
Production Systems and Information Engineering
Volume 10 (1), pp. 100–107
doi: 10.32968/psaie.2022.1.9.

CONVERSION OF COMPANIES AND SOURCE CODE COMMENT ONTOLOGY INTO RELONTOUML

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

Abstract. In this article, the conversion of a Companies and a source code commenting (SEM) ontology system into the RelOntoUML modelling system is presented. The two ontology systems are in OWL format available on github. After the simplification of the OWL files, I converted these two systems to RelOntoUML. RelOntoUML is a proprietary modelling that combines ontology, UML, and relational database modeling.

Keywords: UML, ontology, OWL, database, Protége

1. Introduction

Ontology [1] is the representation of knowledge. It represents systems with classes, individuals, object properties, datatype properties, and inheritance. In addition to representing the systems, the purpose is to infer from the data using the reasoning engine. Using google scholar, searching for the keyword 'Ontology' over the years 2000–2021 gives the following results, as shown in Figure 1.

The x-axis (Figure 1) shows the year and the y-axis presents the number of publications in that year. We can see that the number of publications increased between 2000 and 2012 and decreased between 2013 and 2021. UML (Unified Modelling Language) [2] is a schematic description of systems that is easy to understand. It is a very popular modelling language for software developers. It can be used with structural, behavioural diagrams. For example, a structural diagram is a class diagram or a package diagram, and a behaviour diagram is a use case diagram. Figure 2 shows an annual breakdown of published publications using the google scholar search engine, where keyword = 'UML'.

The figure shows that in 2002 we found most of the articles on UML. After a drastic decrease, the number of articles increased again until 2010, and then it

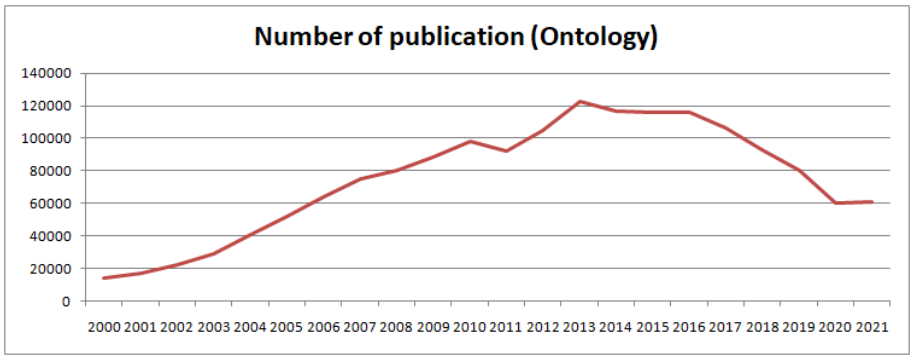

Figure 1. Number of publication (Ontology)

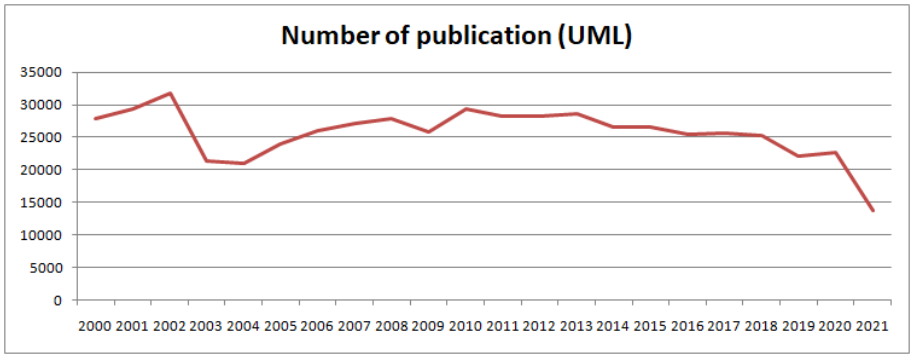

Figure 2. Number of publication (UML)

started to decrease again. Figure 3 shows the search results for the 'Relational database model'.

The figure shows that until 2018, the number of publications on the topic increased steadily, and then it started to decrease. Figure 4 illustrates the results of the publications of the three models.

Figure 4 together illustrates the results of the three modelling. Most publications on ontology have been born since 2002. About the same number of publications have been produced in the UML and relational database topics. In the following, I will present the ontology system that implements companies and source code commenting, and then their conversion to the RelOntoUML [3] model.

