УНИВЕРСИТЕТСКИЙ
КОНСОРЦИУМ

УН Университетский научный журнал

**Информационные технологии**

№ 8 (2014)

# Rand Model Designer in Manufacturing Applications

*Yu. B. Senichenkov*
*St. Petersburg State Polytechnical University*

*Yu. B. Kolesov*
*The Federal center of information of Accounting chamber of Russian Federation*

# Использование визуальной среды Rand Model Designer для разработки промышленных приложений

*Ю. Б. Сениченков*
*Санкт-Петербургский государственный политехнический университет*

*Ю. Б. Колесов*
*Федеральный центр информатизации Счетной палаты Российской Федерации*

**Keywords:** *modeling; simulators; training; real time computer systems; large- scale systems, hybrid systems; visual modeling languages*

*Equation based computer modeling is traditionally used for investigation, simulation, and optimization of a test subject at an early stage of designing. For this purpose a computer model may be a real time embedded component of a hardware-software system. Manufacturing models, especially for hardware-software systems, are large-scale models, usually designed by teams. Such complex projects demand using special tools for joint operation. Designing large-scale models in business competition in specified time also needs using special tools for modeling and simulation such as a family of tools developed by MathWorks Inc., for example. In this paper we present a new tool named Rand Model Designer and demonstrate its properties and features using examples of manufacturing models.*

**Ключевые слова:** *моделирование; виртуальная реальность; тренажеры; обучение; системы реального времени; визуальное моделирование*

*Компьютерное моделирование, в основе которого лежат математические модели, традиционно используется для исследования, оптимизации и проектирования реальных объектов. Компьютерные модели также часто используются как встроенные компоненты программно-аппаратных систем реального времени. Промышленные программно-аппаратные системы разрабатываются большими коллективами и требуют специальных инструментов разработки. К ним относятся, например, продукты корпорации MathWorks. В этой статье представлена еще одна среда разработки, Rand Model Designer, обсуждаются ее возможности для проектирования промышленных систем.*

## 1. Introduction

Rand Model Designer (RMD) is an object-oriented modeling tool [6, 10, 17]. It considers the Unified Modeling Language [15] as a standard and accommodates it for modeling multi-component complex dynamical systems with different physical nature components using modeling technologies with «oriented» and «non-oriented» blocks [12, 13]. A slightly modified variant of the UML state machine is a basis of RMD's visual notation for hybrid systems and planning an experimental graphical language [16]. Hybrid systems used inside «oriented» and «non-oriented» blocks have no restrictions inherent in the Modelica language. Simulation in RMD also has some particularity. Well-known numerical methods [5, 7, 8, 14] were modified taking into account the structure of the solved system in order to increase computation speed and cut down memory usage.

We will consider two types of manufacturing models to demonstrate Rand Model Designer's features and benefits. Rand Model Designer (RMD) may be used for traditional prototyping complex engineering systems at an early design stage and to design embedded real time applications.

## 2. Modeling Technologies of Rand Model Designer

RMD uses hybrid system to model event-driven systems with multiple-mode behavior (Fig. 1). A hybrid system is considered as generalization of a classical continuous dynamical system [16]. RMD's visual notation for hybrid systems is called Behavior-Chart (Fig. 1). Local continuous behaviors (modes) and discrete actions associated with them alternate one-by-one to generate a hybrid system trajectory. A hybrid system trajectory, in general, is a sequence of solutions to systems of algebraic-differential equations (Fig. 2).

