] http://www.yworks.com/en/products_yed_about.html
[5] http://graphml.graphdrawing.org/

To both check the correctness of the specification and translate it into OWL 2, we developed a dedicated tool. The tool provides a syntactic validation of a given diagram: while parsing the GraphML file, if a portion of the graph is found that does not respect the Graphol syntax, the tool reports an error to the user in a pop-up window and visualizes this portion by means of an external yEd viewer. A screenshot of this tool showing an error identified in a Graphol diagram is given in Figure 2. In this example, the error consists in linking a class to a property with a solid arrow, which actually corresponds to a wrong subsumption between a concept and role.

The translator to obtain OWL 2 encodings from Graphol will be used during the demo. We will also show the compatibility of the produced OWL 2 functional syntax file with popular tools for ontology editing and management, like Protégé [6].

3 Future Work

Our main future work on Graphol is the development of editing tools (stand-alone systems or plugins of existing ontology development environments) tailored to the specification of ontologies in our graphical language and integrated with state-of the art reasoners. At the same time, we are working to improve ontology visualization in Graphol, by investigating mechanisms to automatically extract ontology views at different levels of detail on the basis of specific user requests.

References

1. do Amaral, F.N.: Model outlines: A visual language for DL concept descriptions. Semantic Web J. 4(4), 429–455 (2013)
2. Antonioli, N., Castanò, F., Coletta, S., Grossi, S., Lembo, D., Lenzerini, M., Poggi, A., Virardi, E., Castracane, P.: Ontology-based data management for the italian public debt. In: Proc. of FOIS (2014), to Appear
3. Brockmans, S., Volz, R., Eberhart, A., Löffler, P.: Visual modeling of OWL DL ontologies using UML. In: Proc. of ISWC. pp. 198–213. Springer (2004)
4. Cerans, K., Ovcinnikova, J., Liepins, R., Sprogis, A.: Advanced OWL 2.0 ontology visualization in OWLGrEd. In: In Proc. of DB&IS. pp. 41–54 (2012)
5. Console, M., Lembo, D., Santarelli, V., Savo, D.F.: The Graphol language for ontology specification, available at http://www.dis.uniroma1.it/~graphol/documentation/GrapholLVPrel.pdf
6. Dau, F., Eklund, P.W.: A diagrammatic reasoning system for the description logic $\mathcal{ALC}$. J. Vis. Lang. Comput. 19(5), 539–573 (2008)
7. Krivov, S., Williams, R., Villa, F.: GrOWL: A tool for visualization and editing of OWL ontologies. J. of Web Semantics 5(2), 54–57 (2007)
8. Object Management Group: Ontology definition metamodel. Tech. Rep. formal/2009-05-01, OMG (2009), available at http://www.omg.org/spec/ODM/1.0
9. Staab, S., Studer, R. (eds.): Handbook on Ontologies. International Handbooks on Information Systems, Springer, 2nd edn. (2009)

[6] A preview of the demo is available at http://www.dis.uniroma1.it/~graphol/research.html.