Műszaki Tudományos Közlemények vol. 16. (2022) 1–4.
DOI: Hungarian: https://doi.org/10.33895/mtk-2022.16.01
English: https://doi.org/10.33894/mtk-2022.16.01

RELONTOUML: DEVELOPMENT OF A MODEL BASED ON RELATIONAL MODEL, ONTOLOGY AND UML

Anita AGÁRDI

University of Miskolc, Faculty of Mechanical Engineering and Informatics, Institute of Informatics, agardianita@iit.uni-miskolc.hu

Abstract

This paper presents a model that combines ontology, UML modeling, and a relational model. The ontology model (and the Ontology Web Language - OWL), UML, and relational model are first introduced in the article. After a review of the literature, the comparison and conversion of the systems are presented. The created model is then presented and a real ontology is modeled using the presented model.

Keywords*: ontology, OWL, UML, relational model.*

1. Introduction

The modeling and visualization of a system is an important task. In the literature, such standard modeling languages are the Unified Modeling Language (UML), the relational database model, and the ontology model. UML (Unified Modeling Language) [1] is a standard, general-purpose modeling language. A tool for visually documenting models of large-scale software systems can be used to create textual and graphical models from a variety of perspectives, including: systems, organizations, business, processes, software, programs and databases. They include structural diagrams and behavioural / dynamic diagrams. Structural diagrams refer to the elements of the modeled system. These subtypes are: Class Diagrams, Component Diagrams, Complex Structure Diagrams, Deployment Diagrams, Object Diagrams and Package Diagrams. Behavioural / dynamic diagrams describe what should happen in the modeled system. The following are the subtypes: Activity Diagrams, State Machine Diagrams, Use Case Diagrams and Interaction Diagrams.

Ontology [2] is a description of knowledge in a particular area. It contains semantic objects in a given area. One of the best-known languages of ontology is Ontology Web Language (OWL). OWL builds the system from classes - subclasses. An individual is a specific occurrence of a class. Classes can contain different properties. The property has a domain and range. In the datatype property, the domain is a class, but the range is a datatype, while in the object property, both the domain and the range are a class, so the datatype property associates an entity with a data type, while the object property associates one entity with another entity. The annotation property connects an individual, class, property with an annotation.

The OWL ontology, like the UML and the relational model, has been used in many places, such as health: health ontology systems [3], COVID-19 [4, 5], education: university ontology [6], software technology [7].

Among the database models, the relational model [8] is the most common today. The relation is a table. The relation itself is given with a unique name, its rows represent the data. Column names are given with a unique name within the relation, but another relation can contain a column with the same name. The intersection of a column and a row is called a field.

2. RELONTOUML model

In the following, the developed model is presented. The model is based on the relational model, ontology, and the UML model, and called the RelOntoUML model. The goal was to develop a model that is close to the approach of software developers and easy for software developers to understand. The essence of the model is that it consists of properties from both UML, ontology, and relational models, making it easier to model software systems.

The steps of the transformation are described below, during which the ontology model is converted to the RELOntoUML model.

The ontology uses namespaces; they must be unique names within a namespace. In UML, classes within a package and tables within a database must have unique names [9]. This is converted to the RelOntoUML model as a package, shown in the figure 1.

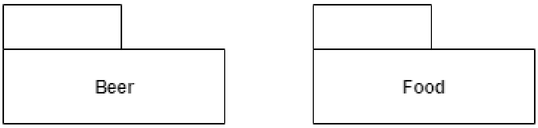

Figure 1. *Namespace (RelOntoUML)*

Classes are descriptors of the system. Ontology classes are simply converted to UML classes [9].

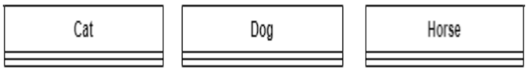

Figure 2. *Classes (RelOntoUML)*

The class-subclass hierarchy is retained in the model. [9]

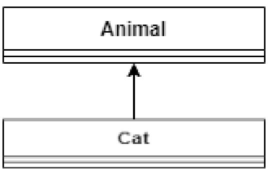

Figure 3. *Subclass (RelOntoUML)*

Individuals are occurrences of classes [9].

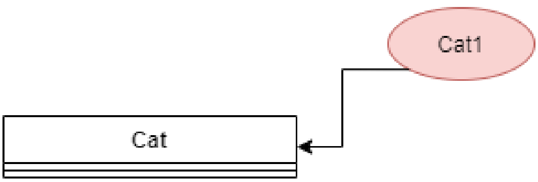

Figure 4. *Individual (RelOntoUML)*

The object property connects two objects, so both the domain and the range are objects.[9]

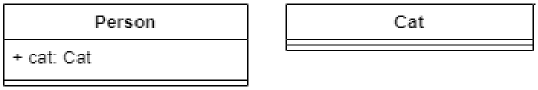

Figure 5. *Object property (RelOntoUML)*

The minCardinality, maxCardinality determines the number of times a class is linked to another class or data type [9].