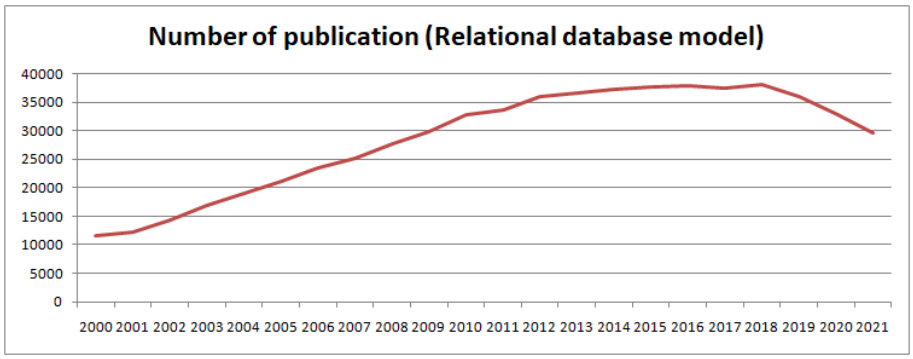

Figure 3. Number of publication (Relational database model)

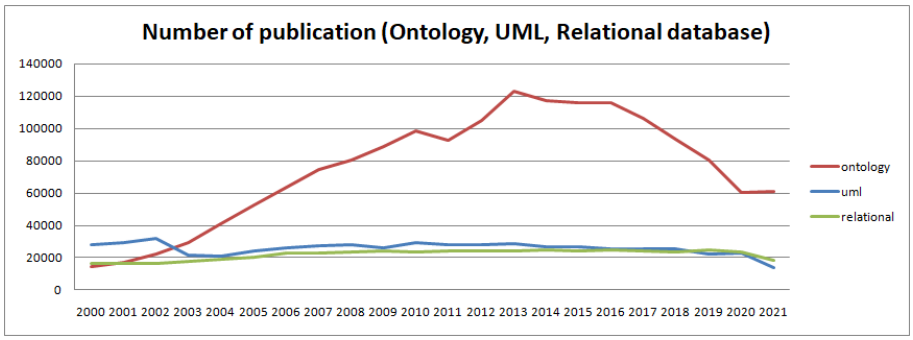

Figure 4. Number of publication (Ontology, UML, Relatinal database model)

2. Transforming companies and software source code commenting ontology into a RELOntoUML model

In this section the transformation of two OWL ontologies into RelOntoUML model [3] is presented. The RelOntoUML model is the combination of relational model, ontology and UML modeling. This model contains namespaces, classes, subclasses, individuals, object properties, minCardinality, maxCardinality, datatype properties, transitive properties, symmetric properties, functional properties, inverseOf properties, equivalentClasses, equivalent properties, sameAs properties.
Classes describe the system and it can be hierarchized into classes-subclasses. Individuals are the instances of the classes. The individuals can be also hierarchized into individuals-subindividuals. The object property assiciates two objects, the datatype property associates a datatype and an individual. The

min and max cardinalities are describe a range: a number of times a class is associated with another class or datatype. Transitive property is between 3 properties. If property A -> property B and property B -> property C then property A -> property C. Symmetric is between 2 properties. If property A -> property B then property B -> property A. Functional property is between 2 properties. If property A -> property B and property A -> property C then property b = property C. The inverseOf property is between 2 properties. It means, that if property A is the inverse of property B and vice versa. The equivalentClasses means, that 2 classes (with different names) are equivalent. The equivalentProperties means, that 2 properties (with different names) are equivalent. The sameAs means, that 2 individuals (with different names) are equivalent. [3]
The Companies ontology [4] is for general company modeling. It contains different types of companies such as Energy, Healt_Care, Food, etc. These types also have subtypes, e.g., the Energy class has the following subclasses: Oil_Exploration, Solar_Energy . This ontology also includes few individuals, such as Bank_Of_America, BMW, Capital_One, Charles_Schwabb. These individuals can be individuals of several classes, for example, the BMW entity is an individual of the Automobile_Manufacturing and German_Corporations classes. This otology contains few individuals, the last level classes, which no longer have subclasses. The ontology consists of of 26 individuals.

