

SQL-PL4OCL : An automatic code generator from OCL to SQL Procedural Language

Marina Egea
Minsait (by Indra)
Madrid, Spain
Email: msegea@minsait.com

Carolina Dania
IMDEA Software Institute
Madrid, Spain
Email: carolina.dania@imdea.org

I. ABSTRACT

Design models are widely spread as core artifacts in software engineering. Yet, a key problem is how to fulfill correctly these blueprint specifications when code components are developed. The best possible scenario occurs when a source modeling language can be perfectly linked to a target language of election. Namely, a well defined mapping bridges the gap between the source and the target language. Otherwise, manual encoding of the system design is cumbersome and error prone. In this setting, we introduce a SQL-PL¹ code generator for OCL expressions that, in contrast to other proposals, is able to map OCL iterate and iterator expressions thanks to our use of stored procedures.

More in detail, our source language is the Object Constraint Language (OCL), which nowadays is an ISO standard used to express constraints and queries in a textual notation on UML models. Our target language is the procedural language (PL) extension to the Structured Query Language (SQL). SQL is a special-purpose programming language designed for managing data in relational database management systems (RDBMS). The purpose of PL for SQL is to combine database language and procedural programming language.

Although SQL is also an ISO standard, different RDBMS implement certain syntactic variations to the standard SQL notation. Thus, we had to adapt the implementation of our mapping to each of them. As implementation targets we selected MariaDB, PostgreSQL, and MS SQL Server. MariaDB and PostgreSQL were selected because they are open source and widely used by developers. MS SQL server was selected to be able to compare evaluation time from open source to commercial RDBMS.

A variety of applications arises for a mapping from OCL to SQL expressions. Among others, there are three prominent types. These are i) evaluation of OCL expressions (analysis queries and metrics) on large model's instances, ii) identification of constraints during data modeling that have to be checked as integrity constraints on actual data; iii) automatic code generation from models. Indeed, our implementation was used as a key component of a toolkit that automatically generated ready-to-deploy web applications for secure data

management from design models. Our component mapped and evaluated OCL constraints specified within authorization policies.

Our code generator is defined recursively over the structure of OCL expressions and it is implemented in the SQL-PL4OCL tool that is publicly available at [1]. The seminal work of the mapping presented here can be found in [2], [3]. The key idea that enables the mapping from OCL iterator expressions to iterative stored procedures remains the same, but the work detailed in [4] introduces a novel mapping from OCL expressions to SQL-PL stored procedures.

In the novel mapping we have taken design decisions which have facilitated the recursive definition of the code generator and simplified its definition. These decisions have also helped to significantly decrease the time required for the evaluation of the code generated. Regarding semantics, the new mapping is able to deal properly with the three-valued evaluation semantics of OCL. In addition, our original work and implementation was intended only for the procedural extension of MySQL, while our new definition eased the implementation of the mapping into other relational database management systems. In turn, we can now evaluate the resulting code using different RDBMS, which permits us to widen our discussion regarding efficiency in terms of *evaluation-time* of the code produced by SQL-PL4OCL tool.

REFERENCES

- [1] C. Dania and M. Egea, "SQLPL4OCL tool," 2016, <http://software.imdea.org/dania/tools/sqlpl4ocl>.
- [2] M. Egea, C. Dania, and M. Clavel, "MySQL4OCL: A stored procedure-based MySQL code generator for OCL," *ECEASST*, vol. 36, 2010. [Online]. Available: <http://journal.ub.tu-berlin.de/index.php/eceasst/article/view/445>
- [3] C. Dania, "MySQL4OCL: Un compilador de OCL a MySQL," 2011, master thesis. Universidad Complutense de Madrid. <http://eprints.ucm.es/13800>.
- [4] M. Egea and C. Dania, "SQL-PL4OCL: an automatic code generator from OCL to SQL procedural language," *Software & Systems Modeling*, May 2017. [Online]. Available: <https://doi.org/10.1007/s10270-017-0597-6>

¹Please, notice that in this paper SQL-PL stands simply for SQL Procedural Language. It is not bound to any proprietary PL dialect.