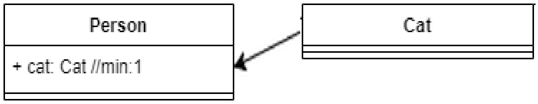

Figure 6. *minCardinality, maxCardinality (RelOntoUML)*

The datatype property associates an individual with a data type [9].

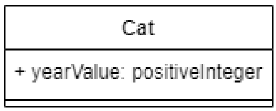

Figure 7. *Datatype property (RelOntoUML)*

A transitive property means that if a property is both (A,B) and (B,C) then it is also (A,C) [9].

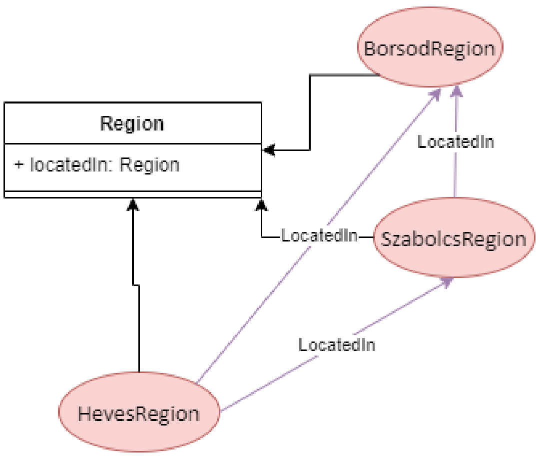

Figure 8. *Transitive property (RelOntoUML)*

A symmetric property in the ontology means that if a property is (A,B), it is also (B,A) [9].

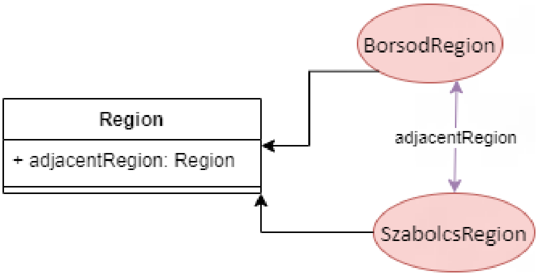

Figure 9. *Symmetric property (RelOntoUML)*

A functional property means that if a property is (A,B) and (A,C), then B = C **[9]**.

Figure 10. *Functional property (RelOntoUML)*

The inverse object property means that one object property is the inverse of the other **[9]**.

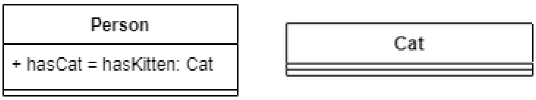

Figure 11. *InverseOf property (RelOntoUML)*

In an ontology, two classes can be created with different names, which, if denoted by the equivalentClass, are equal classes with different names **[9]**.

Figure 12. *EquivalentClasses (RelOntoUML)*

In an ontology, you can create two properties with different names, which, if given the equivalentProperty flag, are equal properties with different names **[9]**.

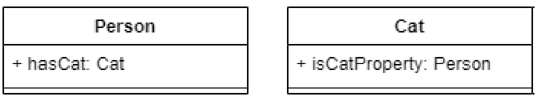

Figure 13. *Eqivalent property (RelOntoUML)*

The sameAs means that two individuals with different names in the ontology are equal **[9]**.

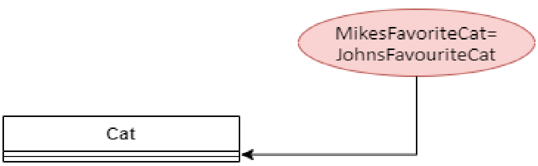

Figure 14. *sameAs (RelOntoUML)*

3. Application of RELONTOUML model on a sample system

SoftwareTechnology **[10]** is a larger ontology that models software technology. It models software development companies, programming and data description languages, software licenses, platforms (cloud, os), protocols (eg http, ftp) and software product types (eg CMS, CRM).

The ontology models software technology 4 levels with classes and one level with individuals. There are no properties for either individuals or classes.

On the first level is the Thing class, which is the ancestor of all classes in all ontologies. On the second level is Software_Technology. At the third level are the following classes: Software_Company, Software_Language, Software_License, Software_Platforms, Software_Product, Software_Protocol, System_Monitoring. At the fourth level are the following classes: Data_Description_Language, Programming_Language, Cloud_Computing, Operating_Systems, Virtual_Machines, Content_Management_Systems, Search_Engines, Customer_Relationship_Management, Security, Network_Protocol, Security_Protocol .

