
Production Systems and Information Engineering
Volume 10 (1), pp. 64–71
doi: 10.32968/psaie.2022.1.5.

RELONTOUML MODEL OF THE ARCHAEOLOGICAL FINDINGS

Anita Agárdi
University of Miskolc, Hungary
Institute of Informatics
agardianita@iit.uni-miskolc.hu

Abstract. In this article, an ontology describing archaeological finds (DhiQar) is converted into the RelOntoUML model, which combines relational database modelling, ontology, and UML modelling. The ontology describing the archaeological finds can be found in OWL on github, it is open source. In this article, the original OWL ontology is also presented using the graphical representation system of the Protége editor. And the RelOntoUML model of the system is also detailed.

Keywords: Ontology, UML, relational model, Protége, draw.io

1. Introduction

In this article, the transformation of an OWL ontology describing archaeological finds (DhiQar [1]) into the RelOntoUML [2] model is presented. Ontology [3] is the representation of the knowledge, and modelling it. UML [4] is a unified modelling language used in software development to the description of systems. The relational database model [5] consists of tables, where the tables consist of columns and rows. Columns determine the types of values are within the tables, while rows represent data with different values. One of the most common languages for ontologies is OWL (Web Ontology Language [6]), which is an XML-like description. However, we don't have to write the system ourselves with hundreds of lines of OWL, we can also generate the OWL with graphical development using Protége [7]. Protége represents and creates ontological classes and properties graphically. Draw.io [8] is a general web interface for drawing. In the following, the literature review is presented, how many publications Protége and draw.io have been used over the years. For this, the google scholar is used, where the search keywords were 'Protége' and 'draw.io'.

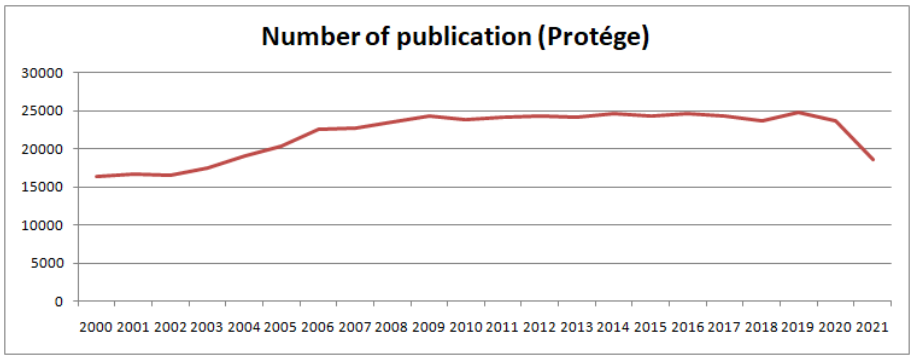

Figure 1. Number of publication (Protége)

Figure 1 presents the number of publications broken down by year. The number of publications increased until 2019 and then started to decrease from 2019.

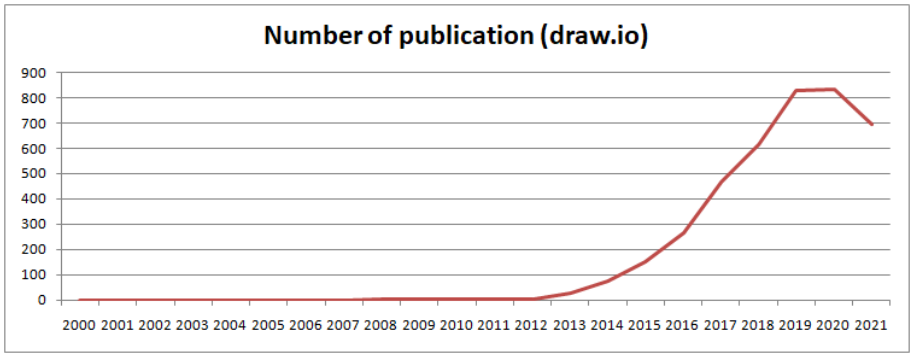

Figure 2. Number of publication (draw.io)

Figure 2 illustrates the number of draw.io publications. As the draw.io is not so old, there was no publication on the subject in the early 2000s. There are no publications since 2013, and the number of publications shows an increasing trend until 2018, from which it decreases slightly.

Figure 3 shows the number of publications in a chart with Protége and the draw.io editor. There have been more publications with the Protége editor as with draw.io.

