

Equation-Based Object-Oriented Modeling Languages and Tools

Report on the Workshop EOOLT 2010 at MODELS 2010
Oslo, Norway, October 3, 2010

Peter Fritzson¹, Edward A. Lee², François E. Cellier³, and David Broman¹

¹Linköping University, Sweden, peter.fritzson@liu.se, david.broman@liu.se

²UC Berkeley, CA, USA, eal@eecs.berkeley.edu

³ETH Zurich, Switzerland, fcellier@inf.ethz.ch

Abstract. EOOLT 2010 was the third edition of the EOOLT workshop series. The workshop is intended to bring together researchers and practitioners from different equation-based object-oriented (EEO) modeling language communities. This year's workshop also expands the scope to include the whole design space of languages for cyber-physical systems, where physical dynamics are mixed with networks and software. The workshop gathered 31 participants to present and discuss thirteen different papers grouped into the four areas of real-time oriented modeling languages and tools, modeling language design, simulation and model compilation, and modeling and simulation tools.

1 Introduction

During the last decade, integrated model-based design of complex cyber-physical systems (which mix physical dynamics with software and networks) has gained significant attention. Hybrid modeling languages based on equations, supporting both continuous-time and event-based aspects (e.g. Modelica, SysML, VHDL-AMS, and Simulink/ Simscape) enable high-level reuse and integrated modeling capabilities of both the physically surrounding system and software for embedded systems. Using such equation-based object-oriented (EEO) modeling languages, it has become possible to model complex systems covering multiple application domains at a high level of abstraction through reusable model components.

The interest in EEO languages and tools is rapidly growing in the industry because of their increasing importance in modeling, simulation, and specification of complex systems. There exist several different EEO language communities today that grew out of different application areas (multi-body system dynamics, electronic circuit simulation, chemical process engineering). The members of these disparate communities rarely talk to each other in spite of the similarities of their modeling and simulation needs.

The EOOLT workshop series aims at bringing these different communities together to discuss their common needs and goals as well as the algorithms and tools that best support them.

It was a good response to the call-for-papers. Eleven papers were accepted for full presentations and two papers for short presentations in the workshop program out of eighteen submissions. All papers were subject to rather detailed reviews by the program committee, on the average four reviews per paper. The workshop program started with a welcome and introduction to the area of equation-based object-oriented languages, followed by paper presentations. Discussion sessions were held after presentations of each set of related papers. There were 31 participants in the workshop, more than doubled compared to EOOLT 2008 which was held in conjunction with ECOOP 2008.

After the event of the workshop, a nomination request for the best paper of the workshop was sent out to all PC members, authors, and workshop participants. Authors were not allowed to nominate themselves. Seven different papers were nominated, where the paper “Modal Models in Ptolemy” by Edward A. Lee and Stavros Tripakis received a clear majority of the nominations. This paper was selected as the best paper of EOOLT 2010, and the abstract is published in a post proceedings of the MODELS conference.

The venue for EOOLT 2010 was Oslo, Norway, in conjunction with the MODELS 2010 conference.

2 Program Chairs/Organizers and Program Committee

Peter Fritzon, Chair	Linköping University, Linköping, Sweden
Edward A. Lee, Co-Chair	U.C. Berkeley, USA
François E. Cellier, Co-Chair	ETH Zurich, Switzerland
David Broman, Co-Chair	Linköping University, Linköping, Sweden
Bernhard Bachmann	Univ. of Applied Sciences, Bielefeld, Germany
Bert van Beek	Eindhoven Univ. of Technology, Netherlands
Felix Breitenecker	Technical University of Vienna, Vienna, Austria
Jan Broenink	University of Twente, Netherlands
Peter Bunus	Linköping University, Linköping, Sweden
Francesco Casella	Politecnico di Milano, Italy
Hilding Elmqvist	Dassault Systèmes, Lund, Sweden
Olaf Enge-Rosenblatt	Fraunhofer Inst. for Integrated Circuits, Dresden
Petter Krus	Linköping University, Linköping, Sweden
Sven-Erik Mattsson	Dassault Systèmes, Lund, Sweden
Jacob Mauss	QTronic GmbH, Berlin, Germany
Pieter Mosterman	MathWorks, Inc., Natick, MA, USA
Toby Myers	Griffith University, Brisbane, Australia
Henrik Nilsson	University of Nottingham, Nottingham, UK
Dionisio de Niz Villasenor	Carnegie Mellon University, Pittsburgh, USA
Hans Olsson	Dassault Systèmes, Lund, Sweden
Martin Otter	DLR Oberpfaffenhofen, Germany
Chris Paredis	Georgia Institute of Technology, Atlanta, USA

