

Tool Demonstration Abstract: OpenModelica Graphical Editor and Debugger

Adeel Asghar Peter Fritzon

Department of Computer and Information Science, Linköping University, Sweden,
{adeel.asghar,peter.fritzon}@liu.se

Abstract

This paper demonstrates the OpenModelica graphic editor for easy-to-use graphic modeling of Modelica models and the Modelica debugger.

The graphic editor aims at providing a user friendly open source Modelica modeling graphical user interface since most of the already existing open source tools were either textual or not so user friendly. The target audiences for the tool are the Modelica users who want easy-to-use model creation, library browsing, connection editing, simulation of models, plotting results and visualization of components.

Modeling errors and problems are often hard to find because of the high abstraction level of languages like Modelica. Models containing functions with huge algorithm sections increase the need for run-time debugging. The OpenModelica debugger provides a debugging of such models. The debugger currently supports debugging of algorithmic code. The debugger uses the Gnu low-level C-language debugger (GDB) for low-level manipulation and control of the executing program during debugging.

Keywords Graphic editor, Connection Diagrams, Run-time Debugging, Modeling and Simulation, Algorithmic code.

1. OpenModelica Graphical Editor

The OpenModelica graphical editor (OMEdit) is an integrated development environment for Modelica where users can model, simulate and plot their physical systems designs.

It supports the Modelica standard library 3.2.1 through the graphical annotations. It is based on OpenModelica's interactive scripting environment. The scripting environment is part of the OpenModelica compiler. The communication with the scripting environment is carried out through the CORBA interface. During the communication the OpenModelica compiler acts as server

while the graphic editor acts as a client.

OMEdit provides user friendly features like;

- Modeling – easy Modelica model creation.
- Browsing – Modelica standard library browsing.
- Component interfaces – smart connection editing for drawing and editing connections between model interfaces.
- Simulation subsystem – subsystem for running simulations and specifying simulation parameters start and stop time, etc.
- Plotting – interface to plot variables from simulated models.

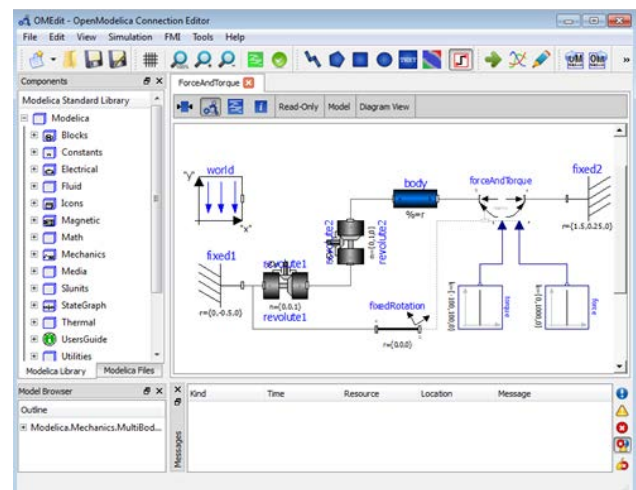


Figure 1. OpenModelica Graphical Editor.

2. OpenModelica Debugger

The debugger is integrated within the Modelica Development Tooling (MDT) which is an Eclipse plugin. It communicates with the Gnu debugger (GDB) via its Machine Interface (MI) channel. Figure 2 shows the Eclipse-based user interface of the debugger.

The debugger provides the following general functionalities:

- Adding/Removing breakpoints.
- Step Over – moves to the next line, skipping the function calls.