In the RelOntoUML model, we can also see that the ontology does not contain object and datatype properties. In the figure, it can be also seen, that the ontology contains 2 levels in addition to owl: Thing. The individuals are also located on the last level. On the second level are the following classes: Food, Energy, Financial_Services, Foreign_Corporations, Health_Care, Hospitality, Manufacturing, Retail_Stores, US_Corporations. The third level classes are Oil_Exploration, Solar_Energy, Commercial_Bank, Consulting, Financial_Advisor, Investment_Bank, Japanese_Corporations,
British_Corporations, French_Corporations, German_Corporations, Medical_Equipment, Pharmaceutical_Manufacturer, Aircraft_Manufacturing, Automobile_Manufacturing, Electrical_Equipment_Manufacturing,
Medical_Equipment, Fortune_100_Companies. The ontology also contains many individuals, which are: Examples of the Food class: General_Mills, Kraft. Instances of the class Oil_Exploration are Exxon_Mobil. Individuals of Commercial_Bank are Bank_Of_America, Citibank, Capital_One, Wells_Fargo, and Chase_Bank. Individuals of Japanese_Corporations are Toyota. Individuals of the Consulting class: Price_Waterhouse_Cooper, Deloitte, McKinsey. The individuals of the Financial_Advisor class are
Charles_Schwabb, Fidelity, Morgan_Stanley, Franklin_Templeton. Individuals of the Investment_Bank class: Credit_Suisse, Goldman_Sachs,

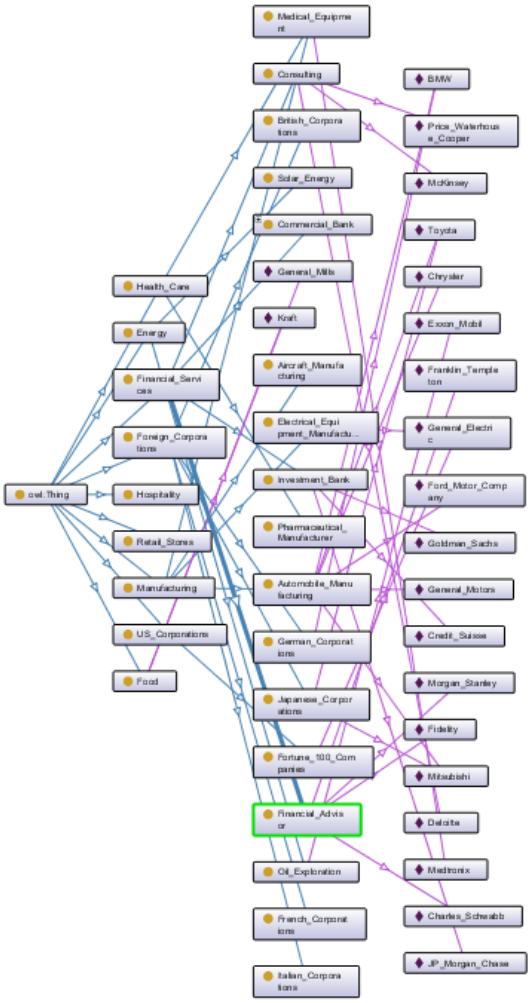

Figure 5. Visualization of the Companies with OntoGraf

JP_Morgan_Chase.
Individuals in the German_Corporations class: BMW. The individual of Medical_Equipment is Medtronix. The Automobile_Manufacturing class has more individuals than BMW, Chrysler, Ford_Motor_Company, General_Motors.

SEM [5] presents software commenting. The following classes have been defined by the ontology, which are located on a total of 4 levels. The first level is the 'Thing' class. The second level is 'comment' class. The third level

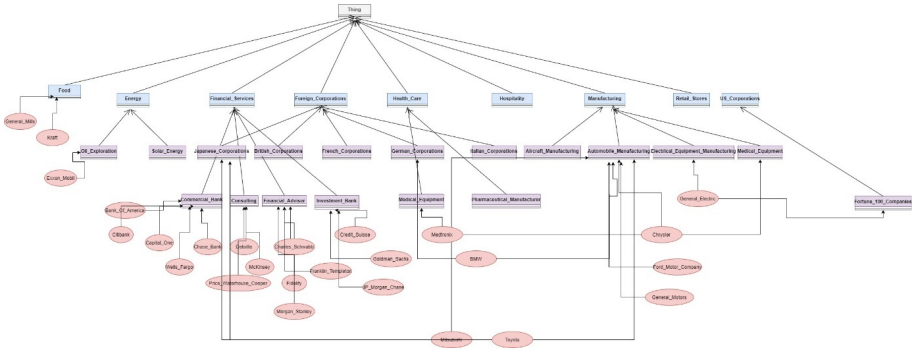

Figure 6. Visualization of the Companies with RelOntoUML

is 'I_have_written_it' class. The 'XY_have_written_it' class can be also added here, depending on software developers who have written the comment. At the last level, the classes are the following:

- autogenerated (IDEs automatically generate a comment when we generate classes and functions with their help),
- documention_exists (software developers also create documentation about the source code so that
- later apply to the code,
- it is easier to understand the operation,
- easier to remember the writer, or other software developer easier to understand),
- it_was_a_one_shot,
- documention_is_missing,
- i_do_not_understand_how_it_could_work,
- i_do_not_remember_why_I,
- just_a_proof_of_concept,
- it_is_deprecated (some features are no longer used, not supported).

