Model driven testing of component based systems

Gábor Bátori, Zoltán Theisz, and Domonkos Asztalos

The growing demand of the telecommunication market for complex systems cannot be easily satisfied without new development paradigms. Model Driven Architecture (MDA) [1] offers a good way to achieve the desired complexity management. However, MDA uses UML as a notation and in many cases UML is too complicated to use because its philosophy does not match to the one of the modelled system. In that case a well defined modeling language, that is, a Domain Specific Language for the problem is more effective. Domain specific modelling requires at least three ingredients to be well-defined:

- Metamodel that defines which concepts have relevance in the problem domain, how they are related to each other and in which manner they can be put together correctly to provide consistency.
- Model that uses the concepts defined in the metamodel and establishes an instance set of these concepts which describes the problem.
- Model translators that add semantical meanings to the metamodel concepts. A translator could be e.g. a model-to-code translator which generates source code from the model or a model-to-model translator which translate a model describing one aspect of the system to another model describing different aspect. An example model translator is the one which creates the architecture model of the modelled system from the functional model.

The modelling tool we use that enables the domain specific modelling of systems is the Generic Modeling Environment (GME) [2]. GME supports metamodelling and metamodel aware modelling of the application domain. Furthermore, it provides two ways of defining model transformation; an interpreter based and a more formalised graph transformation based method. Increased customer requirements for building highly reconfigurable and reusable system can be handled with an appropriate system architecture. To control these demands component system architecture is necessary that enables reusable modularized services to be composed, interconnected, configured and deployed to create applications rapidly and robustly in dynamically changing distributed environments. To fulfil all requirement of the telecommunication software development we created the ErlCOM system [3], that is, an innovative combination of the beneficial aspects of component-based programming and model based development. There is a strong need for effective testing of model based applications but there is no commonly accepted method. Our main goal is to create a generic framework for testing generated from the metamodel and the model of the application. The generated framework can be used for writing test cases manually but the main goal is to serve as a basis for different automatic test generation algorithm plug-ins. Moreover, since all plug-in test generation algorithms rely on the services of the underlying test framework, therefore, they provide comparable results.

References

[1] R. Soley. Model Driven Architecture: An introduction, `http://www.omg.org/mda`.

[2] G. Karsai, J. Sztipanovits, A. Ledeczi, and T. Bapty. Model-integrated development of embedded software, *Proceedings of the IEEE*, vol. 91, pp. 145-164, January 2003.

[3] G. Batori, Z. Theisz, and D. Asztalos. Robust Reconfigurable Erlang Component System, *Erlang User Conference*, Stockholm, Sweden, November 2005.