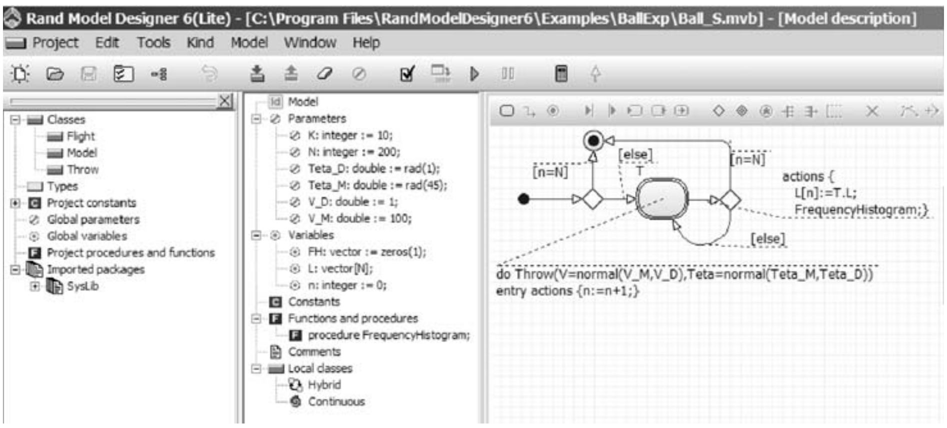


Fig. 1. RMD B-CHART with do-activity in the form of algebraic-differential equations

RMD's modeling language called the Model Vision Language (MVL) allows modeling with oriented ("causal" modeling, Fig. 3) and non-oriented ("physical" modeling, Fig. 4) components with hybrid behavior [13].

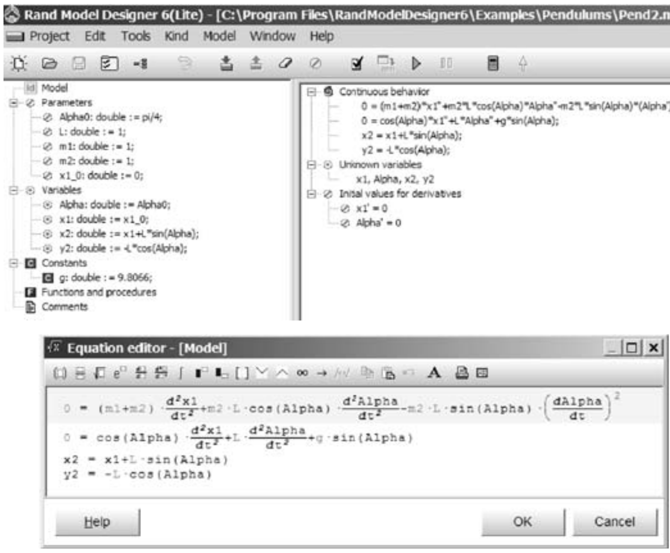


Fig. 2. RMD's Equation Editor

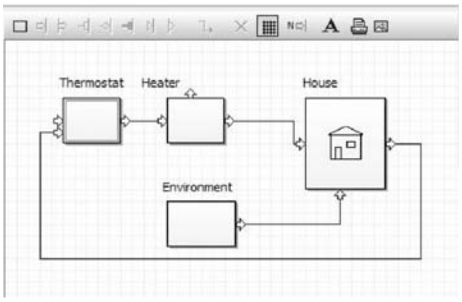


Fig. 3. "Causal" modeling block-diagram

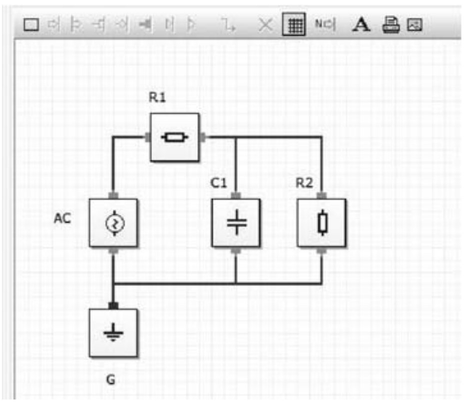


Fig. 4. "Physical" modeling block-diagram

RMD builds two sorts of executable models:

• A visual model that is the Windows stand along application used for testing, visual debugging, carrying out computational experiments, and processing results;

• A "hidden" model which is the Windows dynamical linked library (dll) used as an embedded interactive application.

Hereby we can summarize:

1. RMD is oriented on well-defined mathematical models, which are "event-driven set of algebraic-differential equations". The dimension, type, and numerical properties of such systems depend on the current model mode.