It can be seen that neither data nor object properties were created for the SEM ontology. The ontology does not have any individuals.

3. Summary

In this article, I transformed two ontologies into the RelOntoUML [3] model, bringing the ontologies closer to the software developer approach. The transformed models differ greatly in both the modelled topic and the size of the ontology. The companies [4] model contains many classes and individuals, SEM [5] ontology contains no individuals and contains only a few classes, it

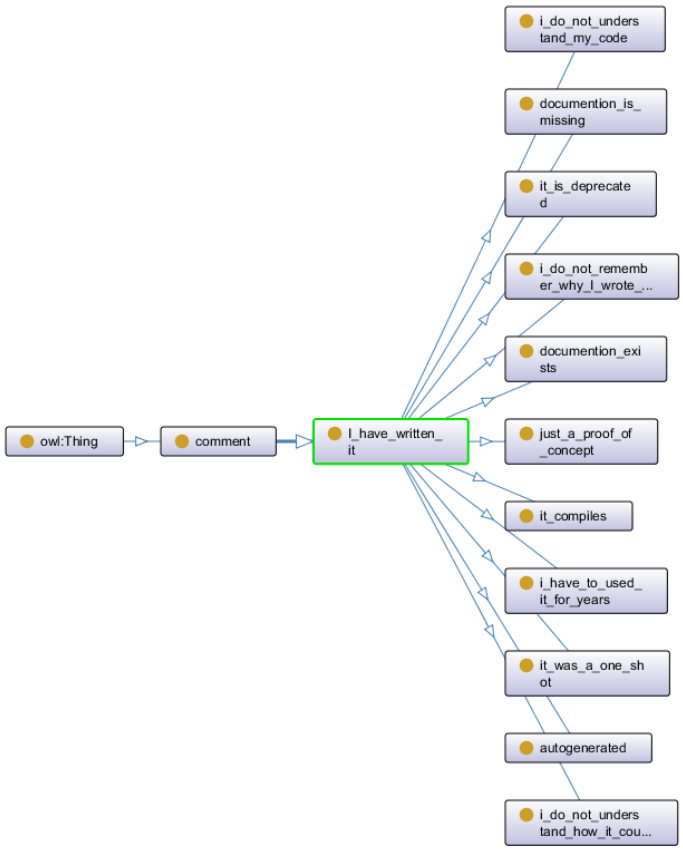

Figure 7. Visualization of the SEM with Ontograf

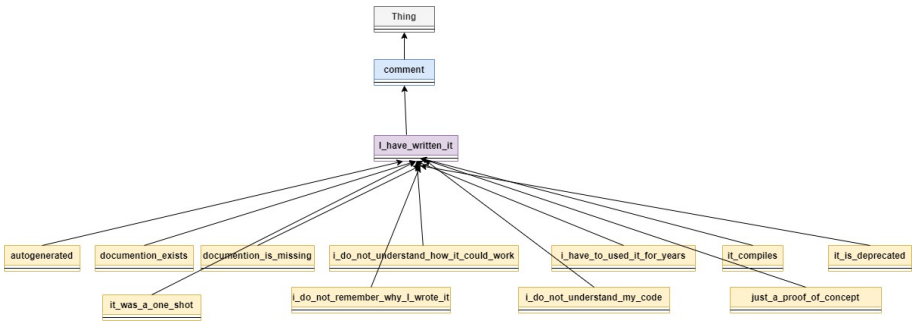

Figure 8. Visualization of the SEM with RelOntoUML

can be considered as a small ontology. The investigated ontologies do not contain properties. My future work plan is the refinement of the RelOntoUML [3] model and the modelling of other sample ontology systems.

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] Smith, B.: Ontology. In *The furniture of the world*, pp. 47–68, Brill, 2012, URL https://doi.org/10.1163/9789401207799_005.

[2] Pilone, D. and Pitman, N.: *UML 2.0 in a Nutshell.* " O'Reilly Media, Inc.", 2005.

[3] Agárdi, A.: Relontouml: Development of a model based on relational model, ontology and uml. *Műszaki Tudományos Közlemények*, **16**(1), (2022), 1–4.

[4] Companies. URL https://github.com/detnavillus/rdf-owl-ontologies.git. Accessed: 2022-05-14.

[5] Sem. URL https://github.com/lindenb/semontology.gitt. Accessed: 2022-05-14.