Physical Modeling with ModelVision, a DAE Simulator with Features for Hybrid Automata

Abstract for short talk

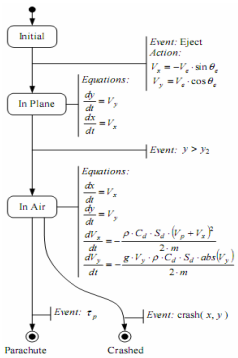
Yuri Senichenkov[1], Felix Breitenecker[2], Günther Zauner[2]
[1]Technical University St.Petersburg, Dept. for Distributed Computing and Networking
[2]Vienna University of Technology, Wiedner Hauptstr. 8-10, 1040 Vienna, Austria
senyb@dcn.infos.ru

Abstract. This contribution presents the modeling capabilities of the fully hybrid simulator ModelVision. Development of ModelVision started in the 1990ies, in autumn 2007 an English version is to be released. Basis of ModelVision are hybrid statecharts, allowing any parallel, serial, and conditional combination of continuous models, described by DAEs. State models itself are objects to be instantiated in various kinds, so that structural-dynamic systems of any kind can be modeled. DAE modeling is supported by an editor capable of editing mathematical formula.

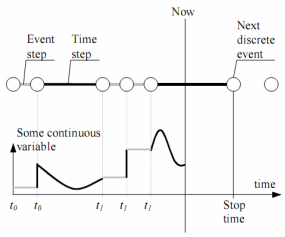
Project representation can be done in visual and textual form, respectively. These representations can be restored from each other.

A subset of UML for Real Time was chosen and extended to incorporate continuous behavior. The modeling language implemented in our tool supports two types of UML diagrams: collaboration diagrams and state chart (state machine) diagrams with some changes. In collaboration diagrams we have added unidirectional continuous connections between objects (capsules in UML-RT) and the corresponding interface elements – input and output variables. UML state charts are made hybrid: a system of algebraic-differential equations over variables (interface or object's internal ones) can be associated with each simple or composite state. To make such an UML-based model fully executable we have taken Java as a reasonably high-level language for defining data types and data transformation.

As an example we consider an airplane ejection process. This model has a hybrid structure. The equations and states can be represented in our notation like in the lower left figure.



The catapulting will start when external signal 'Eject' occurs. The first state of catapulting is track moving with fixed velocity. The second state is free flight which will start after an arm-chair has left airplane and will stop either by parachute opening after delay *TauP*, or by arm-chair and airplane collision and destruction. The B-Chart of the device is shown in the left figure. The solution is calculated following the hybrid event calendar in the right figure.



Users are recommended to construct a Behavior-Chart. Thus, the solved system has fixed structure and smooth right-hand sides.

Three Equation Solvers may be suggested, which are ODE-solver, DAE-solver and AE-solver. By Equation Solver we mean a heuristic algorithm that chooses the simplest numerical software for solving a problem with user prescribed tolerance. All our Solvers are based on numerical software available in the Web, namely ODEPACK, Hairer, Norsett, Wanner and Hairer, Wanner collection, DASSL and some others programs.