Test-driven verification of model transformations

Péter Hudák and László Lengyel

Model-driven engineering provides a numerous number of fast and efficient solutions on the field of software engineering. The focused problem is decomposed to different abstraction levels. This approach helps understanding and modeling the problem to be solved. As a result, software models represent the essence of the actual problem and show the critical aspects of it.

Different planning and development phases of a software project produces models belonging to different abstraction levels. Model processors are dedicated to map the different abstraction levels. A set of rules what are applied in a defined order on source models and result target models are called transformation. Model-driven engineering utilizes this method widely. One of the current significant fields is the software development for cyber-physical systems (next generation of embedded systems).

Both syntactically and semantically appropriate outputs should be generated by model processors. This means that the resulted software artifact has to contain the correct semantic information. This is the point where test-driven verification of model transformations gives proper feedback to check the correctness of model processors. Testing of model transformations is a challenging and complex problem. Design errors are usually the main source of defects in model transformations. Complexity of software artifacts and models make searching for bugs and errors even harder. The goal of test-driven method is to test model transformations by automatically generating right input models. Transformations execute these input models. Semantic correctness verification is made after the execution via comparing the input and output models. The method is hard to be fully automatized because additional information about the models are usually required by the process. Models tend to change continuously during software development so model transformation tests should adapt to these changes. Model transformations should conform to predefinitions so coherent test cases could be generated in this way. This set of tests is handled then as unit tests.

Visual Modeling and Transformation System (VMTS) is a domain-specific modeling and metamodeling framework, which supports defining domain-specific languages, editing models, furthermore, defining and performing model-to-model and model-to-code transformations. The latest version is currently under development, we are working on the extension of the model transformation engine. A new automatic test generation solution will be added that supports test-driven verification of model transformations. This framework generates a set of input models. These input models function as unit tests therefore unit tests could be generated for model transformations which fill predefinitions. In this paper we give an overview about the test-driven verification methods of model transformations, furthermore, we demonstrate the test generation engine of VMTS 4.