Implementation of the Transformation of an OntoUML model into Its Realization in a Relational Database

FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Author: Ing. Jakub Jabůrek ⟨jaburjak@fit.cvut.cz⟩ | Supervisor: Ing. Zdeněk Rybola, Ph.D.

Download OpenPonk

1. Motivation

The *model-driven development* approach to software engineering proposes that the engineer should create a model prior to writing source code. This provides *abstraction* — data structures, processes, entities, interactions, etc. can be thought out in advance without the engineer being burdened by implementation details.

Ideally, the model is so precise that it can be (at least partially) automatically converted to a working source code. This approach brings two main benefits:

1. The amount of human work is reduced by automating the transformation of the model to source code.
2. The model can act as documentation, and the resulting software system does not deviate from it, as it was generated from it.

2. Problem Statement

OntoUML is used in the construction of precise domain models; we seek to implement **a fully automatic transformation** of such models into their realization in a relational database, in the form of a sequence of **SQL statements** for Oracle Database that initialize the database schema.

Although methods for the transformation of domain models to relational databases are well-known, we focus on **preserving the precision and integrity constraints** specified by OntoUML.

3. State of the Art

OntoUML is one of the established model notations. It is an ontologically well-founded profile of UML. By assigning stereotypes to types and relationships, precise semantic meaning of the entire model can be established. OntoUML specifies various integrity constraints corresponding to the stereotypes. [1]

Oracle Database is of the most used implementations of relational database management systems. It uses SQL to both define the database schema and manipulate the data within.

OpenPonk is a modeling tool that supports authoring OntoUML models [2].

The method for the transformation of an OntoUML model to SQL was originally proposed by Dr. Zdeněk Rybola [3]. In our work, we review their approach and implement it as an extension of OpenPonk.

5. Results

In our work, **we successfully implemented** an extension of OpenPonk that can **automatically transform any OntoUML model to SQL**.

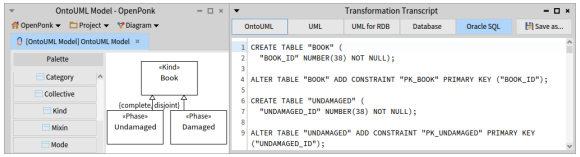

Figure 2: OpenPonk user interface with transformation result

The implementation is based on the approach proposed by Dr. Rybola [3], which was revised for OntoUML version 2. Our implementation supports all proposed transformation options, and **preserves all integrity constraints** specified in the input model.

We successfully tested the resulting software on a large real-world conceptual model and confirmed it is able to correctly transform such model.

Additionally, a generic transformation framework was created (written in Pharo), which can be used to implement transformations between any model notations.

4. Transformation Process

The transformation of the input OntoUML model to SQL we implemented in OpenPonk is done in three stages with two intermediate models, as illustrated by the diagram in figure 1.

1. OntoUML
   ↳ 2. UML
      ↳ 3. Relational Data Model
         ↳ 4. SQL

Figure 1: Transformation stages

The resulting sequence of SQL statements can be exported and immediately used to initialize an empty database. The intermediate UML and RDB models can be inspected in OpenPonk.

The OntoUML model can define various constraints, some may not be natively representable in the transformed models. In UML and RDB models, those constraints are represented by OCL invariants and conditions. In the resulting SQL, they are realized by triggers that prevent data insertion and modification that would violate the defined constraints.

Some model constructs can be transformed in multiple different ways — in those cases, the user chooses which transformation approach will be used. Similarly, the user can apply optimizations that reduce the complexity of the transformed model.

6. References

[1] GUIZZARDI, Giancarlo, et al. *Towards Ontological Foundations for Conceptual Modeling: The Unified Foundational Ontology (UFO) Story*. Applied Ontology. 2015, vol. 10, no. 3-4. DOI: 10.3233/AO-150157.

[2] FIT CTU. *OpenPonk modeling platform*. 2023. https://openponk.org/.

[3] RYBOLA, Zdeněk. *Towards OntoUML for Software Engineering: Transformation of OntoUML into Relational Databases*. 2017.