2. RMD's hierarchical component diagrams and B-Charts allow designing complex large-scale models in an intuitive manner.

3. RMD's modelling language maintains technologies for causal and/or non-causal modelling.

RMD is used for scientific research and teamwork computer-aided large-scale systems designing. In both cases object-oriented modelling is helpful. Classes worked out for a certain project and classes from libraries that already exist are a basis for any complex model.

## 3. Modeling-Based Designing with Rand Model Designer

modeling-based designing with executable specifications allows modifying, step by step, a workable model and comparing model behavior to the one required by specification through computational experiments (MathWorks Inc.). Specifications written in a high level graphical modeling language are also well-adapted for communication among developers.

RMD is good for modeling-based designing. It has high level languages for modeling, testing, debugging, planning experiments. There are special visual instruments for parametric optimization (Fig. 5), sensitivity analysis (Fig. 6), and statistical experiments (Fig. 7).

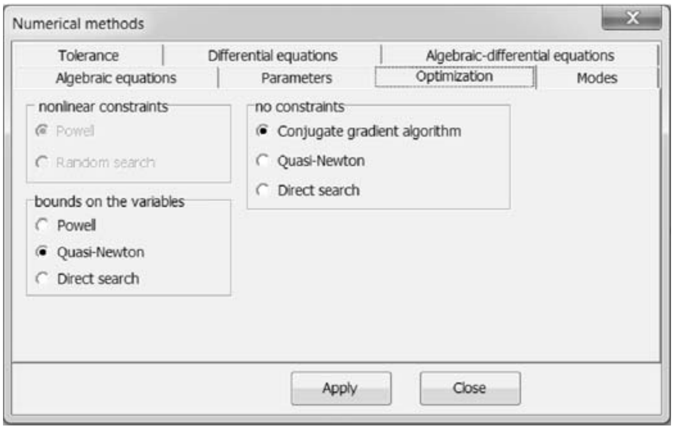


Fig. 5. RMD's numerical methods for optimization

RMD modeling language makes developing complex models easier using "from the simple to the complex" technology. At the first step a set of isolated models is developed. Then these isolated systems may be transformed into a hybrid system. Isolated hybrid systems turn into blocks with external variables ("input", "output", "contact", "flow") to be used as components of a complex hierarchical model.

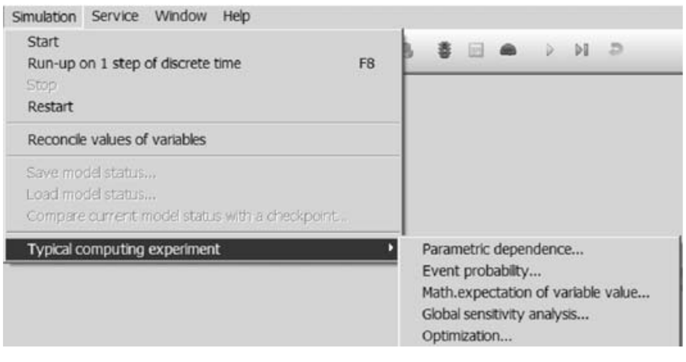


Fig. 6. RMD's tools for computing experiments

RMD's B-Charts are used not only as a graphical specification of hybrid systems but as a visual language for computational experiments planning (Fig. 7). Indeed, a designed model may be automatically transformed into a new class in RMD. A computational experiment is described by hybrid automaton with local behaviors that are instances of this class with particular values of parameters.

