

Type Safety of Equation-Based Object-Oriented Modeling Languages

PhD Thesis Outline

PhD Student: David Broman

Department of Computer and Information Science
Linköping University, Sweden
davbr@ida.liu.se

Supervisor: Peter Fritzson

Department of Computer and Information Science
Linköping University, Sweden
petfr@ida.liu.se

Abstract

Computer aided modeling and simulation of complex physical systems, using components from multiple domains, such as electrical, mechanical, and hydraulic, have in recent years witnessed a significant growth of interest. General-purpose simulation tools, e.g. Simulink [11], using block diagrams and causal connections have dominated the area for years. However, in the last decade novel languages, (e.g. Modelica [5, 7, 12], gPROMS [13], χ language [6], and VHDL-AMS [4]) based on acausal modeling using Differential Algebraic Equations (DAEs), have evolved.

The research in the area of equation-based object-oriented (EOO) languages has until now mainly focused on generation of efficient simulation code and integration with numerical solvers. Programming language theory, formal semantics, and type systems are typically used in a limited way, which may result in inconsistent languages and model errors that are hard to detect for the end user. A common problem concerns the static detection of too few or too many equations giving an under- or over-constrained system. Solutions have been developed at the level of the overall flat system of equations derived from models [2][3], but to the best of our knowledge, no research has been performed at the component and type system level.

Physical unit annotations (e.g. newton, meter, ampere) are ordinary parts of many EOO-languages. However, the precise semantics for static unit type checking is limited or absent. The need for such automatic static control of units is crucial in many simulation experiments, since it is extremely difficult to manually find inconsistencies in models with thousands of equations. Many proposals for conventional programming languages exist [1, 9, 10, 15], but only few [14] sound attempts exist to incorporate unit type checking into the type system of EOO-languages.

Our main objective in this thesis work is to improve the theoretical foundation of EOO-languages by developing a formally specified kernel language from which other modeling constructs can be expressed. The Modeling Kernel Language (MKL) is intended to describe the core concept of an EOO-language by using a limited number of constructs and excluding any syntactic sugar normally needed for convenient modeling. To verify the correctness of our solution, MKL is specified using formal semantics and a formal type system, followed by proof of type safety (soundness) [16]. Moreover, to justify that our solution indeed addresses the problems of static unit type checking and detection of under- and over-constrained systems, models described in the Mod-

elica language are automatically transformed and simulated on the prototype implementation of MKL. The experiments are performed using real models collected from industry.

Hence, on a higher level, the aim of this thesis project is to increase the ability for an engineer to detect model errors at an early stage. By providing better static type checking in EOO-languages, our work intend to meet the increasing industrial requirement of robust modeling and simulation environments.

Acknowledgments

This research work is funded by CUGS (the Swedish National Graduate School in Computer Science), by SSF under the VISIMOD project, and by Vinnova under the NET-PROG Safe and Secure Modeling and Simulation on the GRID project.

References

- [1] Eric Allen, David Chase, Victor Luchangco, Jan-Willem Maessen, and Jr. Guy L. Steele. Object-Oriented Units of Measurement. In *OOPSLA '04: Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*, pages 384–403, New York, USA, 2004. ACM Press.
- [2] Peter Bunus. *Debugging Techniques for Equation-Based Languages*. PhD thesis, Linköping University, 2004.
- [3] Peter Bunus and Peter Fritzson. Automated Static Analysis of Equation-Based Components. *SIMULATION*, 80(7–8):321–245, 2004.
- [4] Ernst Christen and Kenneth Bakalar. VHDL-AMS - A Hardware Description Language for Analog and Mixed-Signal Applications. *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 46(10):1263–1272, 1999.
- [5] Hilding Elmqvist, Sven Erik Mattsson, and Martin Otter. Modelica - A Language for Physical System Modeling, Visualization and Interaction. In *Proceedings of the IEEE International Symposium on Computer Aided Control System Design*, 1999.
- [6] Georgina Fábán. *A Language and Simulator for Hybrid Systems*. PhD thesis, Technische Universiteit Eindhoven, the Netherlands, 1999.
- [7] Peter Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE Press, New York, USA, 2004.
- [8] Peter Fritzson, Peter Aronsson, Håkan Lundvall, Kaj Nyström, Adrian Pop, Levon Saldamli, and David Broman. The OpenModelica Modeling, Simulation, and Development

- Environment. In *Proceedings of the 46th Conference on Simulation and Modeling (SIMS'05)*, pages 83–90, 2005.
- [9] Paul N. Hilfinger. An ADA Package for Dimensional Analysis. *ACM Trans. Program. Lang. Syst.*, 10(2):189–203, 1988.
- [10] Andrew Kennedy. *Programming Languages and Dimensions*. PhD thesis, University of Cambridge, 1996.
- [11] MathWorks. The Mathworks - Simulink - Simulation and Model-Based Design. <http://www.mathworks.com/products/simulink/> [Last accessed: 15 May 2006].
- [12] Modelica Association. *Modelica - A Unified Object-Oriented Language for Physical Systems Modeling - Language Specification Version 2.2*, February 2005. Available from: <http://www.modelica.org> [Last accessed: 29 March 2006].
- [13] M. Oh and C. C. Pantelides. A modelling and Simulation Language for Combined Lumped and Distributed Parameter Systems. *Computers and Chemical Engineering*, 20(6–7):611–633, 1996.
- [14] Mikael Sandberg, Daniel Persson, and Björn Lisper. Automatic Dimensional Consistency Checking for Simulation Specifications. In *SIMS 2003*, page 6, September 2003.
- [15] Zerkis D. Umrigar. Fully static dimensional analysis with C++. *SIGPLAN Not.*, 29(9):135–139, 1994.
- [16] Andrew K. Wright and Matthias Felleisen. A Syntactic Approach to Type Soundness. *Information and Computation*, 115(1):38–94, 1994.