

QVTo model transformations : assessing and improving their quality

Citation for published version (APA):

Gerpeide, C. M., Schiffelers, R. R. H., & Serebrenik, A. (2014). QVTo model transformations : assessing and improving their quality. *ERCIM News*, 99, 32-33.

Document status and date:

Published: 01/01/2014

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

valid product behaviours can be generated explicitly and the resulting specifications can be verified against the same logic property (this is surely less efficient than direct verification, but it makes it possible to identify precisely why the original property failed over the whole family). Experimentation with VMC is on-going in the context of the EU FP7 project QUANTICOL. To date, only a small version of the bike-sharing case study from QUANTICOL has been considered but more effort is needed to evaluate VMC on more realistically sized problems.

Link: <http://fmt.isti.cnr.it/kandisti>

References:

- [1] M.H. ter Beek, A. Fantechi, S. Gnesi, and F. Mazzanti. “A state/event-based model-checking approach for the analysis of abstract system properties.” *Science of Computer Programming* 76(2):119-135, 2011
<http://dx.doi.org/10.1016/j.scico.2010.07.002>
- [2] A. Fantechi, S. Gnesi, A. Lapadula, F. Mazzanti, R. Pugliese, and F. Tiezzi. “A logical verification methodology for service-oriented computing.” *ACM*

Transaction on Software Engineering and Methodology 21(3):16, 2012.
<http://doi.acm.org/10.1145/2211616.2211619>

[3] F. Mazzanti, G. O. Spagnolo, and A. Ferrari. “Designing a deadlock-free train scheduler: A model checking approach.” *NASA Formal Methods, LNCS 8430*, Springer, 2014, 264-269.
http://dx.doi.org/10.1007/978-3-319-06200-6_22

Please contact:

Franco Mazzanti, ISTI-CNR, Italy
E-mail: franco.mazzanti@isti.cnr.it

QVTo Model Transformations: Assessing and Improving their Quality

by Christine M. Gerpheide, Ramon R.H. Schiffelers and Alexander Serebrenik

Applying a unique bottom-up approach, we (Eindhoven University of Technology (The Netherlands)) worked collaboratively with ASML, the leading provider of complex lithography systems for the semiconductor industry, to assess the quality of QVTo model transformations. This approach combines sound qualitative methods with solid engineering to proactively improve QVTo quality in a practitioner setting.

Model-driven engineering (MDE) can be used to develop highly reliable software which offers a range of benefits from systems analysis and verification to code generation. In MDE, system models are created by domain experts and then transformed into other models or code using model transformations. One of the languages used for writing these model transformations is QVT Operational Mappings (QVTo) which was specified in the 2007 Object Management Group (OMG) standard for model-to-model transformation languages. QVTo is regularly used by both academics and industry practitioners, including ASML, the leading provider of complex lithography systems for the semiconductor industry. Currently ASML has more than 20,000 lines of QVTo code, supporting more than a hundred model transformations.

Despite its widespread use, however, QVTo is a relatively new language. For general-purpose languages, developers have had time to share best practices and establish a number of standard reference points against which the quality of a piece of code can be judged. These are yet to be developed for QVTo.

Moreover, QVTo has a large amount of language-specific constructs which are not available in general-purpose languages or even in other model-to-model transformation languages. In fact, QVTo specifications have been described by some as “rather voluminous” and even “fantastically complex”. Therefore, it is unclear whether established intuitions about code quality apply to QVTo and a lack of standardized and codified best practices is recognized as a serious challenge in assessing their transformation quality.

In a response to this challenge, ASML and Eindhoven University of Technology have joined in an ongoing collaboration to investigate the quality of QVTo transformations. In addition to assessing the quality of a transformation, this project is also seeking to promote the creation of higher-quality transformations from their inception, improving quality proactively [1].

To achieve this goal, a bottom-up approach was used which combined three qualitative methodologies. To begin, a broad exploratory study which included the analysis of interviews with

QVTo experts, a review of the existing literature and other materials, and inspection were completed. Then, a QVTo quality model was developed to formalize QVTo transformation quality: this model consisted of high-level quality goals, quality properties, and evaluation procedures. The quality model was validated using the outcomes from a survey of a broad group of QVTo developers in which they were asked to rate each model property on its importance to QVTo code quality.

