Object–Oriented Modeling of Physical Systems: Promises and Expectations

Moderator: François E. Cellier, Dept. of Electr. & Comp. Engr.,
University of Arizona, Tucson, AZ 85721, U.S.A.
E_Mail: Cellier@ECE.Arizona.Edu, URL: http://www.ece.arizona.edu/~cellier/

Panelists: Wolfgang Borutzky, Cologne Polytechnic, Germany
Peter Breedveld, University of Twente, The Netherlands
Hilding Elmqvist, Dynasim AB, Lund, Sweden
Martin Otter, DLR Oberpfaffenhofen, Germany
Miquel Angel Piera, Autonomous University of Barcelona, Spain

The basic foundations of object–oriented modeling as a means to mathematically capture the dynamical behavior of physical systems were laid almost 20 years ago independently by Tom Runge [9] and Hilding Elmqvist [3] in two seminal Ph.D. dissertations. Whereas Runge advocated the use of general–purpose differential algebraic equation (DAE) solvers as the main simulation engine, Elmqvist advanced symbolic formulae manipulation as a means to automatically transform the model equations to state–space from. Both theses were based on newly designed textual languages (*Dymola* and *Model*) as the primary user interfaces; however, Tom Runge even introduced a (still fairly rudimentary) object–diagram editor (*Dragon* [10]) as an alternative approach for model input.

The basic ideas for Tom Runge's dissertation were taken, in his own words, from a visionary paper by Bill Gear [5], his dissertation advisor. A special class of object–oriented modeling methodology, the bond graph, had been around even before then. The bond graph approach to physical system modeling was conceptualized by Hank Paynter on April 24, 1959 [7] who, in turn, claims that his work had been influenced in major ways by the even earlier work of Gabriel Kron [6].

Yet, the two dissertations by Runge and Elmqvist were far ahead of their times. As a consequence, neither technology received a high degree of attention for close to a decade. The computer hardware of the late seventies was far from ready for the proposed software technologies (Hilding once built his own mouse interface, because he couldn't find one for the VAX), and the market demand for such sophisticated modeling software was simply not there yet.

Only the early nineties, with their growing demands for simulation models of ever increasing complexity describing models of advanced engineering systems with component models stemming from different application fields, as they are used in e.g. mechatronics, electronic packaging (with electronic, mechanical, and thermal submodels), aircraft electronics (subjected to high temperature, strong magnetic fields, and heavy mechanical vibrations), the electronic fuel injection control system of a car, to mention just a few examples, called for the sophistication that object–oriented modeling tools have to offer. This growing demand went hand–in–hand with the deployment of graphical workstations that finally provided a platform powerful enough to host a sophisticated object–oriented modeling environment.

The object–oriented physical systems modeling tools [1, 2, 4, 8] available today all offer object–diagram editors as well as textual model representations. They all support a mixture of symbolic model preprocessing with advanced DAE solvers. Had the theses of Runge and Elmqvist been viewed by many as somewhat esoteric at the time they were written, no–one questions any longer the relevance and significance of this research branch. In fact, every one of the above software tools is dramatically better than anything that had been on the market earlier in terms of general–purpose physical system modeling support.

This panel discussion is a first attempt at bringing many the major players together and have them discuss, with inclusion of the attendees, what their views are w.r.t. future developments in this exciting and dynamic research area.

References

[1] M. Andersson, *Object–Oriented Modeling and Simulation of Hybrid Systems*, Ph.D. dissertation

TFRT–1043, Dept. of Automatic Control, Lund Inst. of Technology, Lund, Sweden, 1994.

[2] W. Borutzky, J.F. Broenink, and K.C.J. Wijbrans, "Graphical Description of Physical System Models Containing Discontinuities," *Proc. ESM'93, European Simulation Multiconference*, Lyon, France, 1993, pp. 209–214.

[3] H. Elmqvist, *A Structured Model Language for Large Continuous Systems*, Ph.D. Dissertation, Report CODEN: LUTFD2/(TFRT–1015), Dept. of Automatic Control, Lund Inst. of Technology, Lund, Sweden, 1978.

[4] H. Elmqvist, *Dymola: Dynamic Modeling Language — User's Guide*, Dynasim AB, Lund, Sweden, 1995.

[5] C.W. Gear, "A Generalized Interactive Network Analysis and Simulation System," *Proc. 1st USA–Japan Computer Conf.*, 1972, pp. 559–566.

[6] G. Kron, *Diakoptics — The Piecewise Solution to Large–Scale Systems*, MacDonald & Co., London, 1962.

[7] H.M. Paynter, *Analysis and Design of Engineering Systems*, MIT Press, Cambridge, MA, 1961.

[8] J.J. Ramos, M.A. Piera, and I. Serra, "A Modelling Tool to Guide Computational Causality Assignment Through Physical Causality Analysis," *Proc. EUROSIM'95 Simulation Congress*, Vienna, Austria, 1995, pp. 105–110.

[9] T.F. Runge, *A Universal Language for Continuous Network Simulation*, Ph.D. dissertation, Form: UIUCDCS-R-77-866, Dept. of Comp. Science, University of Illinois, Urbana–Champaign, U.S.A., 1977.

[10] P. Watterberg, and V. Lo, *The Dragon Users' Manual*, MS thesis, Form: UIUCDCS-R-78-934, Dept. of Comp. Science, University of Illinois, Urbana–Champaign, U.S.A., 1978.