Rewriting-Logic-Based Formal Modeling and Analysis of Interacting Hybrid Systems

Muhammad Fadlisyah
University of Oslo, Norway

Erika Ábrahám
RWTH Aachen, Germany

Peter Csaba Ölveczky
University of Oslo, Norway

Introduction. *Real-Time Maude* [6, 7] is a formal specification language and a simulation, reachability analysis, and LTL model checking tool for real-time systems. Real-Time Maude is based on rewriting logic [5] and emphasizes expressiveness and ease of specification. The tool has proved useful for formally modeling and analyzing a wide range of advanced real-time systems with sophisticated data types and communication models, and unbounded data structures.

In the work summarized in this abstract, we have investigated to what degree – and *how* – Real-Time Maude can also be successfully applied to formally model and analyze advanced *hybrid* systems. In particular, and in contrast to many formal approaches to hybrid systems, we target complex hybrid systems (with both continuous and discrete behaviors) where the physical entities interact and may influence each other's continuous behavior. For example, a hot cup of coffee in a room interacts with the room through different kinds of heat transfer, leading to a decrease in the temperature of the coffee over time, but also to a slight increase in the temperature of the room through heat transfer from the cup. Such a system may also exhibit discrete behaviors, e.g., when the coffee transitions from a "non-boiling" to a "boiling" state, and when we add a computer-controlled heater that turns itself on and off in order to keep the coffee in a desired temperature range.

We have defined a general object-oriented modeling methodology for modeling such interacting hybrid systems in Real-Time Maude. In particular, we consider physical interactions to be first-class citizens, and model a hybrid system as a network of *physical entities* and *physical interactions*; each entity and each interaction between a pair of entities is modeled by an object in Real-Time Maude.

We have adapted four numerical methods for approximating the continuous behaviors of the system to our modeling fraemwork, and have applied our techniques to model, simulate, and model check a set of thermal systems with realistic parameters.

Modeling Interacting Physical Systems in Real-Time Maude. In [2] we present a framework for the modeling and analysis of physical systems based on the *effort* and *flow* approach [8]. [1] A physical system is modeled as a network of *physical entities* and *physical interactions*. A *physical entity* (such as a cup of coffee) consists of a set of *attributes* and a *continuous dynamics*. We consider three kinds of attributes: *continuous* variables (denoting physical quantities, such as the temperature, that change with time), *discrete* variables, and *constants*. A physical entity has one *effort variable*, which is a continuous variable and the "main" attribute of the physical entity. A physical entity can have one or more physical interactions with one or more physical entities. The values of the flow variables of these interactions are used in the computation of the continuous dynamics of the physical entity. A *physical interaction* (such as the heat flow from the coffee to the room) represents an interaction between two physical entities. It consists of one *flow variable*, a set of *attributes*, and a *continuous dynamics*. The flow variable is a continuous variable. The values of the effort variables from the two physical entities are used in the computation of the continuous dynamics of the physical interaction. Figure 1 illustrates how our idea is used to model a simple thermal system consisting of a cup of coffee in a room.

[1] The effort/flow approach is applicable to many kinds of physical systems. In mechanical translation systems, the effort and flow variables denote, respectively, force and velocity; in mechanical rotation systems, torque and angular velocity; in electrical systems, voltage and current; in fluidic systems, pressure and volume flow rate; in thermal systems, temperature and heat flow rate.