Many of the model properties recognized as important for QVTo transformation quality are similar to those in traditional languages (e.g., small function size). However, this analysis also highlighted a number of properties which are specific to QVTo or other model transformation languages. For instance, we found that the following QVTo-specific properties were considered important for quality: the use of only a few black boxes, few queries with side effects, little imperative programming (e.g., for-loops) and small init sections. Deletion using trash-bin patterns was also found to be beneficial for per-

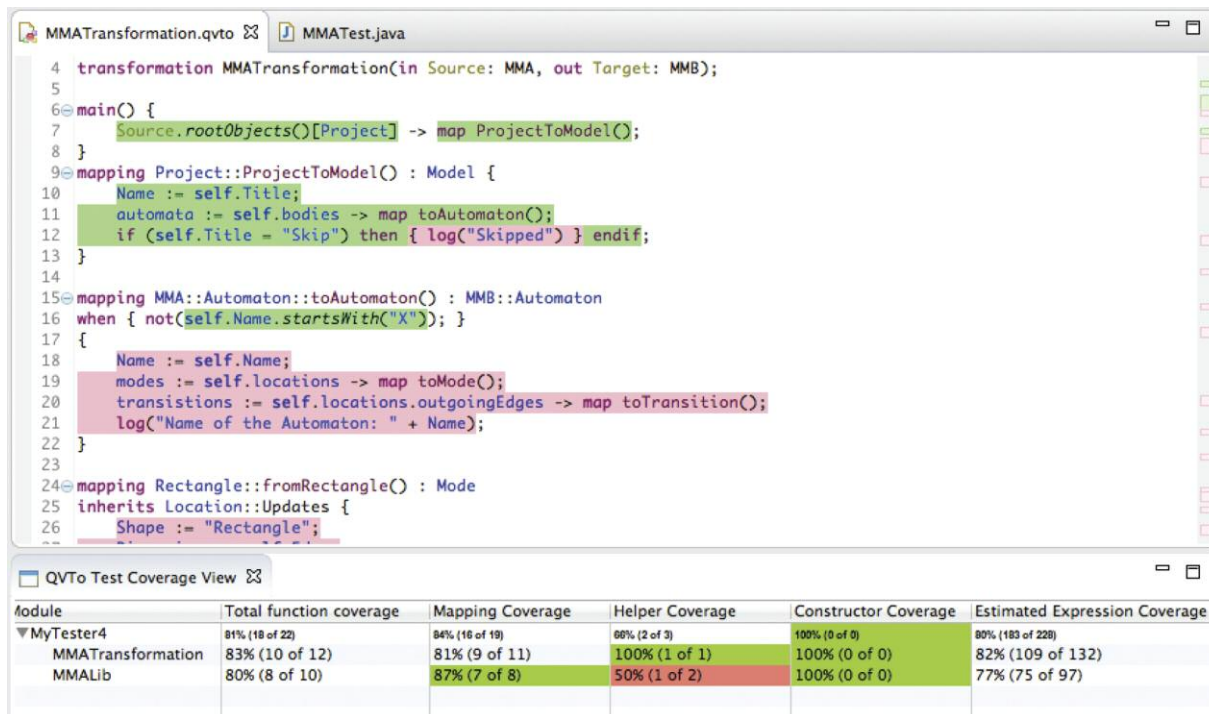


Figure 1: The coverage tool highlights the covered and not covered parts of the QVTo transformation as well as reports coverage-related statistics

formance as it has high test coverage in relation to functionality [2].

Test coverage also emerged as a key issue in the expert interviews, with every interviewee highlighting the lack of a test coverage tool. Consequently, we prioritized this demand and created a test coverage tool for QVTo (Figure 1). Implemented as an Eclipse plugin, the tool helps QVTo developers to create higher-quality transformations from the start: at the developers' request, the tool reports coverage percentages for different units of QVTo transformations (e.g., mappings, helpers, constructors), as well as visualizing expressions covered and not covered by the tests. During the seven week study period, the tool was used 98 times, resulting in execution of 16,714 unit tests. In addition to assisting with debugging issues, the tool was also used by one developer to prepare a user story, a description of how to complete the specific task of an agile sprint. When preparing the user story, the developer ran the entire test suite and looked at the coverage for the modules he knew would be affected by the task to be completed. He then noted exactly which places in the modules were not covered by the test suite by inspecting the coverage overlay. Then, as part of the list of steps for the feature, he added an additional step stating that before

implementation of the feature can begin, additional tests must be written to cover the untested parts. The developers also stated the tool easily identifies dead code. The coverage tool has also been added to the development team's "Way of working" document, making it an official part of their development process.

As a side product of our research, three patches were submitted, accepted and integrated into the Eclipse QVTo core engine which was released with Eclipse Luna on June 25, 2014. Together, these patches make it possible for QVTo interpreters to easily access the test coverage tool, as well as future tools, such as an integrated profiler [3].

References:

- [1] C.M. Gerpheide: "Assessing and Improving Quality in QVTo Model Transformations", M.Sc. thesis, Eindhoven University of Technology, 2014, http://phei.de/docs/academic/Gerpheide_QualityQVToTransformations.pdf
- [2] C.M. Gerpheide, R.R.H. Schiffelers, A. Serebrenik: "A Bottom-Up Quality Model for QVTo", QUATIC 2014, Portugal, 2014)
- [3] C.M. Gerpheide et al.: "Add hooks for contributing 3rd-party visitor decorators", https://bugs.eclipse.org/bugs/show_bug.cgi?id=432969

Please contact:

Alexander Serebrenik
 Eindhoven University of Technology,
 The Netherlands
 Tel: +31402473595
 E-mail: a.serebrenik@tue.nl