Describing interaction relationships in SMV language

Zsolt Borsi

During the phases of the software development process Unified Modeling Language (UML) helps in managing the complexity of problems by separately describing different aspects of the system under development. The created intermediate artifacts are mainly UML diagrams representing a particular view of the system. Since UML models represent specification documents, they provide ideal basis for validation and verification. Model checking is a successful method to prove that a given system meets its specification [3, 4]. It has been found especially useful when the correctness of the system requires a formal approach, for instance when the system under consideration is a safety-critical system.

In this paper the translation of a particular scenario-based model into SMV language (the input language of the NuSMV model-checker tool [5]) will be investigated. With the introduction of Interaction Overview Diagrams (IODs) in UML2.0, relationships between interaction diagrams can be explicitly defined in a standard way. IODs are a graphical representation of relationships between UML interaction diagrams and the control flow passing between them, where each node in the activity graph is a reference to an IOD. This paper takes into account additional relationships, which are not available in IODs. These extension version of Interaction Overview Diagram (namely EIOD) introduced by Whittle [1] includes interruption, continuation, concurrency and the notion of negative scanario. Whittle introduced a way of structured specification as well. A use case chart specifies the scenarios for a system's use cases in such a way that each of the EIOD nodes is refined by a set of use case nodes at the lower level of the use case chart. In fact, in this paper the description of a use case chart in SMV language will be presented. The rationale behind that idea is that the module concept of SMV provides a mean for describing a hierarchical construct and SMV modules (like scenarios defined by interaction diagrams) operate parallel to each other.

Acknowledgements

This work was supported by TÁMOP-4.2.1/B-09/1/KMR-2010-0003

References

[1] J. Whittle, P.K. Jayaraman. Synthesizing hierarchical state machines from expressive scenario descriptions, *ACM Transactions on Software Engineering and Methodology (TOSEM), v.19 n.3, pp.1-45, 2010.*

[2] E.M. Clark, W. Heinle. Modular Translation of Statecharts to SMV, *Technical Report CMU-CS-00-XXX, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213., 2000.*

[3] E.M. Clark, O. Grumberg, D.A. Peled. Model Checking, *The MIT Press, Cambridge, MA, 2000.*

[4] O. Kupferman, M.Y. Vardi, P. Wolper. Module checking, *Information and Computation, 164(2), pp.322-344, 2001.*

[5] NuSMV Model Checker Home Page, *http://nusmv.fbk.eu*