Tools for Understanding, Debugging, and Simulation Performance Improvement of Equation-Based Models

by

Martin Sjölund

May 2013
ISBN 978-91-7519-624-4
Linköping Studies in Science and Technology
Thesis No. 1592
ISSN 0280–7971
LiU–Tek–Lic–2013:26

ABSTRACT

Equation-based object-oriented (EOO) modelling languages provide a convenient, declarative method for describing models of cyber-physical systems. Because of the ease of use of EOO languages, large and complex models can be built with limited effort. However, current state-of-the-art tools do not provide the user with enough information when errors appear or simulation results are wrong. It is paramount that the tools give the user enough information to correct errors or understand where the problems that lead to wrong simulation results are located. However, understanding the model translation process of an EOO compiler is a daunting task that not only requires knowledge of the numerical algorithms that the tool executes during simulation, but also the complex symbolic transformations being performed.

In this work, we develop and explore methods where the EOO tool records the transformations during the translation process in order to provide better diagnostics, explanations, and analysis. This information can be used to generate better error-messages during translation. It can also be used to provide better debugging for a simulation that produces unexpected results or where numerical methods fail.

Meeting deadlines is particularly important for real-time applications. It is usually important to identify possible bottlenecks and either simplify the model or give hints to the compiler that enables it to generate faster code. When profiling and measuring execution times of parts of the model the recorded information can also be used to find out why a particular system is slow. Combined with debugging information, it is possible to find out why this system of equations is slow to solve, which helps understanding what can be done to simplify the model.

Finally, we provide a method and tool prototype suitable for speeding up simulations by compiling a simulation executable for a parallel platform by partitioning the model at appropriate places.

This work has been supported by Vinnova in the ITEA2 OPENPROD and MODRIO projects, by SSF in the Provoking HIPo and EDOP projects, and by the National Graduate School of Computer Science (CUGS). The Open Source Modelica Consortium supports the OpenModelica project.

Department of Computer and Information Science
Linköpings universitet
SE-581 83 Linköping, Sweden

The full thesis is available here:
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-90096