News from Rand Model Designer for Industry and Education

Y. Kolesov,* D. Inikhov,*
Y. Senichenkov, A. Isakov****

**MvSoft, Moscow, Russia, (e-mail: ybk@mail.ru)*
***Peter the Great Saint-Petersburg Polytechnic University, St. Petersburg, Russia, (e-mail: senyb@dcn.icc.spbstu.ru)*

Corresponding author-Yuri Senichenkov, (e-mail: senyb@dcn.icc.spbstu.ru)

Abstract: For the first time Rand Model Designer *(http://www.mvstudium.com)* was represented in the series of MIM-conferences on MIM-2013 (Senichenkov Yu. (2013)). Rand Model Designer is a modern tool for modeling and simulation hierarchical multicomponent event-driven dynamical systems. It has UML-based object-oriented Model Vision Language for designing dynamical and hybrid systems using modification of State Machines, large-scale multicomponent systems: control systems with «inputs-outputs», «physical» systems with «contacts-flows», and (new!) variable structure component systems, particularly «agent» systems. The new tool «Visual Debugger» is available now and there are new modifications of numerical solvers for sparse systems: it is possible now to try to reorder each solved system to block-triangular form with the help of Tarjan' algorithm for acceleration of execution speed.

Keywords: object-oriented modeling, dynamical and hybrid systems, components with oriented and non-oriented links with internal hybrid automata, component models with variable structure.

1. INTRODUCTION

Universal visual tools purpose for designing multicomponent continuous, discreet, and hybrid systems using unified graphical representations. Rand Model Designer Graphical Editor handles (Kolesov Yu. B (2013)):

- usual mathematical notation for systems of equations for continuous «do-activates» of state machines (algebraic, differential, and differential-algebraic systems);
- behavior-charts (a variant of UML state machines without parallel activates) for event-driven discreet and continuous dynamical systems;
- hierarchical component diagrams with static and dynamical open components with «input-output» or «contact-flow» connectors.

Unified (based on standards) graphical notations are preferred among possible. An environment transforms them to large-scale systems of algebraic, differential, differential-algebraic equations in different canonical forms, but it is not enough for getting unified current final system for all tools, because each one transforms and simplifies it additionally in its own manner. After that, numerical or symbolic Solver lunches a final system.

It is important to note that role of symbolic transformations and symbolic Solvers increases in traditional numerical environments for modeling. At the same time, modelling problems become important and for designers of mathematical tools such as Maple, Mathematica (for example see http://www.maplesoft.com/products/maplesim/).

Numerical solvers of differential and differential-algebraic equations are unified in a manner because there are only few open-source authoritative packages (http://www.netlib.org/) and just they are used mostly. Striving for unification is current trend, but it is still a long way off.

Comparative analysis (Breitenecker F. (2009), Martin-Villalba C. (2014)) of visual environments for modeling and simulations of complex dynamical systems based on criterions: «universality», «commonality» and «object-oriented approach» demonstrates that only Rand Model Designer is universal and unified object-oriented tool now, if «unified» and «object-oriented approach» mean following UML what is «de facto» standard:

- «Simulink+Toolboxes» (http://matlab.ru/products/simulink) does not object-oriented tool;
- using hybrid system in Modelica is limited in comparison with UML state machines ((Tiller M. (2001), Fritzson P. (2011));
- it is very difficult to realize «physical» modeling in AnyLogic (http://www.anylogic.ru/) and Ptolemy (http://ptolemy.eecs.berkeley.edu/ptolemyII/).

UML (Rumbauth J. (2005)) was developed as standard for discrete systems and then was extended without extensive discussion on continuous ones. It will be instructive to design similar standard for complex dynamical systems, and for their unified libraries of devices in the wake of the agreement.

The special instruments for debugging and testing, including unified set of test is very important for tools used in industry.

Finally, it is yet important using visual environments for modeling and simulation in education: lite versions for firsthand acquaintance with computer modeling and choice of profession in schools; standard versions for training in modeling courses in universities; professional versions for training in industry.

2. MODELS WITH VARIABLE STRUCTURE

All modern languages support designing even-driven continuous dynamical systems. There is unified standard for hybrid systems in the form of extended version of UML' state machines (http://www.uml.org/). The extensions touch on hybrid time and continuous do-activities in the different forms (http://www.mathworks.com/discovery/finite-state-machine.html; http:www.mvstudium.com) used on a par with discrete time and discrete activities (Fig. 3-4).
Hybrid systems demonstrate changeableness modeling object behavior but complex dynamical systems can change and their structure too.