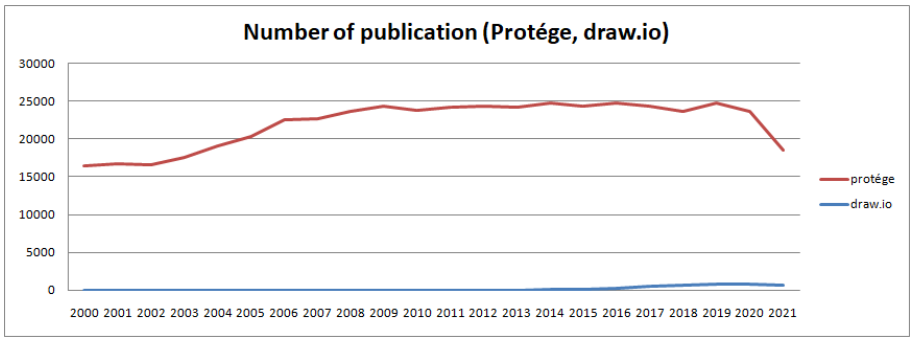

Figure 3. Number of publication (Protége, draw.io)

2. Illustration in Protége and draw.io

Protége [7] represents and creates ontological classes and properties graphically. Inside the entities tab, clicking on classes we will find the classes, for example for the DhiQar ontology, as shown in Figure 4.

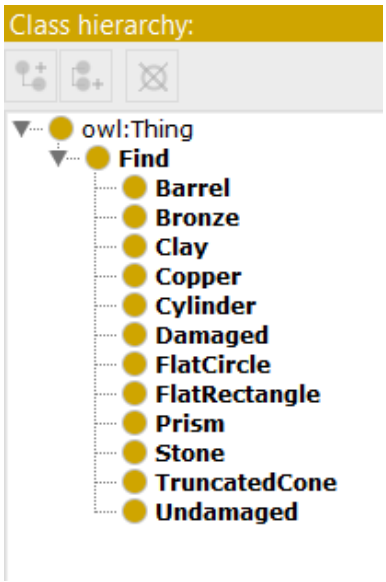

Figure 4. Class hierarchy

The object property, datatype property, annotation property, datatypes, and individuals tabs can be also found in Protége. OWLViz [9] visualizes the

ontology with class-subclass level. Figure 5 illustrates the DhiQar system using OWLViz.

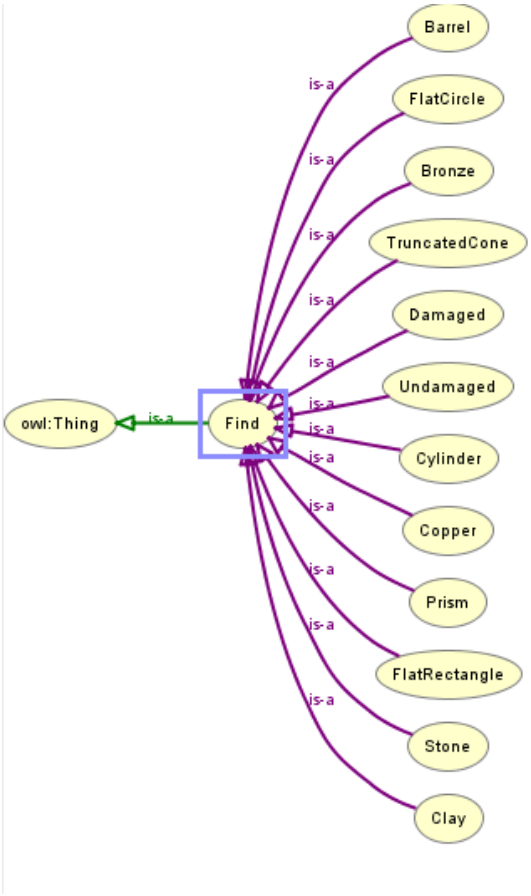

Figure 5. OWLViz representation of the DhiQar

OntoGraf [10] has also been developed to graphically represent ontology. The Ontograf model of the DhiQar system is illustrated in Figure 6.

The VOWL [11] graph represents the ontology. The result is readable for small ontologies (such as DhiQar), but unfortunately for large ontologies we get a tangled graph.

Draw.io [8] is a general drawer that can be used to represent almost any area. It classifies items into the following types:

- Standard (General, Basic, Arrows, Clipart, Flowchart)

