

Static and Dynamic Debugging of Modelica Models

Adrian Pop¹, Martin Sjölund¹, Adeel Asghar¹, Peter Fritzson¹, Francesco Casella²

¹Programming Environments Laboratory

Department of Computer and Information Science

Linköping University, Linköping, Sweden

²Dipartimento di Elettronica e Informazione,

Politecnico di Milano, Milano, Italy

{adrian.pop,martin.sjolund,adeel.asghar,peter.fritzson}@liu.se
casella@elet.polimi.it

Advanced development of today's complex products requires integrated environments and equation-based object-oriented declarative (EEO) languages such as Modelica for modeling and simulation. The increased ease of use, the high abstraction, and the expressivity of such languages are very attractive properties. However, these attractive properties come with the drawback that programming and modeling errors are often hard to find.

In this paper we present static and dynamic debugging methods for Modelica models and a debugger prototype that addresses several of those problems. The goal is an integrated debugging framework that combines classical debugging techniques with special techniques for equation-based languages partly based on graph visualization and interaction.

The static transformational debugging functionality addresses the problem that model compilers are optimized so heavily that it is hard to tell the origin of an equation during runtime. This work proposes and implements a prototype of a method that is efficient with less than one percent overhead, yet manages to keep track of all the transformations/operations that the compiler performs on the model.

Modelica models often contain functions and algorithm sections with algorithmic code. The fraction of algorithmic code is increasing since Modelica, in addition to equation-based modeling, is also used for embedded system control code as well as symbolic model transformations in applications using the MetaModelica language extension.

Our earlier work in debuggers for the algorithmic subset of Modelica used high-level code instrumentation techniques which are portable but turned out to have too much overhead for large applications. The new dynamic algorithmic code debugger is the first Modelica debugger that can operate without high-level code instrumentation. Instead, it communicates with a low-level C-language symbolic debugger to directly extract information from a running executable, set and remove breakpoints, etc. This is made possible by the new bootstrapped OpenModelica compiler which keeps track of a detailed mapping from the high level Modelica code down to the generated C code compiled to machine code.

The dynamic algorithmic code debugger is operational, supports both standard Modelica data structures and tree/list data structures, and operates efficiently on large applications such as the OpenModelica compiler with more than 100 000 lines of code. Moreover, an integrated debugging approach is proposed that combines static and dynamic debugging. To our knowledge, this is the first Modelica debugger that supports both transformational and algorithmic code debugging.