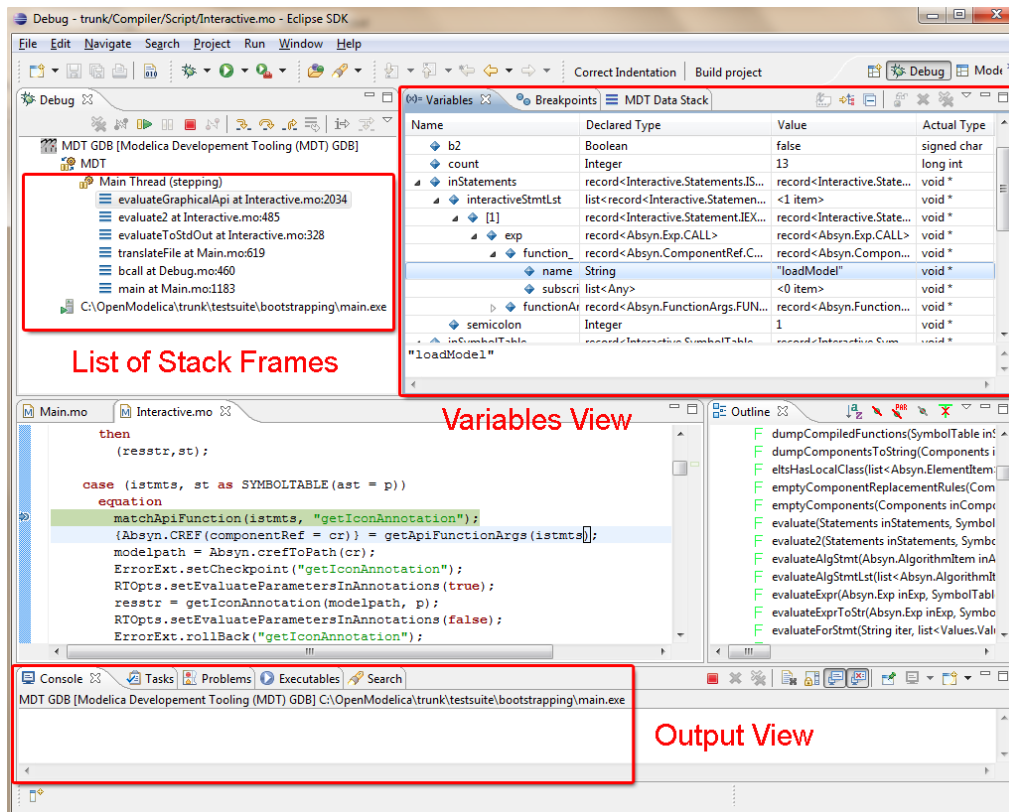


Figure 2. The debug view of the debugger within the MDT Eclipse plugin.

- Step In – takes the user into the function call.
- Step Return – completes the execution of the function and takes the user back to the point from where the function is called.
- Suspend – interrupts the running program.

The debug view primarily consists of two main views:

- Stack Frames View
- Variables View

The stack frames view shows a list of frames that indicates the program flow. The variables view shows the list of variables at the current stack position.

Acknowledgements

This work has been supported by the Open Source Modelica Consortium (OSMC), by Vinnova in the RTSIM, ITEA2 OPENPROD and ITEA2 MODRIO projects, and by SSF in the EDOp project.

References

[1] Peter Fritzon. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1, 940 pp., ISBN 0-471-471631, Wiley-IEEE Press, 2004.

[2] The Modelica Association. The Modelica Language Specification Version 3.2. <http://www.modelica.org>

[3] Adeel Asghar, Sonia Tariq, Mohsen Torabzadeh-Tari, Peter Fritzon, Adrian Pop, Martin Sjölund, Parham Vasaiely, and Wladimir Schamai. An Open Source Modelica Graphic Editor Integrated with Electronic Notebooks and Interactive Simulation. In Proceedings of

the 8th International Modelica Conference, Dresden, Germany, March.20-22, 2011.

[4] Peter Bunus. Debugging Techniques for Equation-Based Languages. PhD Thesis. Department of Computer and Information Science, Linköping University, 2004.

[5] Adrian Pop and Peter Fritzon. A Portable Debugger for Algorithmic Modelica Code. In Proceedings of the 4th International Modelica Conference, Hamburg, Germany, March 7-8, 2005.

[6] Adeel Asghar, Adrian Pop, Martin Sjölund, and Peter Fritzon. Efficient Debugging of Large Algorithmic Modelica Applications. In Proceedings of MATHMOD 2012 7th Vienna International Conference on Mathematical Modelling, Vienna University of Technology, Vienna, Austria, February 15 - 17, 2012.

[7] Adrian Pop, Peter Fritzon, Andreas Remar, Elmir Jagudini, and David Akhvlediani. OpenModelica Development Environment with Eclipse Integration for Browsing, Modeling, and Debugging. In Proceedings of the Modelica'2006, Vienna, Austria, Sept. 4-5, 2006.

[8] Adrian Pop, David Akhvlediani, and Peter Fritzon. Towards Run-time Debugging of Equation-based Object-oriented Languages. In Proceedings of the 48th Scandinavian Conference on Simulation and Modeling (SIMS'2007), available at www.scan-sims.org and <http://www.ep.liu.se>. Göteborg, Sweden. October 30-31, 2007.

[9] Martin Sjölund and Peter Fritzon. Debugging Symbolic Transformations in Equation Systems. In Proceedings of the 4th International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools, (EOOLT'2011), Zürich, Switzerland, Sept 5, 2011.

[10] Richard Stallman, Roland Pesch, Stan Shebs, et al. Debugging with GDB. Free Software Foundation, 2011.