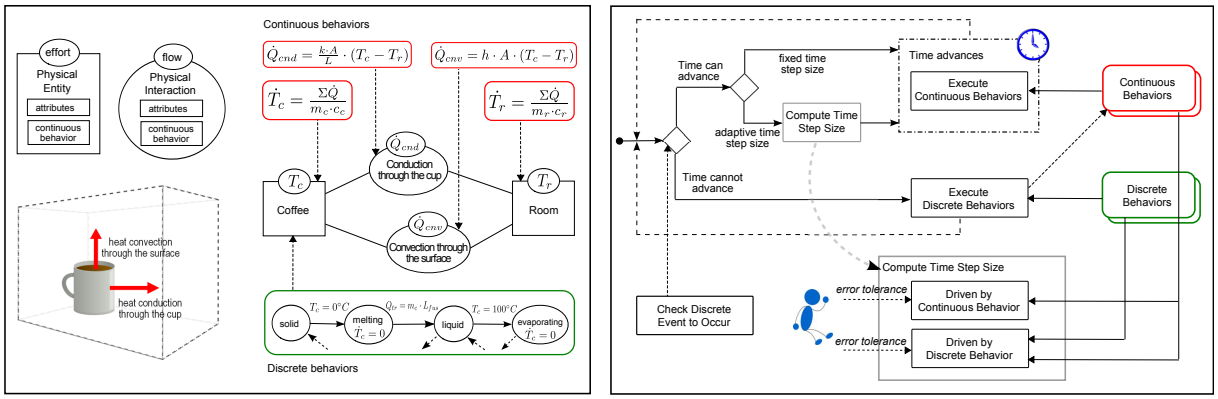

Figure 1: Physical system components and their interaction in a simple thermal system (left), and the execution of system's behaviors (right).

The continuous dynamics of a physical entity is an ordinary differential equation (ODE) with the time derivative of its effort on the left-hand side and an expression possibly referring to the entity's attributes *and* to the flows of connected interactions on the right-hand side. Dually, the continuous dynamics of a physical interaction is an equation with the flow variable on the left-hand side and an expression possibly referring to the interaction's local attributes and the efforts of the connected entities on the right-hand side. This way the direct coupling of the ODEs of physical entities [1] can be avoided.

Executing Continuous Behaviors. We use numerical techniques to *approximate* the continuous behaviors by advancing time in small discrete time increments, and approximating the values of the continuous variables at each "visited" point in time. We define a general method to integrate any *single-step*, *initial-value-problem* numerical method to our effort/flow framework. We have currently adapted the *Euler*, the *Runge-Kutta 2nd order* (RK2), the *Runge-Kutta 4th order* (RK4), and the *Runge-Kutta-Fehlberg 4/5* (RKF45) methods (see [4, 3]). In general, the Euler method is fast and comparatively inaccurate; RK2 is somewhat slower and fairly accurate; and RK4 is significantly slower but quite accurate. The RKF45 method gives the user to possibility to define his/her own error tolerance to balance between desired precision and computational efficiency. This is achieved by *dynamically* changing the time increments in order to: (i) make the analysis more precise by decreasing the time increment when needed to maintain a desired precision of the approximation, and (ii) make the analysis more efficient by increasing the step size whenever the approximation allows it.

We have also developed a *discrete-event-based adaptive* step size technique to approximate the exact time point when a discrete event should occur, and to modify the time step size accordingly.

Case Studies and Experimental Results. Using the above modeling and execution framework, we have modeled, simulated, and formally model checked in Real-Time Maude the following three increasingly more complex case studies, in which we use realistic values for the physical parameters:

1. A cup of coffee (initially 70°) in a room with temperature 20°; heat flows from the cup to the room by *thermal conduction* through the cup, and by *thermal convection* through the coffee surface.
2. The initial (ice) coffee temperature in this case study is $-20°$ (to simulate the various phases of the coffee), and we add a heater that provides constant heat to the cup.
3. Adds a controller to the heater which turns the heater on and off to keep the temperature of the coffee between 70° and 80°.