Changing the number of independent «agents» (open subsystems with constant structure communicated on-stream) is elementary example of systems with variable structure - queuing systems with variable number of servicers and clients.

Case that is more complicated is component queuing system with behavior described by continuous equations. Changing number of services (equipment failure) or number of clients force rebuilding of current system of equations on run-time, that is labor-consuming action especially for models with «contacts-flows» connectors of components.

Let us consider an electric circuit with power supply device, load with constant and dynamic components, and rectifier (Fig. 1-2).

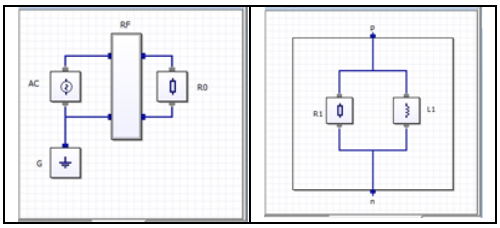

Fig. 1. Constant load – resistor R0 (left) and dynamic load (right).

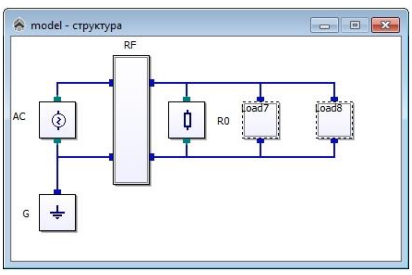

Fig. 2. Circuit with constant and dynamic Load.

A new variant of load is generated by command *new(Load)* associated with external transition of behavior-chart (Fig. 3) on interval [0, t] with random value t, distributed by exponential law with average T. The new object of class «Load» gets new random values for parameters: lifetime (Tend), resistor (RR), and inductance (LL), distributed by normal law. New load is placed on the diagram (Fig.2), new current system of equations is generated, analyzed, and model is going on until object «Load» will be destroyed (*destroy(Load)*) after Tend.

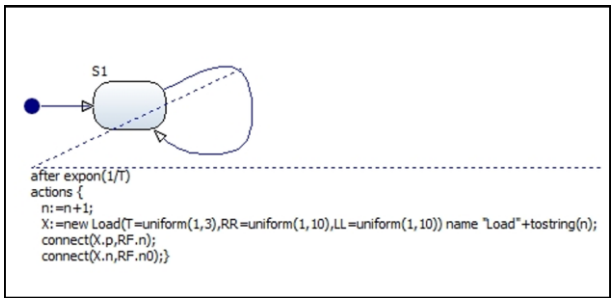

Fig. 3. Generating a dynamic load by behavior-chart

3. VISUAL BEBUGGER

A sequence of events changes behavior of hybrid system (Fig. 4): continuous do-activates (systems of equations) interchange discrete actions (procedures written on high-level language).

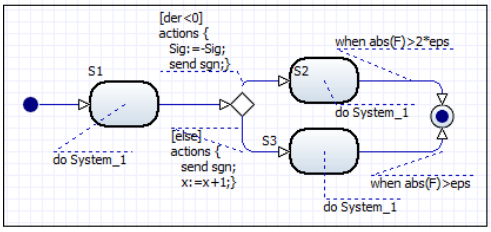

Fig. 4. Behavior of a simple model

By selecting interest events user can form a list of breakpoints for continuous and discrete behavior with breakpoint conditions: time, conditions, transitions, states, signals (Fig.5).

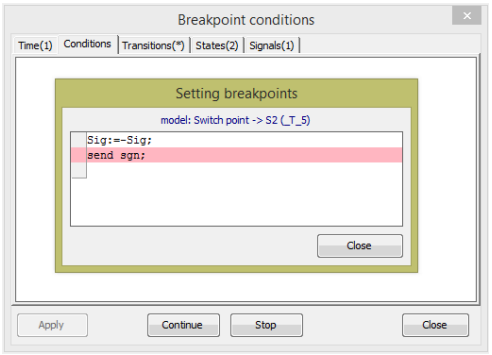

Fig. 5. Breakpoint conditions for hybrid automaton.

Getting additional important information about current solved system of equations is possible with the help of services (Fig. 6): to show solved equations in usual mathematical form, its structure (structural matrix), Jacobi matrix and its eigenvalues.

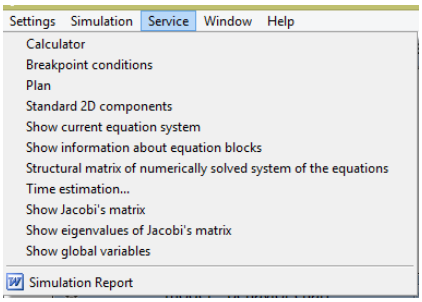

Fig. 6. Additional information about solved system.