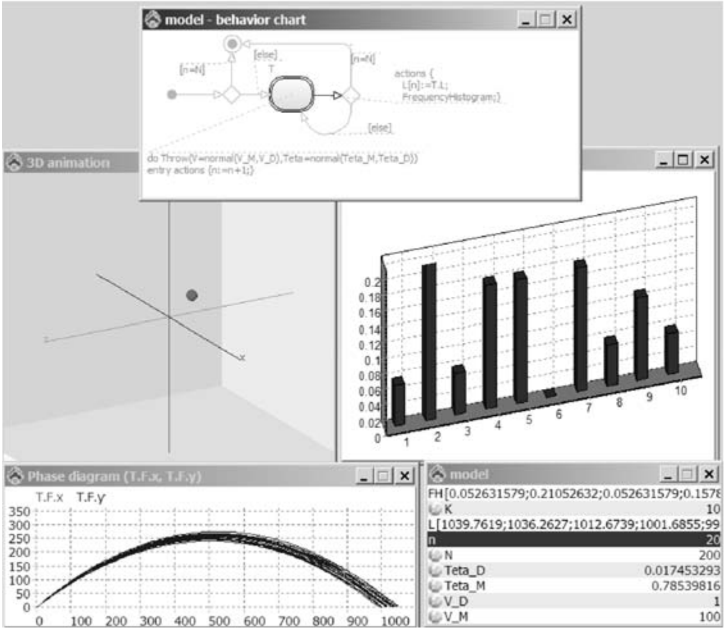


Fig. 7. RMD’s statistical experiments

## 4. RMD’s Numerical Software

RMD’s numerical library contains various well-known numerical methods to solve algebraic, ordinary differential and algebraic-differential systems of equations [16]. RMD tries to detect and consider characteristics of model equations (linearity, block structure, sparseness, and so on), decreasing the number of unknowns, increasing calculation accuracy, and acceleration of calculating speed. RMD has a set of tools to analyse numerical properties of solved equations (it is possible to calculate condition numbers, eigenvalues of the Jacobi matrix while debugging), to process experimental data and automatically compare experimental results.

User-friendly mathematical and B-charts Editors facilitate developing models with complex behavior and simultaneously complicate the designing numerical library for modeling and simulation component models, especially for “physical” modeling. A B-chart generates a sequence of systems of algebraic-differential equations with a different size, structure, and numerical properties for a hybrid system. Algebraic-differential equations, in their turn, may take a form of linear and non-linear algebraic equations, differential equations, high index algebraic-differential equations [1, 7, 8, 14]. However, modern DAE Solvers are oriented on a single problem with known properties. All preliminary work should be crafted individually by specialists beforehand. Strange as it may seem, there is no universal numerical software intended for “hybrid” DAE. Hybrid DAE Solver should automatically analyze every new system from a job stream generated by a B-chart. Analysis usually has two stages that include «symbolic» (a priori) and «numerical» (a posteriori) analysis. Symbolic (structural) analysis allows detecting structural singular systems (they have no solutions for all possible values of parameters), calculating and reducing an index (it needs symbolic differentiation) for algebraic-differential equations. Run time analysis implies estimation of numerical solution accuracy for algebraic systems using condition numbers, stiffness for differential equations and so on. It is important to emphasize that it is necessary to ensure the optimal calculations on the whole trajectory for event-driven systems but not for a separate problem. We have just the same problem for computational experiment, testing, and parametric optimization when it is important to minimize the total time for whole experiment.

There is a special block named Analyzer in RMD for system properties analysis. Symbolic analysis should be done on run time for “physical” models with hybrid behavior with the help of very fast algorithms.

## 5. Ballistic Missile Flight

Ballistic missile designing is an example of traditional computer modeling to prototype complex engineering systems at an early design stage.

