

A Type-Safe Object-Oriented Solution for the Dynamic Construction of Queries

Peter Rosenthal

*Fernuniversität Hagen, Department PI 1, Prof. Schlageter
peter_rosenthal@web.de*

1. Introduction

Many object-oriented applications use large numbers of structurally different database queries. With current technology, writing applications that generate queries at runtime is difficult and error-prone.

FROQUE, our framework for object-oriented queries, provides a secure and purely object-oriented solution to access relational databases. As such, it is easy to use for object-oriented programmers and with the help of object-oriented compilers it guarantees that queries formulated in the object-oriented world at execution time result in correct SQL queries.

Thus, FROQUE is an improvement over existing database frameworks such as Apache OJB¹, the Object Relational Bridge, which are not strongly typed and can lead to runtime errors.

2. FROQUE and its SQL Model

The basic building blocks of FROQUE are its model of SQL, its generators of database schema models and its “lazy” code generation technique.

The SQL Model contains classes representing primitive database types and SQL structure.

The first group includes classes such as `DB_Number` with the methods `plus`, `minus`, `greater` and `smaller`. The second group is made up of classes such as `Query` with the method `execute`.

3. Model of Database Schema

In addition to the schema of user-defined classes that are mapped to the database schema, FROQUE holds a corresponding schema of database classes which applications use to define representations of tuple variables.

Database class names are marked with the prefix `DB_` and their primitive attributes are declared using the primitive types of the SQL model such as `DB_Number`.

FROQUE’s generators build database schemata and database classes from object-relational mappings.

4. The Generation Technique MMI

The methods of the models, such as `plus` of `DB_Number`, are not executed in order to directly calculate a result. Instead, each method invocation is memorized with its parameters in its return object for later code generation. This novel lazy technique is referred to as MMI, memorized method invocation.

When MMI return objects are used as input of other MMI-based methods, syntax trees of arbitrary depth are built. In this sense MMI creates modules that allow the definition of recursive structures as in nested queries, constraint compositions and complex constraints.

By means of the MMI concept, code can be generated according to different contexts. Context-sensitive rules can be applied, e.g. `FROM`-clauses are generated with regard to database class instances used in a query.

5. Conclusion

FROQUE, to our knowledge, is the first type-safe support for the dynamic generation of queries. It has been implemented in Java and proved its merits in the CUBER² project, where it provides highly flexible database access to a schema of over 60 tables. It is well

¹ see <http://db.apache.org/ojb/>

² The CUBER project is supported by the European Community in the Fifth Framework IST programme (Contract IST-1999-10737). This document does not represent the opinion of the European Community nor is the European Community responsible for it.

suited for iterative search processes in which queries are constantly refined.