Discrete part of hybrid behavior written in modelling language presupposes single-step debugging of sequence of instructions, procedures and functions. On run time, a new dialog window appears before executing marked instruction (Fig.7). It shows the control point and values of all used variables. User can execute instructions one by one, change values of variables, and come into subroutines and functions.

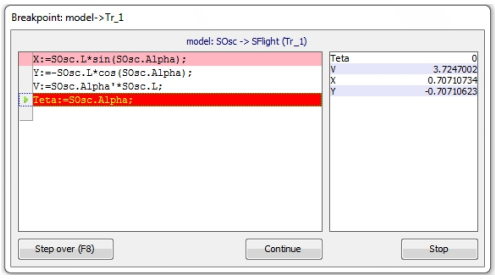

Fig. 7. Single-step debugging.

4. BLOCK-TRIANGULAG FORM OF FINAL SYSTEM

The structure of RND's Numerical Library (Fig. 8)

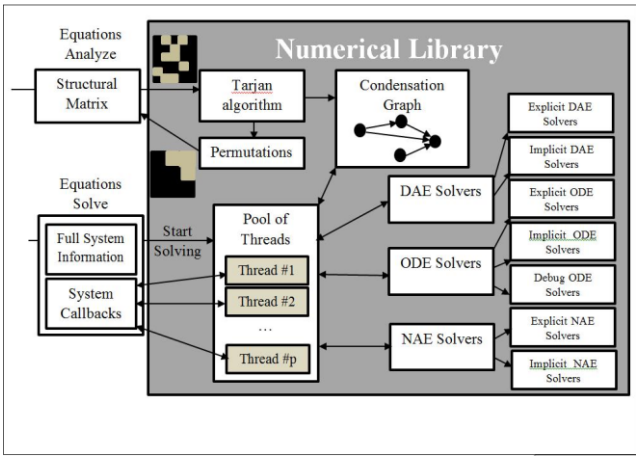

Fig. 8. RMD numerical library

and algorithm of interaction between Numerical library and Model Engine (Fig. 9) have been changed for implementing algorithm of building of final systems on run time (why it is advisable see (Kolesov Yu. B. (2014), Andrey A. Isakov 2015)).

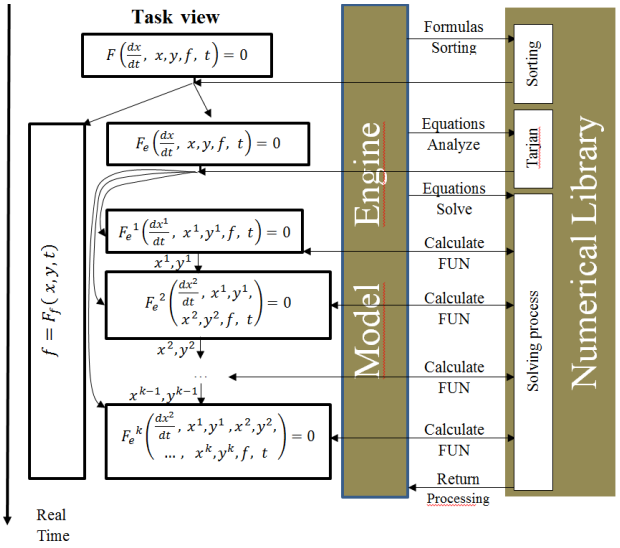

Fig. 9. Interaction between Numerical library and Model Engine.

On run time, if any new event has occurred, it is estimated necessity of rebuilding of current solved system and it is built if necessary: if transversal of new system' structural matrix is full, then structural analysis (for details see (2014, 2015)) of whole model is starting; otherwise (underdetermined system), a model is accepted as incorrect. Automatic searching a subset of unknowns leading to correct system is provided as option.

For correct systems:

1. Block triangular form of structural matrix is built if possible with the help of Tarjan' algorithm for searching strongly connected components of a graph.
2. Block triangular form may be used for parallel calculations. Strongly connected components (diagonal blocks) are associated with algebraic, differential, and differential-algebraic systems of equations. «Subtask» is a process of solving a system of equations corresponding to diagonal block with the help of suitable Solver. Subtasks may be executed sequentially or parallel. A thread pool for parallel execution is forming automatically using information about computer hardware (number of processors, number of kernels for a processor). The subtask readiness for execution is ascertained using condensation of a graph of Tarjan' algorithm (Fig. 11). Initially all nodes (subtasks) of the condensation are marked as «Unresolved». The calculation ends when all nodes become «Solved». If an «unresolved» node has no input edges or its input are «Solved» it becomes «Ready to start», otherwise - «Not ready to start». Running subtask has name «Solving», and after ending becomes «Solved».

