UB2DB: Rodin Plug-in for Automated Database Code Generation

Ahmed Al-Brashdi, Michael Butler, and Abdolbaghi Rezazadeh

University of Southampton, Southampton, UK
{azab1g14,mjb,ra3}@ecs.soton.ac.uk

1 Introduction

Correct design of database systems is an important aspect of software engineering. It requires a verifiable approach in order to eliminate any ambiguity and inconsistency in the database requirements. While the design process can be achieved using formal modelling, it needs a tool support to automatically generate the database code from the formal model such as Event-B [1] and UML-B [7]. While previous work by Wang and Wahls [8] provide such a facility as a Rodin plug-in, their tool evaluation in [6] identifies major performance issues and further evaluation highlighted issues with preserving database integrity as in [3]. This motivates us to build a tool that generates a verified implementation of UML-B and Event-B model that covers building the databases with their constraints as well as the operations that manipulate the database while preserving those constraints.

2 Tool development

We built a tool, UB2DB [5] (UML-B to DataBase), for the Rodin [2] platform that provides an automatic generation of SQL code from a model defined in UML-B in the Rodin platform. The tool complements a database modelling methodology presented in [4] to generate the code from different refinement levels. The generated SQL will create the relational database structure using *create* and *alter* commands and provide procedures that populate and manipulate the data. As the tool generates the procedures that will be embedded in the database, the developer can then call these procedures from any supported programming language. This approach does not restrict the developer to one language or a single platform also. The procedure can be called from a desktop application, web application or a mobile application without compromising the consistency of the database. This increase the productivity of the developer by providing an automated and verified database code. Developers will model their system in UML-B and verify it in the Rodin platform to detect any inconsistency or ambiguity. Then they will use UB2DB to translate the verified model to SQL code. UB2DB provides a refinement approach for generating the database structure in which a refined model in UML-B results in an extension to the database generated from the abstract model using the *alter* command in SQL. However, the notion of refining operations does not exist in the database procedural language, hence, the extension is applicable to the structure only.

3 Tool evaluation

UB2DB has been evaluated and executed against two case studies for student enrollment system and car sharing system. The generated code from the case studies was successfully imported in Oracle DBMS and evaluated against the database constraints. Evaluating the performance of inserting 10000 records using the generated code shows that our code performed around 21% slower than a hand-written code. This shows that its performance is not very different from a hand-written one. Further evaluation and optimisation should improve the efficiency and performance of UB2DB. While the tool successfully generates the code as evaluated using different case studies, it only translates from the UML-B model. Any specifications that are specified in Event-B model outside UML-B will not be treated by UB2DB.

4 Tool limitations

While UB2DB generates the SQL and stored procedures following our translation rules successfully, it does have some limitations in its current version.

One limitation is that the tool does not traverse back in the refinement chain. For a machine *m1* that refines *m0*, UB2DB will not see *m0* if it was executed for *m1*. This limitation has an effect on inheritance translation as the super type arrow does not appear in the refinement in iUML-B and UB2DB will not see the inheritance link which means that the sub classes will not be associated with the super class if translating to class table inheritance pattern.

As UB2DB takes the UML-B model for translation and not the Event-B, any specifications that are specified in Event-B model outside UML-B will not be treated by UB2DB.

Only expressions and predicates that are used in our incremental model approach [4] are translated by the tool. If the modeller specified operations differently, they will not be translated by UB2DB and the modeller will not get a warning or an error in the current version.

5 Conclusion and future work

UB2DB is a plugin for Rodin that provides an automated database code generation from UML-B models. It generates SQL to create and alter the structure of a relational database and stored procedures to populate and modify its data. The plugin increases the productivity of developers by automating the implementation of the database while developers can focus on the rigorous specifications of the system. We applied the plugin on different case studies and evaluated the generated code integrity and performance. The result shows that UB2DB can be used to generate a verified database implementation from a UML-B class diagram. This should increase the usability and efficiency of Rodin for software developers as it provides the facility to generate the database code which is a critical part of any data-intensive software.

The future improvement includes expanding the supported notations to cover more complex operations. An interesting improvement is to support the translation to different database models along with the relational, such as NoSQL. UB2DB could be improved further to support database extension where an existing database is loaded by the tool and extended using UML-B, then translated back with the extension by UB2DB.

References

1. Jean-Raymond Abrial. *Modeling in Event-B: system and software engineering*. Cambridge University Press, 2010.
2. Jean-Raymond Abrial, Michael Butler, Stefan Hallerstede, Thai Son Hoang, Farhad Mehta, and Laurent Voisin. Rodin: an open toolset for modelling and reasoning in Event-B. *International journal on software tools for technology transfer*, 12(6):447–466, 2010.
3. Ahmed Al-Brashdi. Translating Event-B to database application. Master's thesis, University of Southampton, 2015.
4. Ahmed Al-Brashdi, Michael Butler, and Abdolbaghi Rezazadeh. Incremental database design using UML-B and Event-B. In Proceedings Joint Workshop on *Handling IMPlicit and EXplicit knowledge in formal system development (IMPEX)* and *Formal and Model-Driven Techniques for Developing Trustworthy Systems (FM&MDD),* Xi'An, China, 16th November 2017, volume 271 of *Electronic Proceedings in Theoretical Computer Science*, pages 34–47. Open Publishing Association, 2018.
5. Ahmed Al-Brashdi, Michael Butler, Abdolbaghi Rezazadeh, and Colin Snook. Tool support for model-based database design with Event-B. In *FM&MDD Workshop at ICFEM 2016*, pages 1–7, 2016.
6. Néstor Catano and Tim Wahls. A case study on code generation of an ERP system from Event-B. In *Software Quality, Reliability and Security (QRS), 2015 IEEE International Conference on*, pages 183–188. IEEE, 2015.
7. Colin Snook and Michael Butler. UML-B and Event-B: An integration of languages and tools. In *Proceedings of the IASTED International Conference on Software Engineering*, SE '08, pages 336–341, Anaheim, CA, USA, 2008. ACTA Press.
8. Qi Wang and Tim Wahls. Translating Event-B machines to database applications. In *Software Engineering and Formal Methods*, pages 265–270. Springer, 2014.