Test Suite Design for Code Generation Tools

Ingo Stürmer, Mirko Conrad
DaimlerChrysler AG, Research and Technology, Berlin, Germany

Motivation
- The use of executable models in the development of embedded systems has become increasingly important.
- New approaches allow the automatic generation of compact controller code directly from the software-model with so-called code generators.
- At present, code generators do not possess the same quality characteristics as C or ADA compilers which have been proven in use.
- The use of test suites, which make it possible to check compilers systematically, is also a promising approach for code generators.

Test Case Derivation

Stimulate all possible simulation pathways through a given model by applying automated structural testing.
⇒ Find a selection of input data which achieve full model coverage!

A similar approach is followed on code level.
⇒ Create a second set of input data which achieves full code coverage!

The test data created are merged.

Both the model and the code are stimulated with the resulting amount of input data.

The model and the code’s outputs are compared. Different results are due to:
⇒ incorrect code generator
⇒ problem with the tool chain (e.g. compiler, linker)
⇒ faulty test model
⇒ incomplete specification

Summary
• General approach for the systematic verification of code generators
• Seamless test case derivation (process) based on formal specification (i.e. graph transformation rule)
• A method is suggested to create test input data for model and code using automated structural test
• Results are suitable for making general assumptions about whether or not the code generator’s behaviour is correct