Peter Pepper	TU Berlin, Berlin, Germany
Adrian Pop	Linköping University, Linköping, Sweden
Nicolas Rouquette	NASA Jet Propulsion Laboratory, USA
Peter Schwarz	Fraunhofer Inst. for Integrated Circuits, Dresden
Christian Sonntag	TU Dortmund, Dortmund, Germany
Martin Törngren	KTH, Stockholm, Sweden
Alfonso Urquía	National Univ. for Distance Education, Madrid
Hans Vangheluwe	McGill University, Montreal, Canada
Dirk Zimmer	DLR Oberpfaffenhofen, Germany
Johan Åkesson	Lund University, Lund, Sweden

3 Publication

All papers are published electronically by Linköping University Electronic Press [1] and available in the electronic proceedings at <http://www.ep.liu.se/ecp/047/>.

All presentations (together with the papers) are also available at the EOOLT 2010 web site: <http://www.eoolt.org/2010/>.

4 Sessions

The workshop sessions are briefly described below. Each session starts with paper presentations, followed by a discussion related to the topic of that particular session. Some discussion also took place during the paper presentations.

4.1 Real-Time Oriented Modeling Languages and Tools

Session chair: David Broman

In this session, research work was presented related to graphical modeling languages, real-time applications, and profiling. Three papers were presented and discussed.

In “Execution of UML State Machines Using Modelica,” Wladimir Schamai, Uwe Pohlmann, Peter Fritzson, Christiaan J.J. Paredis, Philipp Helle, and Carsten Strobel present the ModelicaML language and how it can be used for modeling UML state machines. Wladimir presented a translational approach for code generation of Modelica code as well as a priority schema for handling the problem of conflicting transitions in UML diagrams.

In “Modal Models in Ptolemy,” Edward A. Lee and Stavros Tripakis discuss the concepts and semantics of modal models and how time is handled in refined sub-models in finite state machines (FSMs). Examples were given in the Ptolemy II environment. One conclusion is that refined modes should have a local notation of time that does not advance while a mode is inactive. Hence, the gap between local time and global time is monotonically increasing.

In “Profiling of Modelica Real-Time Models,” Christian Schulze, Michaela Huhn, and Martin Schöler present an approach and implementation for profiling of Modelica models used in real-time applications. It was concluded using a case study that profil-

ing can help identifying the workload for parts of a model. It was also pointed out that it is important to separate the process of saving result data to a hard disk drive into a non real-time application.

In the following discussion session, questions and comments from the audience concerned all three paper presentations. The main questions and comments for the first presentation related to Stategraph, static checking, and possibility of round-trip engineering. For the second presentation, questions concerned the similarity to Simulink and semantics for stream processing in Ptolemy. It was pointed out that the semantics are not built into the system, but are defined by different directors. The discussions concerning the last presentation focused on different challenges of implementation of a profiling system, in particular regarding the possibility of tracing back to the original Modelica model.

4.2 Modeling Language Design

Session chair: Edward A. Lee

The second session focused on design and implementation aspects of Modelica.

“Towards Improved Class Parameterization and Class Generation in Modelica,” Dirk Zimmer introduces the idea that class parameterization and class generation should be separate concepts in Modelica. The goal is to partially redesign Modelica, to unify concepts, and to simplify the language.

