Towards the Verification of Refactorings of Hybrid Simulink Models

Sebastian Schlesinger Paula Herber Thomas Göthel
Sabine Glesner
Technische Universität Berlin
Software Engineering of Embedded Systems
{Sebastian.Schlesinger, Paula.Herber, Thomas.Goethel, Sabine.Glesner}@tu-berlin.de

Extended Abstract

MATLAB/Simulink is a state-of-the-art tool for model-driven engineering to describe hybrid models that contain discrete and continuous blocks that are connected via signal lines. Blocks modify the signals, which are values evolving over simulation time. In such models, complexity reduction via refactoring plays an important role. However, formal verification of the equivalence between a hybrid Simulink model and its refactored counterpart is still an open problem.

We tackle the problem of verifying behavioural equivalence of refactorings by 1) providing an abstract representation for Simulink models in the form of equations that express the relation between ingoing and outgoing signals at the blocks, 2) proving the soundness of the abstract representation by relating it to an operational semantics of Simulink's simulation engine, 3) adapting the notion of *approximate bisimulation* to Simulink models, which allows values of equivalent steps of source and target model to be 'close' to each other rather than being identical as in bisimulation, and 4) providing a methodology for proving the approximate bisimulation for refactorings for three kinds of Simulink models - unsampled (models without discrete or continuous blocks), discrete and continuous systems. While we consider actual equivalence for the first two kinds of systems, for the latter kind of systems, we provide an epsilon 'tube' for the precision of the approximate bisimulation.

The scenario for our paper is the following. Some user wants to refactor a Simulink model. That is only admissible if the behaviour of source and target model is equivalent. We present a first step towards a framework that enables the user to verify the (approximate) behavioural equivalence of source and target Simulink model. In this context, the representation of Simulink models as equations allows us to abstract from the details of the operational semantics and to verify approximate semantical equivalence on the basis of a syntactical comparison of these equations. Examples for refactorings that we support are 1) arithmetic equivalences (e.g. distribution law) for the unsampled case, 2) substitution of several unit delay blocks by one delay block for the discrete case, and 3) substitution of a system that describes a set of ordinary differential equations by a system that expresses the analytical solution directly.

In future work, we aim at automating our approach. To this end, we want to provide automatic means to 1) generate the equations from Simulink models and 2) transform them into a normal form that serves as basis for syntactical comparison. We also strive for covering *hybrid* models, i.e., models that contain both, discrete and continuous blocks. Furthermore, the precision of the epsilon tube that gets larger with the size of the simulation interval, could be improved by taking more details of the simulation semantics into account.

This work is funded by the Deutsche Forschungsgemeinschaft in the project *Correct Model Transformations* (*CorMoranT*).

Submitted to:
VPT 2015