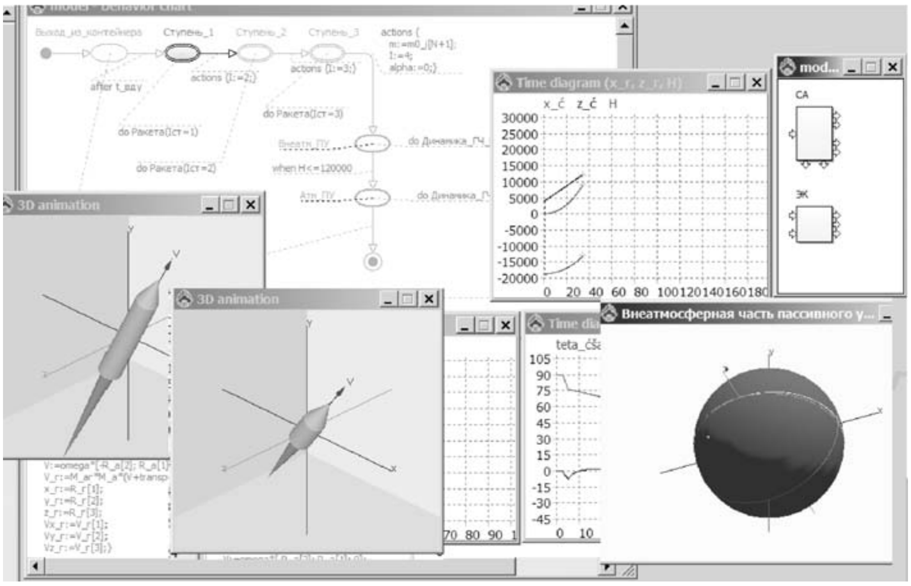


Fig. 8. Three stage ballistic rocket flight: going into an orbit

States of the main B-chart (Fig. 8) describe a boost phase (1-st, 2-nd, and 3-thd rocket stage running), midcourse stage, and terminal stage [2]. Stage changes are associated with changing values of parameters or/and changing a system of equations. The main B-Chart is hierarchical: each of stage states has hybrid behavior and its own B-Chart in its turn. Nested B-Charts describe different control modes of a stage stop (cutoff of thrust rocket engine command, fuel burn-out). When Object-Oriented-Modeling (OOM) is used, complex model behavior is divided into rocket motion, solid-propellant rocket engine dynamics, and control. Each mode is represented by its own class. This decomposition makes the model intuitively obvious and simplifies testing and debugging. In addition, OOM allows reusing debugged models. In our case models of International Standard Atmosphere and Gravitational Field Model for the Earth are imported from the libraries that have been previously worked out.

## 6. Complex Simulator Desing

using RMD to design complex simulators (Fig. 9,10) at “Transas Technology” (www.transas.com) is a good example of the second type industrial modeling [9].

Fig. 9. Simulator operating position

Components of the prototype system with complex switching logic (mechanical, electrical, hydraulic, and pneumatic, etc.) are described by variable structure systems of nonlinear differential-algebraic equations with a dimension of up to ten thousands. Obviously, building such a model manually is almost impossible. The only way to do it is to use Object Oriented Modeling


Fig. 10. Visual presentation of a wharf crane

for designing the applied class libraries. A typical mechanical, electrical, hydraulic, and pneumatic active dynamic object is described as a class or a hierarchy of classes. RMD's Object Oriented Modeling allows inheritance and redefinition of classes, independent design and import of packages by users. A complex model is assembled from instances of classes and used as built-in modules in simulators. Built-in modules should work in real time. Supporting real time environment needs fast numerical methods and algorithms to analyze, transform and build the current system on run-time. Symbolic transformations and simplifications of equations before numerical solution may decrease the total run time if and only if they are fast enough. As a result, an acceptable compromise has been found.

Designing complex simulators reveals a few unsettled problems. Assembling a new device with the use of standard components sometimes leads to structural singular equations for the whole device or causes difficulties during numerical solution. Such kind of errors are very difficult to detect. The only practical way is run time debugging. RMD's debugger visualizes structural matrix, calculates the Jacoby matrix, its eigenvalues for the current system of equations. This information helps to detect errors. Additionally, interactive step by step discrete actions debugging for validation switching logic is available.

## 7. Strategic Audit

Let us consider another hardware-software system named «Strategy - Chamber of Accounts» developed by St. Petersburg Institute for Informatics and Automation of the Russian Academy of Science. The system is intended for strategic audit of socio-economic development of the Russian Federation. Complex selection of development strategy for the country or a region is based on the results of simulation. A stochastic experiment with a model of a stock of projects and estimated measures is carried out during simulation to achieve macroeconomic goals. On the one hand it is impossible to build a model of a stock of projects and estimated measures manually because all the information needed for a certain project is stored in a huge project management database («Spider Project») and users are non-specialist in programming or in modeling. Meanwhile a macroeconomic model may be developed a priori manually and customized with the help of parameters for a certain region. This problem was solved with OOM. A macroeconomic model is considered as a class of «Macroeconomic» Package. Special classes such as "Job", "Computer experiment" and so on have been developed for modeling a stock of projects and estimated measures. A special tool has been designed which is able to transform data base specification of a model of a stock of projects and measures to build a corresponding model into RMD modeling language automatically with special applied library classes for the integration software complex «UPE&PlanDesigner» (developed by «SoftProm»).