In “Notes on the Separate Compilation of Modelica,” Christoph Höger, Florian Lorenzen, and Peter Pepper discuss different problems and implications of introducing separate compilation of Modelica models, e.g., runtime instantiation, introducing coercion functions, and handling of dynamic binding. It is noted that the Modelica language is very complex and that special cases of the semantics need to be reduced.

In “Import of Distributed Parameter Models into Lumped Parameter Model Libraries for Linearly Deformable Solid Bodies,” Tobias Zaiczek and Olaf Enge-Rosenblatt show how distributed parameter models can be included in libraries of lumped parameter models. Discretization, connector definitions, and model order reduction are analyzed with regards to flexible bodies modeling and simulation.

In the following discussion session, the discussion related to the first talk concerned types, models as first class, and different aspects of concrete syntax. The discussion about the second talk focused on when elaboration/flattening and symbolic manipulation should take place. Should it be at compile-time, link-time, or at runtime? Finally for the last talk, questions were raised about related work, i.e., performance comparison with simulation tools such as Dymola and comparison to other PDE Modelica efforts.

4.3 Simulation and Model Compilation

Session chair: François E. Cellier

In this session, three research papers were presented related to synchronous event handling together with a numerical solver, distributed simulation using TLM techniques, and profiling.

In “Synchronous Events in the OpenModelica Compiler with a Petri Net Library Application,” Willi Braun, Bernhard Bachmann, and Sabrina Proß describe improved techniques for synchronous event handling using the DASSL solver in OpenModelica, with applications in a Petri Net library.

In “Towards Efficient Distributed Simulation in Modelica using Transmission Line Modeling,” Martin Sjölund, Robert Braun, Peter Fritzson, and Petter Krus describe the TLM model partitioning technique, how this can be integrated in Modelica to enable efficient simulation, and results from a prototype implementation.

In “Compilation of Modelica Array Computations into Single Assignment C for Efficient Execution on CUDA-enabled GPUs,” Kristian Stavåker, Daniel Rolls, Jing Guo, Peter Fritzson, and Sven-Bodo Scholz describe methods to compile repetitive equations and array equations to SAC code running on GPUs, with measurements.

Discussions touched issues like convergence of event iteration, synchronous event handling (1st talk), fixed or flexible time steps, interpolation (2nd talk), and large arrays, need for combination of task and data parallelism, size of equation systems that can be handled, and handling models with algebraic loops (3rd talk).

4.4 Modeling and Simulation Tools

Session chair: Peter Fritzson

This session presented research related to tool functionality such as XML representation of systems of equations, computer algebra operations on models, a comparison between DASSL and QSS numeric solvers, and model debugging through model reduction. Two long papers followed by two short were presented and discussed.

In “An XML Representation of DAE Systems Obtained from Continuous-time Modelica Models,” Roberto Parrotto, Johan Åkesson, and Francesco Casella describe an XML format for model equations and its usage for model export to other tools.

In “Towards a Computer Algebra System with Automatic Differentiation for Use with Object-Oriented Modelling,” Joel Anderson, Boris Houska, and Moritz Diehl describe a special-purpose small C++ based tool for automatic differentiation.

In “Discretising Time or States? A Comparative Study between DASSL and QSS,” Xenofon Floros, François E. Cellier, and Ernesto Kofman describe a new simulation run-time system for OpenModelica based on quantized state systems (QSS) simulation and compare this approach to the standard DASSL solver.

In “Model Verification and Debugging of EOO Models Aided by Model Reduction Techniques,” Anton Sodja and Borut Zupančič give an overview of model reduction techniques and argue that such techniques are useful for debugging and verification.

The following discussion covered e.g., the difference between FMI and the XML, advantages / issues with QSS, the current status of model reduction techniques, etc.

References

- [1] Peter Fritzson, Edward Lee, François Cellier, and David Broman (Eds.). Proceedings of the 3rd International Workshop on Equation-Based Object-Oriented Modeling Languages and Tools . Oslo, Norway, October 2010. ISSN 1650-3740, LIU Electronic Press.