Computer Modeling and Simulation in Higher Education

Alexander Yu. Glebovsky, Vladimir M. Ivanov, Yuri G. Karpov
Saint-Petersburg State Polytechnic University, Saint-Petersburg, Russian Federation
E-mail: glebovsky@rambler.ru

Abstract

The paper observes methods, techniques and tools that are instrumental for developing models and simulators aimed to be used as computer–based support for a large variety of disciplines in educational programmes of universities. The disciplines may belong to diverse fields of fundamental and applied sciences, engineering, economics and management, social sciences and humanities. Discussed are simulation objectives, teaching methods and advantages achieved by introduction of interactive models as didactic aids and learning tools in education at university level. Taxonomy of models and existing paradigms of simulation are reviewed together with corresponding simulation packages currently available. After reviewing the background the paper focuses on an innovative approach to simulation that is implemented in *AnyLogic* – a multi-purpose simulation tool designed and developed by a group of researches and academics from Saint-Petersburg Polytechnic University in Russia. Possibilities of various modeling techniques and approaches to simulation are illustrated by examples of actual models of different objects, processes, phenomena and systems implemented with the help of the tool in accordance with different simulation paradigms. As a conclusion it is supposed that evolving simulation technologies and next generation modeling tools facilitate a new progressive didactic trend both in e-learning and in education in general.

Modeling and simulation: origins, scientific objectives, prevailing modeling paradigms and application fields

Computer modeling as a major scientific direction can be traced back to the middle of the past century when the main modeling instruments of the time – analog electronic machines – were widely used in research centers around the world for solving problems that yielded definition by sets of algebraic-differential equations with time as independent continuous variable. Analog machines comprised a collection of various functional modules that served as building blocks for a researcher to manually compose an electrical model with. A connection board with multiple jacks was used to link parts of the model into a desired assembly by plugging connecting wires. A significant advantage of electronic analog models was that during runtime all components of a model were functioning concurrently – it was truly parallel operation. Oscilloscopes and graphic plotters served as output devices for analog variables presented by voltage values.

> *Advanced technical universities introduced Analog Modeling as a special discipline in their curricular. In Saint–Petersburg Polytechnic a team led by Professor Taras N. Sokolov has designed and implemented a series of new powerful analog machines that were successfully used to solve complex dynamic problems for industries. Later the team focused on digital computer technologies, has grown into a Chair of Informational and Control Systems and became one of the roots from which, eventually, the recent Engineering Cybernetics Faculty has grown from.*

Analog machinery has soon become outdated with the advent of more versatile digital computers. Yet the concept of building blocks assemblage was projected into digital environment where electronic functional blocks are substituted by software modules. The contemporary *Dynamic Systems* modeling paradigm inherits and refines this approach [1]. (An example follows below.)

A different conceptual basis for complex systems modeling – recently known as *System Dynamics* modeling paradigm – was also originated almost half a century ago. It has been introduced by Professor Jay W. Forrester (Massachusetts Institute of Technology) [2]. The concept owes much of its success and a long life to a specific graphic notation – *stock-and-flow diagrams* – that is an adequate and useful method of presenting cause/effect relationships in complex systems [3]. (Such diagram is shown below in an example of *System Dynamics* model.)

Another fruitful direction in the field of simulation – *Events–Driven Discrete Systems* – that has a very wide sphere of applications has been initiated over 40 years ago by Geoffrey Gordon (IBM) [4]. The underlying principles are also known as *flowchart modeling* or *network–based modeling*. The core of approach is the notion of *transactions*. The methodology is effective for a large variety of problems pertaining to so-called Mass Service Systems. GPSS (General Purpose Simulation System) inaugurated in October 1961 (IBM) was the first implementation of the concept and still remains one of the most popular modeling languages and simulation systems.

Finally, the newer conceptual approach – *Agent-Based Simulation* – stands apart from the three earlier system paradigms. The approach is unique in the sense that it provides an effective basis and is instrumental for solving problems connected with modeling and analysis of very complex systems (social, organizational, corporate structures, etc) that comprise certain elements described by their individual behavior. Typically, such systems can be strongly affected by accumulated results of individual behavior of elements (producing the so-called *emergent* effect). This is exactly the kind of problems and tasks for which *Agent-Based Simulation* is the only correct solution. On the other hand, as shown in [9] *Agent-Based Simulation* is a universal approach in the sense that for any given model based on *System Dynamics* or *Events–Driven Discrete System* paradigms it is possible to develop an equivalent *Agent–Based* model. Which approach is more effective in which case is investigated in [5].

A Table below summarizes the four dominant directions in digital simulation.

