

*DIFFERENTIAL EQUATIONS
AND
CONTROL PROCESSES
№ 1, 2011
Electronic Journal,
reg.Эл № ФС77-39410 at 15.04.2010
ISSN 1817-2172*

<http://www.math.spbu.ru/user/diffjournal>
e-mail: jodiff@mail.ru

Program Complex OpenMVL for Modeling Complex Dynamic Systems

Isakov A.A.

Saint-Petersburg state Polytechnical University,
Polytechnical str. 21, Saint-Petersburg, Russia,
E-mail: isak239@mail.ru

and

Senichenkov Y.B.

Saint-Petersburg state Polytechnical University,
Polytechnical str. 21, Saint-Petersburg, Russia,
E-mail: senyb@dcn.infos.ru

Abstract

This article is devoted to development of a program complex for modeling complex dynamic systems meant for free distribution. For a prototype of the developed program means, OpenModelica package has been chosen. The tool created is supposed to be used for training future professionals in the area of modeling as well as for solving investigation tasks.

Keywords object-oriented modeling, hybrid systems, OpenModelica, open project.

1. Introduction

Developers of program complex Matlab (corporation Mathworks [11]) offered a new concept of designing – Model-Based Design. Presently the offered approach has proved to be efficient [7].

Development of industrial technologies is impossible without scientific research being conducted simultaneously. In connection with this, creating an open research product open for criticism and analysis – a prototype of the industrial technology is topical.

Project making technology is based on modeling an «invariant» of the modeling means. Visual modeling means in this context are understood as complexes including the following components:

- (OO) graphical object-oriented modeling language;
- Simulation language for model research;
- Language for debugging -- graphical debugger;
- Graphical tools for model investigating (toolboxes).

As examples the following packages can be used

1. Matlab [11] is a hierarchy of program means of different languages: Simulink, StateFlow, SimMechanics, SimPowerSystem, SymEvents etc.
2. Dymola [9]. Based on the unified modeling language Modelica, one of the advantages of which is support of «physical» modeling.
3. Model Vision Studium [10]. Based on a single language for modeling hybrid systems Model Vision Language (MVL).

To provide for the quality of commercial visual modeling means and their improvement it is necessary to do scientific research. The object of research may be the modeling language and the mathematical models and new numeric models and specific tools and new numeric methods – everything that can effect the reliability of designing and the speed of modeling which can become an important factor for designing real time systems and large-scale systems.

In order to attract researchers attention to their projects, industrial environment developers:

- Promote academic licenses (limited user rights for the product) and/or functionally limited modeling means.
- Create free circulating software based on existing professional environment, OpenModelica [8] being an example.

The last thing, offered by OpenModelica developers, – to make the product open – seems to have the best prospects.

2. Aims and Goals of the work

In the framework of the given article we see it as our goal to create an open product OpenMVL similar to OpenModelica, to be used freely for scientific research and easily adapted for training future professionals in the sphere of modeling, extended and modified by users. In the package it is offered to use a single language for modeling hybrid systems Model Vision Language (MVL) [6]. In the language MVL the difficulties of modeling hybrid systems have been solved (the components now can be supplied with hybrid automata).

3. OpenMVL Structure

The program means being created:

- a. uses the language Model Vision Language in text form: from the language MVL the structures necessary for running the graphical component of modeling environment are excluded,
- b. retains the part of the language MVL, which provides component modeling of hybrid systems [5] (components with input/output¹ and with contacts², (Fig. 1).

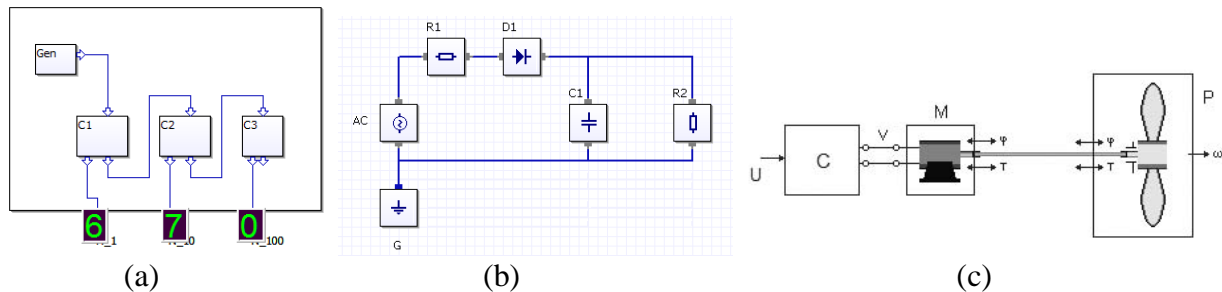


Figure 1. Classical examples of component modeling:

a) components with inputs/outputs b) components with contacts/flows c) components with inputs/outputs and contacts/flows.

- c. makes it possible to support the extendable library of numeric methods [5] in the form of switchable dynamic libraries (DLL).
- d. contains a simple enough test bench on which it is possible to demonstrate the results of the carried out numerical experiment (time and phase diagrams)

Application OpenMVL is a command interpreter (Shell), which allows the user to perform a number of basic commands: edit model files ; load model files into a package (a realized compiler of object-oriented modeling language is started); conduct numerical experiment in given time limits; build a diagram of the results obtained.