Fig. 11 illustrates this technology. The shown model is an object of the "Macroeconomic" class. The "Macroeconomic" class uses inner classes that RMD specification has generated automatically with the use of the data base specification. RMD builds an embedded executable model, runs it, and exports results for visualizing in «UPE&PlanDesigner».

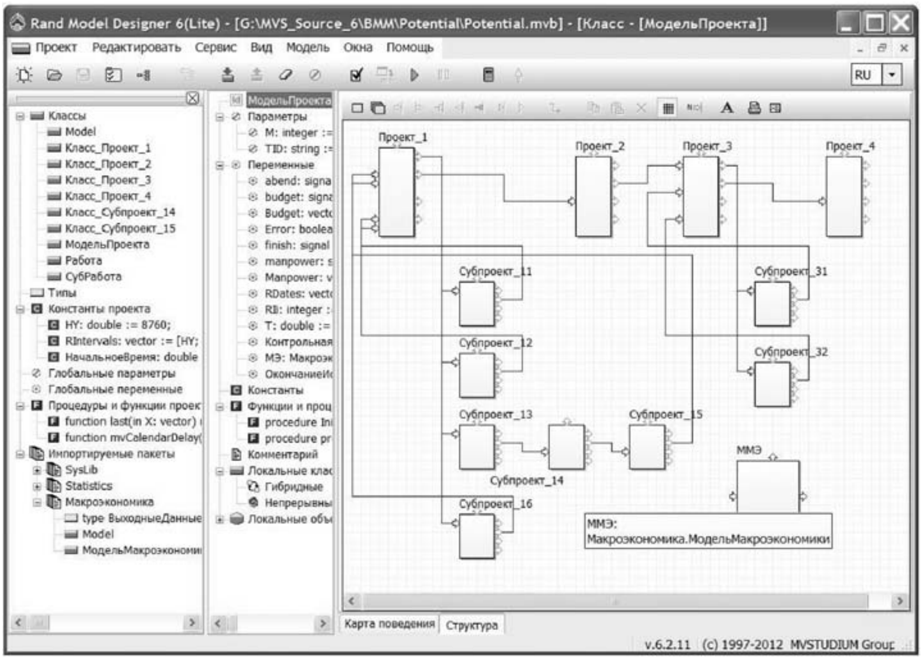


Fig. 11. RMD's macroeconomic model

## 7. Using RMD in Education

Virtual laboratories for schools and universities are a special type of simulators (Fig. 12).

In the article [4] professor S.V. Biryukov wrote: "One of the most difficult problems in teaching physics is real-world experimental skills training. Real world experiments are often used, but they are too expensive. Instead, computer simulations can be helpful in many cases. A computer model must be cheap, variable and present an initial example for students to make their own model.

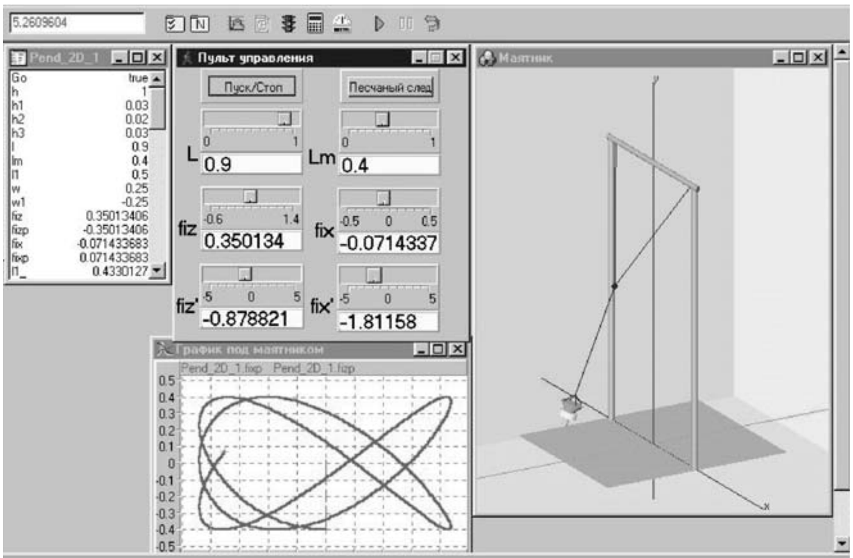


Fig. 12. Professor S.V. Biryukov's model — a test bench for schools

Menu, Help, Tutorial, and Teaching materials must be in the student's native language. All these requirements are satisfied by Model Vision Studium (Former name for RMD)."