	Paradigm	Tools	Major Domains of Application	Comments/Links	Ref
1	**Dynamic Systems**	*Simulink, LabView, Easy5*	Physics, Mechanics, Engineering, Automation, Signal Processing	The origin of the paradigm stems from analog electronic computers of the 1950th decade	[1]
2	**System Dynamics**	*Vinsim, Powersim, Stella, iThink*	Business & Management	Originated in mid 50th of XX century. Site of the System Dynamics Society: [http://www.systemdynamics.org]	[2] [3]
3	**Events Driven Discrete Systems**	*GPSS, Arena*	Mass Service Systems, Information Sciences, Communication Techs	General Purpose Simulation System Introduced by IBM in 1961 [http://www.gpss.ru/index-h.htm]	[4]
4	**Agent-Based Simulation**	*Swarm, Repast*	Distributed Systems	Elements interact with each other and with the environment (emergent effect)	[5]

Table 1. A summary of simulation paradigms, corresponding software packages and major application domains

Paradigm–dependent and Task–dedicated packages vs. a Universal tool

Although each of the software modeling and simulation tools presented in Table 1 adheres to a concrete system paradigm and has a certain sphere of effective application these spheres cover very broad domains.

On the other hand, there are numerous other modeling languages and packages different from the above in the sense that they have a very particular orientation and are strictly dedicated to a certain narrow class of problems pertaining to some specific scientific field or concrete engineering task (below, when an example of computer networking protocols simulation is discussed a number of task–dedicated languages and specialized simulators are mentioned).

In contrast, a *universal* approach would mean that a certain simulation tool is neither *paradigm–dependent* (i.e. it equally support not only each of the four dominating paradigms but any combination of them in a single model), nor *field–oriented* (i.e. it is an efficient tool regardless of the application problem at hand). Recently there is at least one simulation tool – *AnyLogic* – that is claimed to possess such type of universalism. Features and possibilities of this versatile tool are discussed below and are illustrated by examples of models implemented with the technology in accordance with each of the four above mentioned paradigms. There are many points in favor of a universal simulation platform. But an obvious question/objection arises. Is it in principle possible to bypass the well known general rule: "the more universal is a tool – the less effective it turns out for any particular job"? It is difficult or even impossible for us to prove the opposite, but the interesting point is that there are some independent experts' opinions that in certain cases *AnyLogic* has indeed proved to be a more flexible and effective instrument than a competing task–dedicated system (one such case-study is mentioned below in connection with the problem of transport layer networking protocol simulation).

Taxonomy of models

Comprehensive reviews of modeling and simulation general ideas, paradigms, methods, problems, challenges and possible setbacks can by found in [6], [7] and [8]. But a starting point for pursuing this complex subject is, of course, possible approaches to classification of various models.

The following facets are essential as a basis for classification of models used in education and research. There are
- analytical/simulation models;
- analog/discrete/hybrid systems;
- concentrated /distributed system components;
- static/dynamic processes;
- deterministic/stochastic processes;
- linear/non-linear systems.

The type of a certain model depends on the studied phenomena or a concrete problem to be solved. Another influencing factor is the chosen level of abstraction that is determined in accordance with the simulation goals.

Stages of a model development, simulation and analysis

The main stages of model development are
- assessment of simulation goals;

determination of abstraction level;
conceptual model development (structural synthesis – system components and their inter-operation);
definition of parameters, variables and characteristics;
identification of model (calibration of parameters);
model validation;
model design, processes visualization and model animation.

Simulation experiment and analysis include stages of
dedicated experiments with the model,
reports generation (figures, tables, graphs, diagrams, etc),
comparisons, conclusions, decision making.

Fields of simulation application

Diverse application domains of simulation include (but are not restricted to)
physics, kinematics, mechanics;
theory of machines and mechanisms, machines dynamics;
astrophysics;
weather forecasts, meteorology;
biology, micro-biology;
medicine, epidemiology;
social sciences, social systems dynamics;
demographic dynamics, population growth, population migration;
urban dynamics;
financing, market analysis and forecasts;
strategic planning of corporate development;
transportation networks, traffic analysis and regulation;
logistics;
energy supply/distribution systems and networks;
automation systems; automatic control, design of controllers;
mobile robots behavior, artificial life;
informational sciences;
telecommunications and computer networks;
mass service systems.

Educational advantages of modeling and simulation

There is still no unanimous agreement in academic circles about advantages of e-learning methods and computer-aided teaching/learning technologies as a whole. Along with convincing arguments in favor of rapid and comprehensive introduction of such technologies on a wide scale, some valid skeptical opinions are also expressed. Not to indulge into pros and contras of the matter and relevant didactic issues we here restrain this topic by reiterating only most important points. (An explicit argumentation in favor of introduction of modeling technologies into teaching practice on a wide scale can be found in [12].)

Simulators can be effectively used as teaching aids that are able to significantly enhance the quality of knowledge acquisition by learners in comparison with conventional teaching means only if these tools are adequate and meet the requirements of didactic quality.