4. Immediate Tasks

4.1 Automatic formation of the combined (total) system of equations

Traditionally, there is a direct connection between the form of the mathematical model (the form presents the model in the modeling language) and numerical methods corresponding to the given form, i.e. Matlab package and visual environment package AnyLogic [12], induce the user to create calculation procedures for the right part in the programming language of a level which is low enough.

In the OpenMV environment as well as in MvStadium packaging the mathematical model is presented in the form of algebraic-differential equations $F(t, x, \frac{dx}{dt}) = 0; F \in \mathfrak{R}^n, x \in \mathfrak{R}^n$. This form is the most general one, so its application allows the user to present the system being solved in any convenient form. This freedom, though, limits the choice of program realizations of numerical methods: DASSL [3],

¹ used for the so-called «cause-consequence modeling» (a system with a clear one direction regulation connections)

² used for the so-called «physical modeling» (for example, electric chains)

DASPK [4]. Application of other methods presupposes transformation of the system into the way «understood» by the numerical method.

4.2 Choice of «canonical» form

Canonical form in the given context is understood as a form by which we can judge about the system resolution, also there exists a numerical determiner understanding this form without any further transformations. The larger is the number of «canonical» forms $\frac{dx}{dt} = f(t, x)$; $\frac{d^2x}{dt^2} = f(t, x)$; $A(x, t) \frac{dx}{dt} = f(t, x)$ and other used in the package, the more efficient is its work.

In conjunction with this a question arises what forms are to be used in modeling environments.

4.3 Formation of the combined (total) system

As the project OpenMVL is meant for modeling hybrid systems based on component modeling, a question arises concerning the algorithms of forming the total mathematical system combining the interconnected components (with input/outputs or with contacts), giving the possibility to use efficiently the existing numerical methods, for example take into consideration the scarcity of the system. No less important is a question of the possibility of the system to be written by the user (i.e. what its structure analysis can give) . (Fig. 2)

4.4 Re-establishing combined (total) system in the canonical form

One of the drawbacks of visual modeling is a lack of possibility of seeing the combined system in the usual mathematical form. For the systems of small size, such a form makes it possible to investigate and check the system created by the package. As it has turned out, «re-establishing» the system and the choice of form convenient for the analysis is not an easy job.

5. Results planned

As the result of the work we are planning to build a program modeling complex which can be used both for training the basics of object-oriented modeling and as a tool for the research in the sphere of professional environment (in particular, Model Vision Studium).

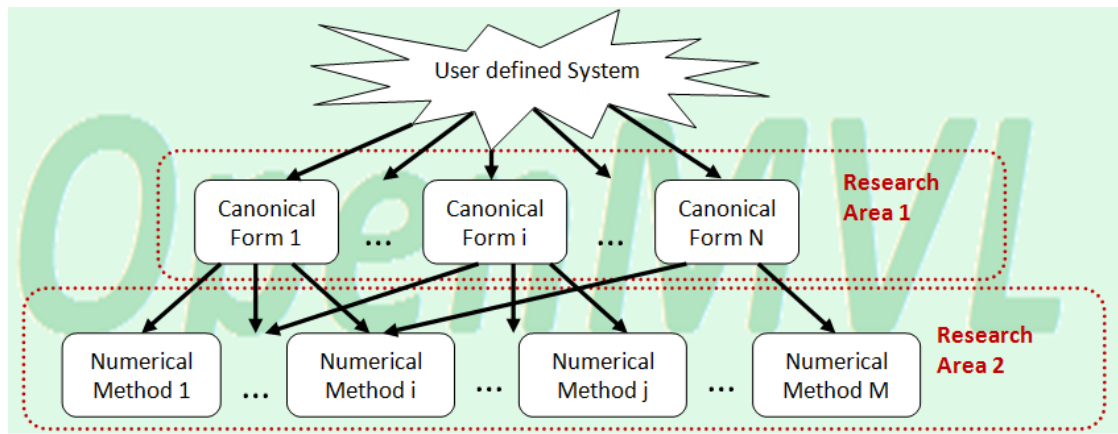


Figure 2. Scientific research areas

References

- [1] Ascher U.M., Petzold L.R., «Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations», 1998.
- [2] A.C. Hindmarsh, «ODEPACK. A Systematized Collection of ODE Solvers in Scientific Computing», R. S. Stepleman, editor, 1983.
- [3] K. Brenar, S. Campbell, L. Petzold, «Numeric Solution of Initial-Value Problems in Differential-Algebraic Equations», 1989.
- [4] P. N. Brown, A. C. Hindmarsh, L. R. Petzold, «Using Krylov Methods in Solution of Large-Scale Differential-Algebraic Systems», 1994.
- [5] Y.B. Kolesov, Y.B. Senichenkov. », “Modeling of systems. Dynamical and hybrid systems” (in Russian), – BHV-St. Petersburg, 2006.
- [6] Y.B. Kolesov, Y.B. Senichenkov “Modeling of systems. Practical modeling” (in Russian), – BHV-St. Petersburg, 2007.
- [7] "Using MATLAB and Simulink for modelling control systems" (in Russian), (Softline Group, MathWorks): http://seminars.softline.ru/it_page.php?id=291&programma=1.
- [8] <http://www.openmodelica.org> –OpenModelica’s site.
- [9] <http://www.modelica.org> –Modelica’s site (Dymola).
- [10] <http://mvstudium.com> –MvStudium Group’s site.
- [11] <http://www.mathworks.com> –Matlab’s site.
- [12] <http://www.xjtek.com> –AnyLogic’ site.