*Production Systems and Information Engineering*
Volume 10 (1), pp. 72–76
doi: 10.32968/psaie.2022.1.6.

# VISUALIZATION OF THE DESCRIPTION OF A CRYPTOCURRENCY (DOACC) ONTOLOGY WITH RELONTOUML

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

**Abstract.** This article transforms the Description of a Cryptocurrency (DOACC) ontology OWL into RelOn-toUML model. The ontology (DOACC) can be downloaded from github, an open-source project. The RelOntoUML model is a combination of the relational database model, the ontology model, and the UML model. The relational model stores data in relations (tables). Ontology is a representation of knowledge and also has an inference engine. The UML model is a popular and frequently used model-ling by software developers. The article also presents the doacc ontology and the conversion of the ontology.

*Keywords*: ontology, UML, OWL

## 1. Introduction

The purpose of this paper is to present a Description of a Cryptocurrency (doacc [1]) and convert it to the RelOntoUML model. RelOntoUML [2] combines relational database, ontology, and UML model-ling. In the relational model [3], the data is stored in tables, and the tables are named. Tables consist of rows and columns, where columns represent what property we store, and rows represent a specific piece of data. The intersections of the rows and columns are the fields, in which the elementary data can be found. Ontology [4] is the representation of the knowledge. Ontology consists of individuals and their classes. Classes are organized into a class-subclass hierarchy. Classes can have properties that can also be organized into a hierarchy. We distinguish between two types of properties, the object property, which connects two entities, and the data property, which connects an individual to an elementary type. The annotation property connects an individual to an annotation. UML (Unified Modelling Language) [5] is a popular modelling language for software developers that includes many chart types. One of the best-known such diagrams is the class diagram, which includes the classes of the system, their properties and methods. In the following, I will present some publications investigating with relational model, ontology and UML modelling, their comparison and conversion. Ha, Y., & Lee, R. [6] investigated the ontology-UML conversion. The following have been reported: Ontology = Package; minCardinality, maxCardinality, inverseOf = Multiplicity; disjointWith, unionOf = disjointWith, cover; Domain, Range = Navigable, non-navigable; One of = Enumeration; Class, property = N-ary association; Subclass, Subproperty = Subclass, generalization; Property = ownedAt-tribute, binary association; Individual = Instance; Class = Class, property, ownedAttribute, type; Evermann, J. [7] also investigated the possibilities of ontology – UML conversion. The conversion result is the following: Multiple Constraint = FunctionalProperty, SameAs, DifferentFrom, AllDiffer-ent; Mutual generalization = EquivalentProperty; Mutual generalization = EquivalentClass, Disjoint-With; Attribute variability and initial value = Data type property (DatatypeProperty) constraint: has-Value; Attribute typeFeature = Data type property (DatatypeProperty) value constraint: allValuesFrom, Data type property (DatatypeProperty) value constraint: someValuesFrom; Associationend changeability = Object property (ObjectProperty) value constraint: hasValue; AssociationEnd partici-pant = Object property (ObjectProperty) value constraint: allValuesFrom, Object property (Object-Property) value constraint: someValuesFrom; Objects = OneOf (class definition with enums); Asso-ciationEnd isNavigable = Inverse (InverseOf), Transitive Property (TransitiveProperty), Symmetric Property (SymmetricProperty); Generalization = IntersectionOf, CoplementOf; Generalization = Un-ionOf; Multiplicity = Cardinality; Generalization between associations = subPropertyOf; Generalization between classes = subClassOf; AssociationClass = Class and object property (ObjectProperty); Association = ObjectProperty; Attribute = DatatypeProperty; Class = Class; Pătraşcu, A. [8] also investigated with the equivalent of the UML and ontology: package = ontology; Multiplicity = minCardinality, maxCardinality; Enumeration = OneOff; Subclass = Subclass; Attribute, Binary association = Property; Individual = Individual; Class = Class; In the following, I present the Description of a Cryptocurrency (doacc [1]) system, the ontology model, and then the conversion of RelOntoUML, which is also based on the ontology-UML conversions mentioned above.

## 2. Modeling the Description of a Cryptocurrency (doacc) with ontology and RelOntoUML

The doacc (Description Of A CryptoCurrency) [1] ontology contains metadata about cryptocurrencies. The ‘Thing’ is the first class level of the ontology, and it has a number of properties. These include ’+ audience: string’, ’+’ block reward scheme ’: string’, ’+’ block target time ’: string’, ’+ category: string’, ’+’ exchange trading symbol ’: string’, ’ + expiration: string ’etc. The ‘Thing’ has the following subclasses: ‘Approach to protecting the public ledger’, ‘Blockchain protocol’, ‘Collection’, ‘Concept’, ‘Concept Scheme’, ‘Cryptocurrency’, ‘Cryptographic hash function’.

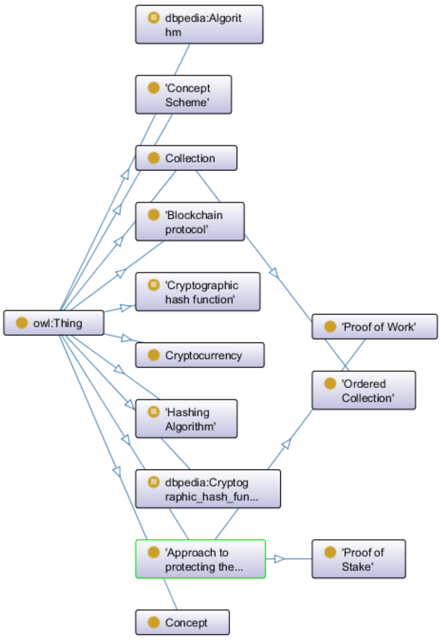

**Figure 1.** Visualization of the doacc ontology with ontograf

The ontology contains only classes and properties, but it does not contain individuals.