The ontology also contains a number of individuals. The Software_Company includes the following entities: Adobe, Amazon, Google, Microsoft, Hewlett-Packard, IBM, Oracle, Jetbrains. Data_Description_Language contains the following entities: HTML, XML. The following programming languages are included: C, C ++, Java, JavaScript, Perl, PHP, Ruby.

4. Conclusions

The article presents ontology, relational, and UML models. After the literature of model conversion, the RelOntoUML model was presented, and the transformation of the ontology OWL to the RelOntoUML model. The SoftwareTechnology OWL was converted into RelOntoUML model. The model presents software technology, and its elements such as software company, programming language, license, platform, software product, protocol. The advantage of the developed model is that it provides software developers with easy-to-use modeling. The system created in this way will be well-structured and easy to understand. Another area of research is the conversion of new ontologies to the RelOntoUML model, and possible refinements of the model if required.

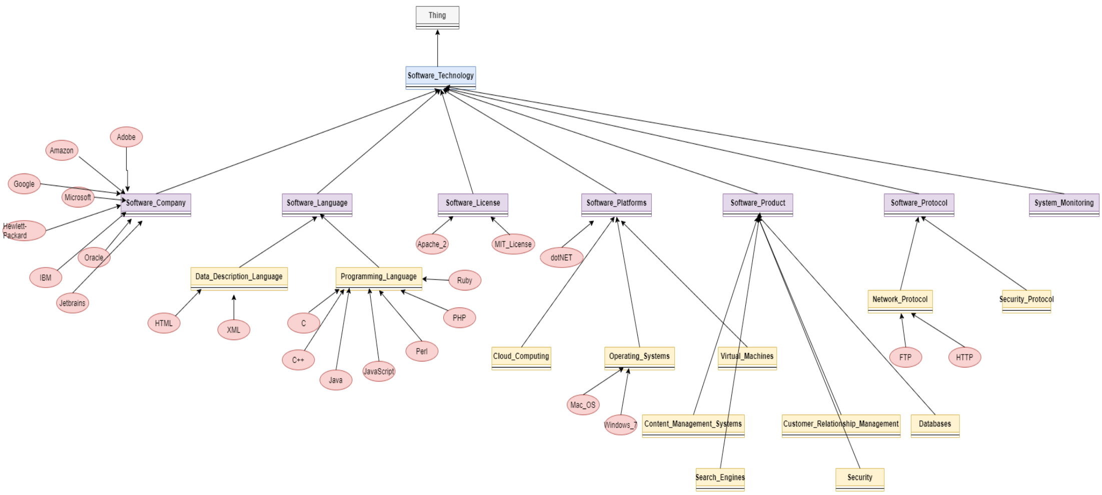

Figure 15. *Visualization of SoftwareTechnology with RelOntoUML*

Acknowledgement

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] Dobing B., Parsons J.: *Dimensions of UML Diagram Use: a Survey of Practitioners*. Journal of Database Management (JDM), 19/1. (2008) 1–18.

[2] Antoniou G., Van Harmelen F.: *Web Ontology Language: Owl.* In: Handbook on Ontologies (2004) 67–92.

[3] Kiong Y. C., Palaniappan S., Yahaya N. A.: *Health Ontology System.* In: 2011 7th International Conference on Information Technology in Asia (2011) 1–4.

[4] Dutta B., DeBellis M.: *CODO: An Ontology for Collection and Analysis of COVID-19 Data*. arXiv preprint arXiv:2009.01210 (2020).

[5] Kachaoui J., Larioui J., Belangour A.: *Towards an Ontology Proposal Model in Data Lake for Real-Time COVID-19 Cases Prevention*. (2020).

[6] Malviya N., Mishra N., Sahu S.: *Developing University Ontology Using Protégé Owl Tool: Process and Reasoning*. International Journal of Scientific & Engineering Research, 2/9. (2011) 1–8.

[7] Gherabi N., Bahaj M.: *A New Method for Mapping UML Class into OWL Ontology*. Spec. Issue Int. J. Comput. Appl.(0975 8887) Soft. Eng. Databases Expert Syst. SEDEXS, (2012) 5–9.

[8] Codd E. F.: *Relational Database: A Practical Foundation for Productivity.* In: *Readings in Artificial Intelligence and Databases* (1989) 60–68.

[9] OWL Guide: https://www.w3.org/TR/owl-guide/ (accessed on: 2021. 11. 09.)

[10] SoftwareTechnology. (accessed on: 2021. 11. 09.) https://github.com/detnavillus/rdf-owl-ontologies.git