Experiments with computer models and computer models designing help students to understand real world phenomena better. Simulation is an active method of learning and teaching not only physics.

Several virtual workshops for the course of general physics (mechanics) developed in Model Vision Studium are presented in [4]: "Motion investigation using Atwood's machine", "Bullet velocity determination by kinematic and dynamic methods", "Measuring the moment of inertia of a bicycle's wheel using dynamic and oscillation methods", etc.

Professor S.V. Biryukov believes: "These workshops are very similar to real world workshops in the General Physics Labs of the Moscow Pedagogical State University and are used as an additional home study for campus students. Furthermore, it can be used in distance learning of physics as it was initially developed for an open educational portal. "Physical pendulum oscillations" is a workshop with real-time simulation and at the same time a simple real world experiment with a hand-made physical pendulum – a modified additional computer mouse. Modification is so simple that it can be done by many students and thus it can be recommended as a real-world and real-time computer experiment in distance learning or just for fun. "

## 8. Model Converting

There are a lot of tools to model and simulate complex dynamical systems [3]. Modern tools based on the Object Oriented Modeling approach are more advanced than the old ones but have no extensive and infallible applied libraries. To overcome this typical disadvantage it is possible to use model convertors (preprocessors) from one modeling language to another.

RMD has a special library (SysLib) with Simulink-like components (Fig. 13–14). They have been written manually and intended for Simulink's users taking their first steps in RMD.

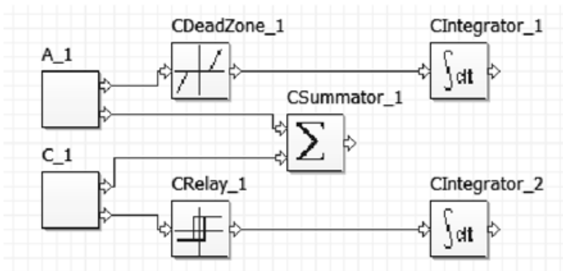


Fig. 13. RMD's Simulink simulator

Practice shows that a convertor from Simulink to RMD models (and vise-versa) is required not only to reuse the applied libraries but for comparative study of separate models developed with the help of different technologies. Model converting takes advantage of using both tools for simultaneous modeling.

RMD convertor transforms Simulink specification of a model into MVL language through an extended SySLib library for Simulink blocks.

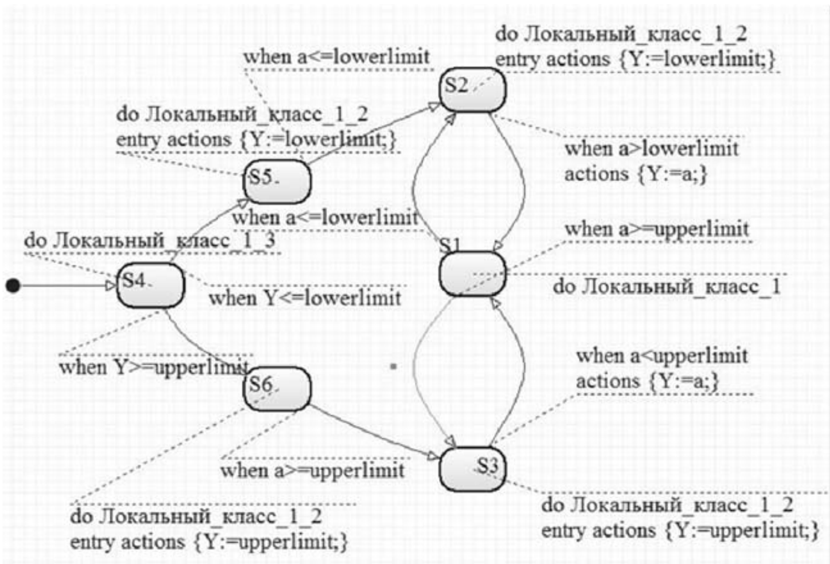


Fig. 14. RMD's "Integrator" simulator

COMMENT

The first version of this paper was discussed at MIM — 2013 conference.