Let us consider an example of multithreaded calculations for algebraic system of equations with triangular form shown in Fig. 10. The matrix diagonal blocks are associated with subtasks.

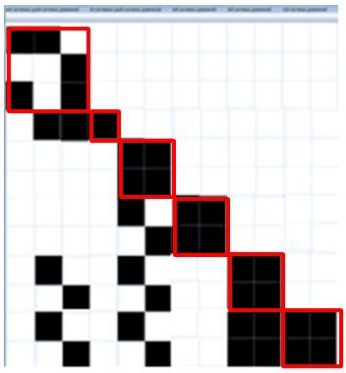

Fig. 10. Structural matrix after ending Tarjan's algorithm

The condensation with current labeling is shown in Fig. 11, and time diagram for threads - in Fig. 12.

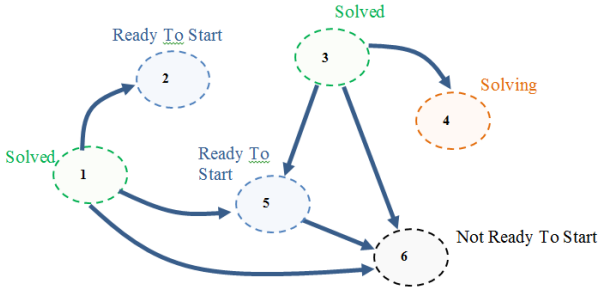

Fig.11. Current labeling of the condensation.

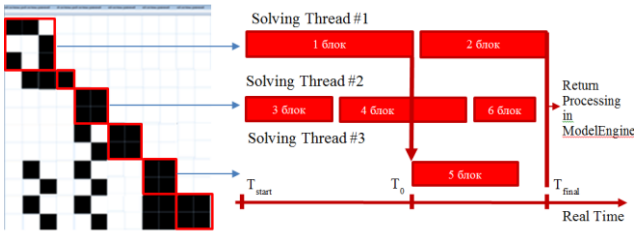

Fig. 12. Time diagram for threads.

A set of models developed by Transas company (Tarasov (2012)) (http://www.transas.com/products) was used for testing new approach. The results of numerical experiment for two most difficult problems are shown in Table 1 and Table 2. The computer used for calculations had four processors, so it was possible to create maximum four threads, but even if only one thread was used then total time of calculations decreased in two times.

Table 1. Product Tanker, Cargo System, about 2500 equations.

Thread Count	1	2	4
Old Numerical Library	8,53	-	-
New Numerical Library	4,72	3,80	3,67
Time difference	**44,67%**	**55,45%**	**56,98%**
Time on Right Part Calculation	2,44	2,47	2,47
Right Part Calculation Count	28964	28964	28964
Control Time	2,28	1,33	1,20
From one thread	100,00%	**58,33%**	**52,63%**

Table 2. Chemical Tanker, Inert Gas System, about 500 equations + 1000 formulas

Thread Count	1	2	4
Old Numerical Library	8,24	-	-
New Numerical Library	3,90	3,61	3,57
Time difference	**52,67%**	**56,19%**	**56,67%**
Time on Right Part Calculation	2,87	2,96	2,94
Right Part Calculation Count	23329	23329	23329
Control Time	1,03	0,65	0,63
From one thread	100,00%	**63,11%**	**61,17%**

5. USING RAND MODEL DESIGN IN EDUCATION

Rand Model Designer is used for training in schools, universities, and industrial enterprises. (Fig. 13).

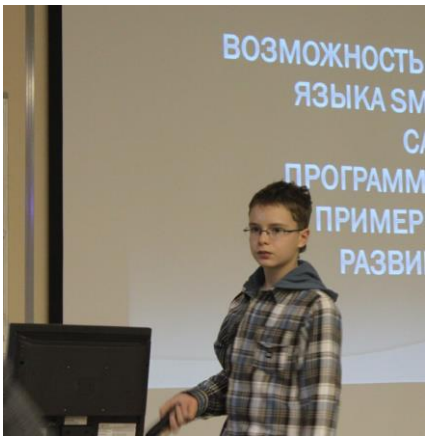

Fig. 13. Presentation of young programmer.