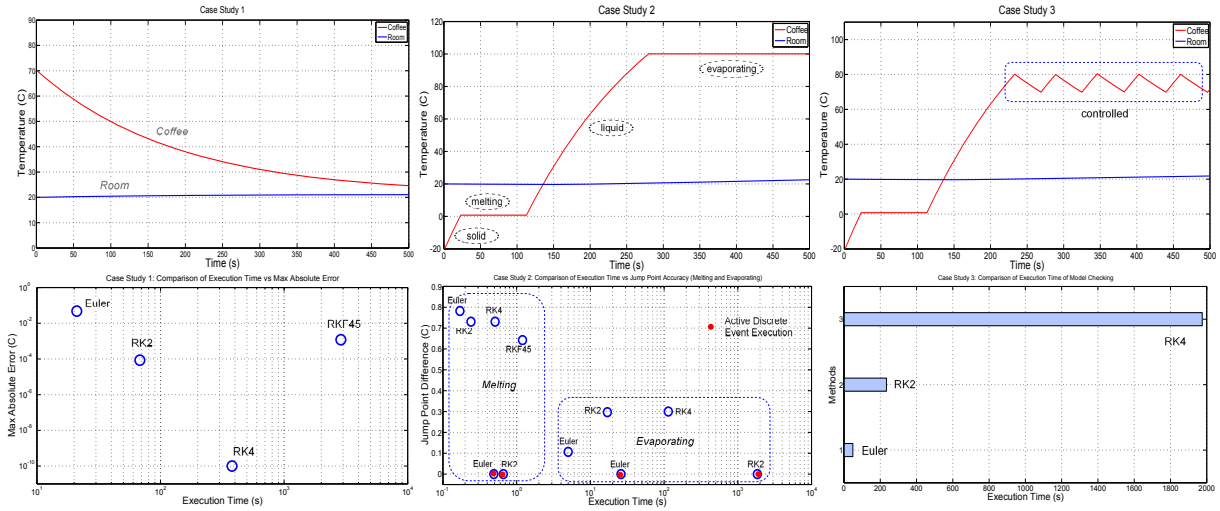

Figure 2: Simulation of three case studies (top) and some results of analysis (bottom).

We have experimented with all four numerical methods. Figure 2 (top) shows the results of the Real-Time Maude simulations[2] of the three case studies (given as the temperature of the two entities over time). For case study 1, we have also computed the analytic solution of the set of ODEs defining the system dynamics (this is not possible in general), and show the maximum deviation from the analytic solution and the execution time for simulation up to model time 500 for the four numerical methods (Fig. 2, bottom left). We notice that even though the adaptive step size method reduces the number of visited time points, the cost of computing the step size is high. Finally, Fig. 2, bottom right, shows the execution times for model checking in case study 3 the LTL *stability property* that once the coffee reaches the desired temperature interval $(69.5^\circ, 80.5^\circ)$[3], it stays within this range, for all behaviors up to time 500 (this corresponds to the LTL property $\Box(OK\_temp \rightarrow \Box\, OK\_temp)$).

References

[1] W. Cheney and D. Kincaid. *Numerical Mathematics and Computing*. Brooks & Cole Publishing Co., 1994.

[2] M. Fadlisyah, E. Ábrahám, D. Lepri, and P. C. Ölveczky. A rewriting-logic-based technique for modeling thermal systems. In *Proc. RTRTS'10*, volume 36 of *Electronic Proceedings in Theoretical Computer Science*, 2010.

[3] M. Fadlisyah, E. Ábrahám, and P. C. Ölveczky. Adaptive-step-size numerical methods in rewriting-logic-based formal analysis of interacting hybrid systems. Submitted for publication, 2010.

[4] M. Fadlisyah, E. Ábrahám, and P. C. Ölveczky. Formal modeling and analysis of hybrid systems in rewriting logic using higher order numerical methods. Submitted for publication, 2010.

[5] J. Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96:73–155, 1992.

[6] P. C. Ölveczky and J. Meseguer. Semantics and pragmatics of Real-Time Maude. *Higher-Order and Symbolic Computation*, 20(1-2):161–196, 2007.

[7] P. C. Ölveczky and J. Meseguer. The Real-Time Maude tool. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS'08)*, volume 4963 of *LNCS*, pages 332–336. Springer, 2008.

[8] P. E. Wellstead. *Introduction to physical system modelling*. Academic Press, 1979.

[2] We use time increment 1 for the fixed-step-size methods, and error tolerance 10^{-5} for the adaptive one.

[3] All LTL model checking is of course relative to the inaccuracies due to the approximation of the continuous behaviors