## REFERENCES

1. Ascher U.M., Petzold L.R. (1998). Computer methods for ordinary differential equations and differential-algebraic equations, SIAM.

2. Benevelsky C.B., Kolesov Yu. B. (2009) Modeling. Object-Oriented Modeling for exterior ballistics problems. St. Petersburg, STU, 126 p.

3. Breitenecker F., Proper N. (2009). Classification and evaluation of features in advanced simulators. Proceedings MATHMOD 09 Vienna, Full papers CD Volume.

4. Biryukov S.V., Guskov D.N., Fedyanin V.V. Visual Simulation of Physical Processes in Model Vision Studium, (2005). Proc. of the Int. Conf. "Informational Technology in Education", Moscow

5. Brown P.N., Hindmarsh A.G., Petzold L.R. (1993). Using Krylov methods in solution of large-scale differential-algebraic systems. Technical report. Numerical Mathematics Group, UCRL-JC-113507 (https://computation.llnl.gov/casc/nsde/pubs/221215.pdf)

6. Fritzson P. (2006). Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. Wiley-IEEE Press.

7. Hindmarsh A.C., Peter N. B., Petzold L.R. (1993) Using Krylov Methods in the Solution of Large-Scale Differential-Algebraic Systems.

8. Hindmarsh A.C. (1983), ODEPACK. A Systematized Collection of ODE Solvers in Scientific Computing, R. S. Stepleman, editor.

9. Kiptiliy D. B., Kolesov Yu. B., Lebedev D. V., Senichenkov Yu. B., Tarasov S. V. (2011) Object-oriented approach to designing cargo shipping operations simulator. Proceedings of SCM MEMTS, Saint-Petersburg.

10. Kolesov Yu. B. (2004) Object-Oriented modeling of complex dynamical systems. St. Petersburg, STU, 238 p.

11. Kolesov Yu. B., Senichenkov Yu. B. (2006, a) Modeling of Systems. Dynamical and Hybrid Systems. St. Petersburg, BHV, 224 p.

12. Kolesov Yu. B., Senichenkov Yu. B. (2006, b) Modeling of Systems. Object-Oriented Modeling. St. Petersburg, BHV, 186 p.

13. Kolesov Yu. B., Senichenkov Yu. B. (2007) Modeling of Systems. Practical Work on Computer Modeling. St. Petersburg, BHV, 352 p.

14. Li S., Petzold L. (1999) Design of new DASPK for sensitivity analysis. S. Technical Report, Department of Computer Science, University of California Santa Barbara, (www.cs.ucsb.edu/~cse/Files/LiPetz99.ps)

15. Rumbauth J., Jacobson I., Booch G. (2005). The unifiedmodeling language. Reference manual. Second edition. Addison-Wesley.

16. Senichenkov Yu. B. (2004) Numerical Modeling of Hybrid Systems. Practical Work on Computer Modeling. St. Petersburg, STU206 p.

17. Tiller M. (2001). Introduction to physical modeling with Modelica. The Springer International Series in Engineering and Computer Science.

**Сениченков Юрий Борисович**
Доктор технических наук.
Научная отрасль:
Санкт-Петербургский государственный политехнический университет,
профессор кафедры распределенных вычислений и компьютерных сетей,
Институт информационных технологий и управления
E-mail: senyb@dcn.icc.spbstu.ru
**Колесов Юрий Борисович**
Доктор технических наук.
Научная отрасль:
Федеральный центр информатизации Счетной палаты Российской Федерации,
ведущий научный сотрудник.
E-mail: ybk2@mail.ru