The VOWL [9] visualization is shown in Figure 2. The VOWL presentation is not very readable and does not contain all the ontological elements. Figure 3 illustrates the RelOntoUML [2] visualization, which contains all the ontological elements, in a way that is popular with software developers. Gray is the ‘Thing’ class, which is the ancestor of each class, with the classes below (children)

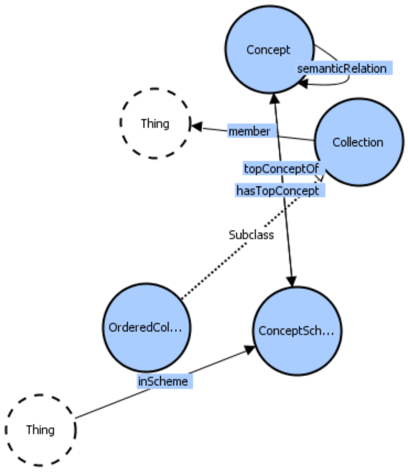

**Figure 2.** Visualization of the doacc with VOWL

illustrated in blue. And the children in the blue classes are depicted in purple. Below the classes, in the rectangle, are the properties of the classes.

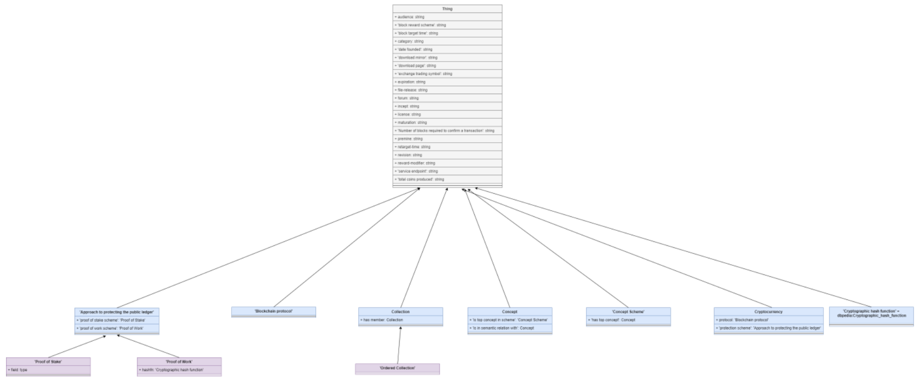
Thing
+ audience: string
+ 'block reward scheme': string
+ 'block target time': string
+ category: string
+ 'date founded': string
+ 'download mirror': string
+ 'download page': string
+ 'exchange trading symbol': string
+ expiration: string
+ file-release: string
+ forum: string
+ incept: string
+ license: string
+ maturation: string
+ 'Number of blocks required to confirm a transaction': string
+ premine: string
+ retarget-time: string
+ revision: string
+ reward-modifier: string
+ 'service endpoint': string
+ 'total coins produced': string
'Approach to protecting the public ledger'
+ 'proof of stake scheme': 'Proof of Stake'
+ 'proof of work scheme': 'Proof of Work'
'Blockchain protocol'
Collection
+ has member: Collection
Concept
+ 'is top concept in scheme': 'Concept Scheme'
+ 'is in semantic relation with': Concept
'Concept Scheme'
+ 'has top concept': Concept
Cryptocurrency
+ protocol: 'Blockchain protocol'
+ 'protection scheme': 'Approach to protecting the public ledger'
'Cryptographic hash function' = dbpedia:Cryptographic_hash_function
'Proof of Stake'
+ field: type
'Proof of Work'
+ hashfn: 'Cryptographic hash function'
'Ordered Collection'

**Figure 3.** Visualization of the doacc with VOWL

## 3. Summary

This article first introduces some ontology-UML conversions. Then the Description of a Cryptocur-rency (DOACC) [1] ontology was presented and converted to the RelOntoUML [2] model. I downloaded the ontology from github. The model contains some classes, and the 'Thing' class also contains several properties. RelOntoUML [2] is a relational model made from a combination of ontol-ogy and UML model, combining the advantages of the three models.

## 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] *Description of a Cryptocurrency (DOACC)* https://github.com/DOACC/doacc (Accessed: 2022.05.14.)

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

[3] Paredaens, J., De Bra, P., Gyssens, M., & Van Gucht, D. (1989). *The structure of the relational database model* Springer Science & Business Media. https://doi.org/10.1007/978-3-642-69956-6

[4] Lin, H. K., & Harding, J. A. (2007). *A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration.* Computers in Industry, 58(5), 428-437. https://doi.org/10.1016/j.compind.2006.09.015

[5] UML, O. (2001). *Unified modeling language.* Object Management Group, 105.

[6] Ha, Y., & Lee, R. (2006, June). *Semantic Web Service Modeling using UML for e-business environment.* In Seventh ACIS International Conference on Software Engineering, Artificial In-telligence, Networking, and Parallel/Distributed Computing (SNPD'06) (pp. 368-374). IEEE. https://doi.org/10.1109/snpd-sawn.2006.62

[7] Evermann, J. (2009). *A UML and OWL description of Bunge's upper-level ontology model.* Software & Systems Modeling, 8(2), 235-249. https://doi.org/10.1007/s10270-008-0082-3

[8] Pătraşcu, A. (2015). *Comparative analysis between OWL modelling and UML modelling.* Pe-troleum-Gas University of Ploiesti Bulletin, Technical Series, 67(2).

[9] *VOWL* http://vowl.visualdataweb.org/ (Accessed: 2022.05.14.)