

From Business Modeling to Verified Applications

Christian Ammann and Stephan Kleuker
Hochschule Osnabrück, University of Applied Sciences
Postbox 1940, 49009 Osnabrück, Germany
{c.ammann, s.kleuker}@hs-osnabrueck.de

Elke Pulvermüller
University of Osnabrück
Albrechtstr. 28, 49076 Osnabrück, Germany
elke.pulvermueller@informatik.uni-osnabrueck.de

Abstract: UML activity diagrams can be used to model business processes which are implemented in a software project. It is a worthwhile goal to automatically transform at least parts of UML diagrams into software. Automated code generation reduces the total amount of errors in a software project but the model itself can still violate specified requirements. A quality improvement is the usage of a model checker which searches through the whole state space of model and checks whether all requirements are met. A model checker requires a formal description of a model for a complete verification. Activity diagrams often describe processes informally which is difficult to verify with a model checker. We therefore propose the transformation of activity- to statechart diagrams which allow a more detailed and formal description. Several algorithms exist to map UML statecharts into a model checker input language for a successful formal verification. Afterwards, the model checker searches through the whole state space of a statechart and therefore has to store each state in memory. UML statecharts can reach a high degree of complexity which is problematic for a complete state space traversal because the total amount of available memory is exhausted. Accordingly, we present the domain specific language UDL (UML Statechart Modeling Language) and a transformation from UDL into the Spin model checker input language Promela. UDL contains features for property preserving abstraction which reduces the models state space and therefore the memory consumption of a model checker. Furthermore, we introduce an optimisation technique for the transformation process from UDL to Promela which focuses on a reduced model checker run-time. A case study with a movement tracking system demonstrates how our approach could significantly reduce the memory consumption of a model checker and allows the verification of complex models.