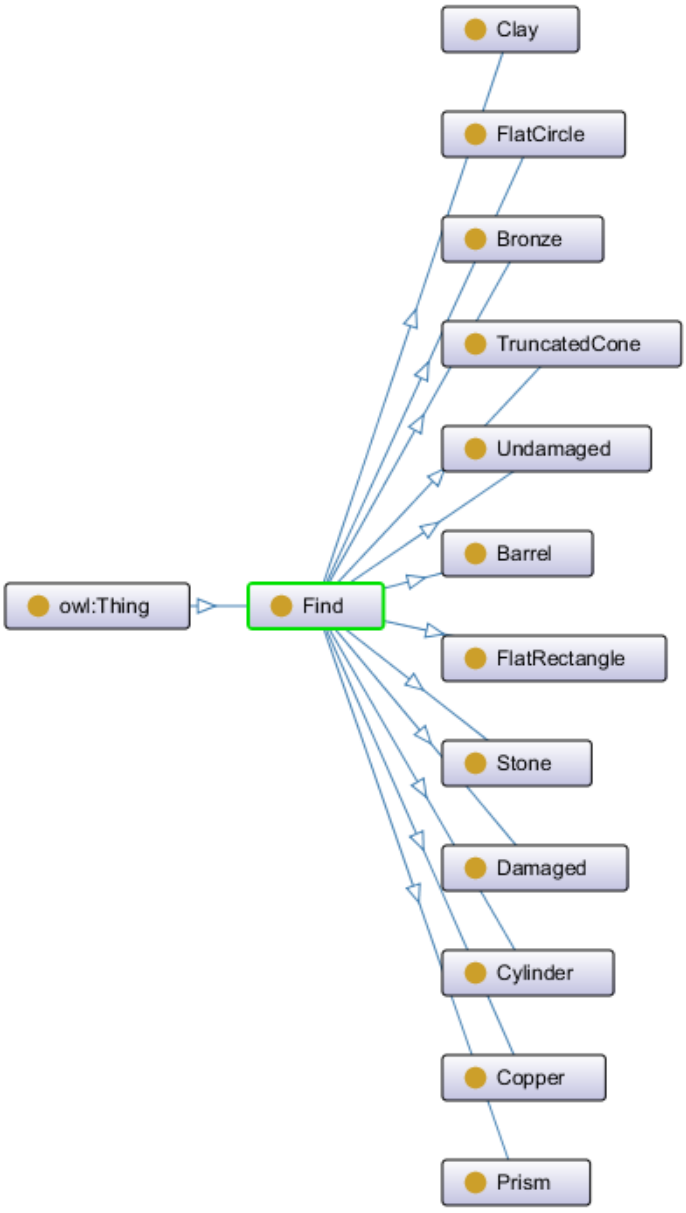

Figure 6. OntoGraf representation of the DhiQar

- Software (Active Directory, Android, Atlassian, Bootstrap, C4, Data Flow Diagram, Entity Relation, iOS, Mockups, Sitemap, UML 2.5, UML)

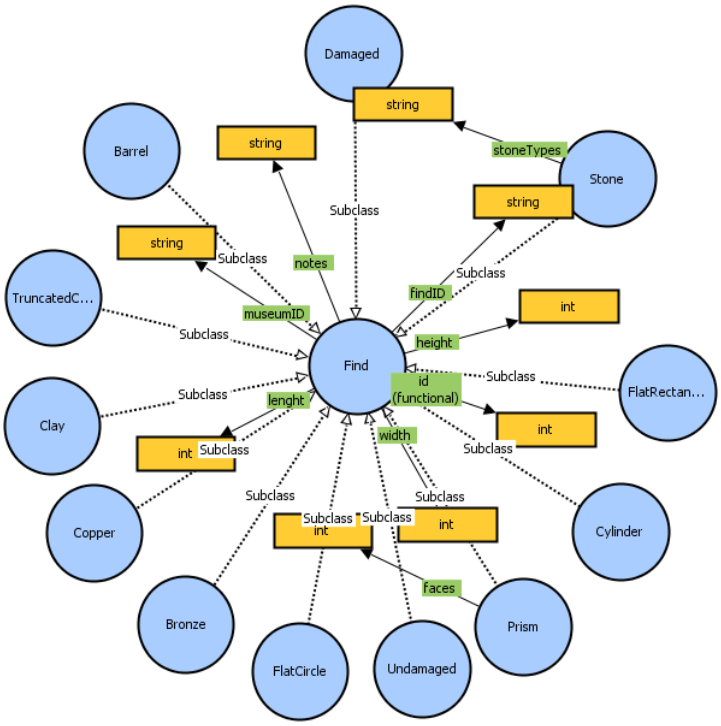

Figure 7. VOWL representation of the DhiQar

- Networking (Allied Telesis, AWS17, AWS18, AWS19, AWS 3D, Azure, Cloud & Enterprise, Cisco, Cisco19, Cisco Safe, Cumulus, Citrix, Google Cloud Platform, IBM, Kubernetes, Network, Office, Rack, Veeam, VMware)
- Business (ArchiMate 3.0, ArchiMate 2.1, BPMN 2.0, SysML, Value Stream Mapping)
- Other (Cabinets, Infographic, EIP, Electrical, Floorplans, Fluid Power (ISO 1219), Material Design, Proc. Eng., Threat Modeling, Web Icons, Signs)

3. Conversion of an ontology describing archaeological finds to a RelOntoUML model

The DhiQar [1] ontology describes archaeological findings. The child of the 'Thing' class is the 'Find' class, which contains a number of properties and comments. Identification of the find, size (height, width), weight, museum to which it belongs. This class includes additional subclasses. Each subclass

contains a comment, describing the meaining of the class. These classes are 'Barrel', 'Bronze', 'Clay', 'Copper', 'Cylinder', 'Damaged', 'FlatCircle', 'FlatRectangle', 'Prism', 'Stone', 'TruncatedCone ',' Undamaged '. The ontology does not contain individuals.

