SV_tL: System Verification through Logic
Tool Support for Verifying Sliced Hierarchical Statecharts

Sara Van Langenhove*, Albert Hoogewijs

Department of Pure Mathematics and Computer Algebra
Ghent University, Belgium
Sara.VanLangenhove@UGent.be, Albert.Hoogewijs@UGent.be

The best known example of an automata based approach to software verification is model checking. It has been shown to be effective in finding subtle errors in software designs, and thus substantially reducing cost and development time. Model checking object-based systems designed in UML necessitates the transformation of the statechart diagrams (describing the behavior of the objects) to a verification model. If the entire state space (design space) has to be searched in order to establish correctness, model checking is applicable only to small portions of a design. Through slicing however, model checking can be applied to larger designs. Slicing statechart models focusses on the identification and elimination of states and transitions that are irrelevant for the property under verification. It would thus be very useful to have a tool that automates this proces and allows the integration of a formal design method and verification of the model in an early phase of the development of the system.

The SV_tL (System Verification through Logic) framework is the core of a slicing-based verification environment for improving the quality of systems whose behavior is designed using UML statechart models. Specifically, the tool acts as a front end to the model checker SMV (Cadence Version). It takes as input a statechart model and a temporal logic property. The model is then sliced according to that property and the sliced model is translated into the input of the model checker. Additionally, SV_tL has an assistant that guides the user in writing properties to be verified using temporal logic. Eventually the tool presents a counterexample in the statechart formalism. This all happens without intervention of the user. The framework only assumes that the analyst knows UML and the system to be developed. It is worthwhile to note that SV_tL has been designed and built entirely in UML and Java.

We present an overview of the SV_tL software architecture together with some design decisions and the underlying slicing-based formal methodology. Special attention will be given to the slicing approach. Slicing positively influences the complexity of the verification approach, based on removing pieces of the model that are not of interest during verification. Some slicing algorithms have been proposed for statecharts, but they were not able to handle orthogonal regions efficiently. We optimize such an algorithm by removing false dependencies, relying on the broadcasting mechanism between different parts of the statechart model.

*Funded by Ghent University (BOF/GOA project B/03667/01 IV1)