Schoolchildren interested in computer modelling may do their first steps with Rand Model Designer. School knowledge of physics and mathematics is enough for designing simple continuous and hybrid models: dancing ball, parachute jump, and so on, following by 2D- or 3D-animation are good and executable for schoolchildren tasks. Professional electrical and hydraulic libraries is possible to use for designing simple virtual experimental facilities and playing with them.

Rand Model Designer has specifics allowing its using in training in special manner in universities. Its primary function is illustration main concepts of «Basis of mathematical modeling», «Computer modeling of complex dynamical

systems», for example, cources in university education. Besides this it is possible learn basis of object-oriented modeling, because Rand Model Designer classes, inheritance, packagers follow UML standard. Blocks of special library «Syslib» is indistinguishable outwardly from the same Simulink blocks, so modeling in Simulink manner is possible. «Physical» libraries «electricity» and «hydraulics» have much in common with corresponding libraries written in Modelica language.

6. NEWS

Symbolic differentiation of functions now is available in Rand Model Designer.
Let us consider an example of inverse problem for mechanical system, when solution x(t) of differential equation is known and we want to find force F generated it (Fig.14). In this case it is necessary to differentiate x(t).

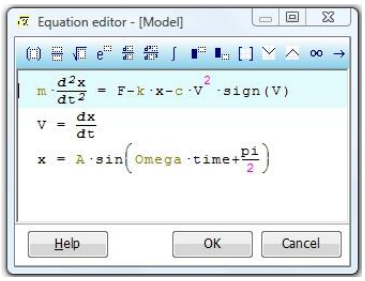

Fig. 14. Inverse problem for pendulum.

Until now, the following trick was used: additional differential equations (linear differentiator for example) were added to initial equations (Fig. 15) automatically.

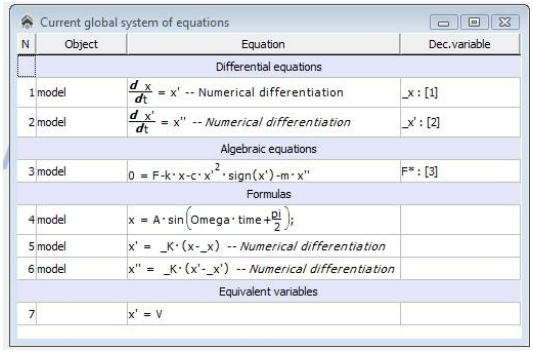

Fig. 15. Initial system with linear differentiator.

Accuracy and even qualitative accordance of solution in this case strongly dependы on initial conditions of added equations.
Now Rand Model Designer can build solution with the help of symbolic differentiation (Fig. 16).

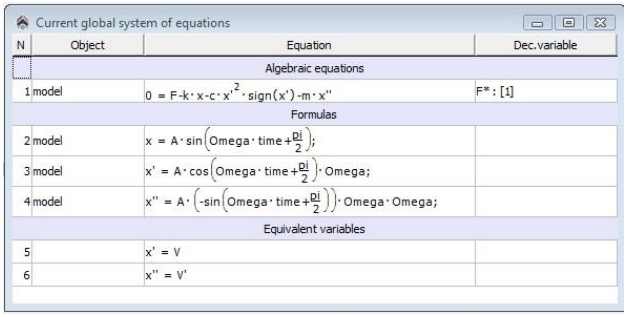

Fig. 16. Using symbolic differentiation for building solution.

REFERENCES

Senichenkov Y., Kolesov Y., Inikhov D. Rand Model Designer in Manufacturing Applications (2013) //, IFAC-PapersOnLine: Manufacturing Modelling, Management, and Control, Vol. 7, Part 1. P. 1572-1577, doi: 10.3182/20130619-3-RU-3018.00300.

Kolesov Yu. B., Senichenkov Yu. B. (2013). *Mathematical modeling. Component technologies.* St. Petersburg, Peter the Great Polytechnic University. 2013, - 223 pp. ISBN 9768-5-7422-3997-0.

Martin-Villalba C., Urquia A., Senichenkov Y., Kolesov Y. (2014). Two approaches to facilitate virtual lab implementation. Computing in Science and Engineering 16 (1) PP. 78 – 86, doi: 10.1109/MCSE.2014.29

Andrey A. Isakov, Yuri B. Kolesov, Yuri B. Senichenkov. (2015). A new tool for visual modelling – Rand Model Designer 7. IFAC-PapersOnLine 48-1(2015) 661-662.

Fritzson P. (2011) Introduction to Modeling and Simulation of Technical and Physical Systems with Modelica, 232 pp, Wiley-IEEE Press

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

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

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

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