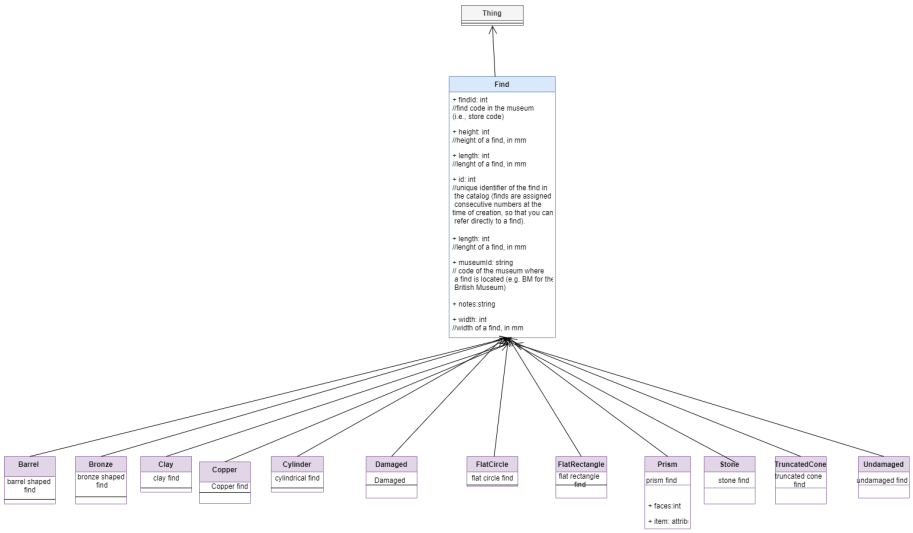

Figure 8. Visualization of the DhiQar with RelOntoUML

4. Summary

This article describes an ontology describing archaeological finds for the conversion of DhiQar to RelOntoUML model. First, the article reviews the editing interface, the Protége ontology interface, and the draw.io drawing interface. The article then presents the RelOntoUML model of the ontology describing the archaeological finds. The DhiQar ontology contains some classes and properties. Another area of research is the conversion of other ontologies to the RelOntoUML model.

Acknowledgements

SUPPORTED BY THE ÚNKP-21-3 NEW NATIONAL EXCELLENCE PROGRAM OF THE MINISTRY FOR INNOVATION AND TECHNOLOGY FROM THE SOURCE OF THE NATIONAL RESEARCH, DEVELOPMENT AND INNOVATION FUND.

References

[1] *DhiQar ontology* https://github.com/Epistematica/dhiqar-ontology.git (Accessed: 2022.05.14.)

[2] Agárdi, A. *RelOntoUML: Development of a model based on relational model, ontology and UML.* Műszaki Tudományos Közlemények (accepted)

[3] Li, S. P., Yin, Q. W., Hu, Y. J., Guo, M., & FU, X. J. (2004). *Overview of researches on ontology.* Journal of computer research and development, 7, 1041-1052.

[4] France, R., Evans, A., Lano, K., & Rumpe, B. (1998). *The UML as a formal modeling notation.* Computer Standards & Interfaces, 19(7), 325-334.

[5] Chou, H. T., & DeWitt, D. J. (1986). *An evaluation of buffer management strategies for rela-tional database systems.* Algorithmica, 1(1-4), 311-336. https://doi.org/10.1007/bf01840450

[6] Antoniou, G., & Van Harmelen, F. (2004). *Web ontology language: Owl.* In Handbook on on-tologies (pp. 67-92). https://doi.org/10.1007/978-3-540-24750-0_4

[7] Noy, N. F., Crubézy, M., Fergerson, R. W., Knublauch, H., Tu, S. W., Vendetti, J., & Musen, M. A. (2003, January). *Protégé-2000: an open-source ontology-development and knowledge-acquisition environment.* In AMIA... Annual Symposium proceedings. AMIA Symposium (pp. 953-953).

[8] *draw.io* https://app.diagrams.net/ (Accessed: 2022.05.14.)

[9] *OWLVIZ* https://github.com/protegeproject/owlviz (Accessed: 2022.05.14.)

[10] *OntoGraf* https://github.com/protegeproject/ontograf (Accessed: 2022.05.14.)

[11] *VOWL* https://protegewiki.stanford.edu/wiki/VOWL (Accessed: 2022.05.14.)