Models and simulators may bring the best effect in certain didactic contexts, when cognitive advantages for learners are more obvious. Such contexts should be of the "problem–based" learning type that may include
problems of the "what if?" type,
optimization problems,
assessment of heuristics for complex problems that deny analytic solutions,
case studies.

Simulators are intrinsic part of
computer–based training,
training stands,
virtual laboratories.

Modern simulation is combined with processes visualization, model animation and multimedia technologies. Visualization should as close as possible reflect the true nature of the phenomena or behavior of the real complex system under study. Visualized modeling results and especially animated models are advantageous as they facilitate deeper insight into the essence of the investigated phenomena or a problem and allow for profound understanding of the inner "nuts, bolts and springs" of a complex mechanism or a system that is studied via simulation.

Simulation with *AnyLogic* modeling tool

AnyLogic [http://www.anylogic.ru] allows discrete, continuous and hybrid modeling. It supports all types of models and all four modeling paradigms mentioned above in any combination in a single model, permits multi-agent simulation of complex distributed homogeneous/heterogeneous systems and concurrent processes. This new generation tool allows any level of abstraction and provides dynamically reconfigurable models. As a result it is not restricted to certain application fields.

Features and functionality of the tool include and are based on
a large library of ready-made system components (timers, statecharts, application-specific objects, etc),
flexible GUI for model developing,
hybrid statecharts (these allow to integrate discrete logic and behavior in continuous time),
UML-RT based structure diagram,
drag-and-drop composition of system elements,
visualization and animation features.
debugging and analysis features.

Usability of the tool is characterized by
emphasis on the model itself (not on modeling issues, techniques or details),
minimal or no programming (object oriented implementation, Java),
minimized labor and duration of model development cycle.

The user of *AnyLogic* does not need to learn a new block diagram or scripting language. Modeling is performed in the framework of UML and Java. "Drag-and-drop modeling" can be done with the help of field-oriented libraries (available for such domains as queuing systems, electrical circuitry, networking, mechanics, etc)

A variety of *Anylogic* models

The actual number of models designed and implemented in *AnyLogic* by hundreds of developers in research centers and universities in many countries can not be traced and is unknown. However, dozens of models developed by the team in St.-Petersburg are presented as samples by scripts available for downloading from http://www.anylogic.ru. A much abridged list given as an extraction can illustrate the broad scope of objects, processes, phenomena and systems of quite different nature that are presented by these sample models:

- Billiard Balls (Brown movement model)
- Heart beats model (continuous dynamic model)
- The Four Bars Assembly (kinematics problem)
- Constrained Pendulum (mechanics problem)
- Car suspension model (engineering mechanics problem)
- Tanker Fleet of a Company (logistics and transportation)
- Harvest Collection (operational model)
- Street traffic model (multi–agent stochastic model)
- "Firing Squad" (synchronization problem)
- Infected (Contagious) Ants (infection propagation by direct contacts)
- Population Development (demographic dynamics model)
- Urban Development (agent-based model)
- Mobile Communication Resources Allocation
- Enterprise Library Functioning
- Hospital Emergency Department
- Air Defense System Simulator
- Airport Terminal Simulator
- Tanker Unloading Terminal Simulator
- Subway in St.-Petersburg
- Blackout in NY State
- Electrical Power Grid
- Beverage Manufacturing
- Strategic Planning of Cellulose (pulp) Production
- Price fluctuation/relaxation [http://www.exponenta.ru/educat/class/courses/ode/theme17/theory.asp]
- Innovations and New Products Adoption

Examples of *Anylogic* models

Principles, methods and techniques of modeling and simulation in research fields and for educational goals are explicitly discussed in [9]. The monograph illustrates possibilities of *AnyLogic* by quite a number of models of various natures at all stages of their development and finally in action.

In this paper – to constrain within prescribed space limits – only four models will be described in brief. The selected problems are classified in accordance with the underlying modeling paradigm. (During the actual exposition of the report at the Conference it is planned to present another four such models marked by (+) in the list of eight models below. Scripts with animated graphic interfaces will be included in the multimedia presentation.)

1. Dynamic Systems Paradigm
- • Automatic Control System (Controller of a Boiler)
- + Bouncing Balls on Stairs (continuous/discreet dynamics model implemented by state-charts)

2. System Dynamics Paradigm
- • Influenza Epidemics Development
- + Weather Changes (“Determined Chaos” – a phenomenon known as the “Edward Lorenz Attractor” [13])

3. Event-driven Discrete Systems Paradigm
- • Transmission Control Protocol Module Simulator (transport layer Internet protocol of TCP/IP stack)
- + Distributed Process Termination (E. Dijkstra [14])

4. Agent-based Simulation
- • Heat Bugs Behavior (a simple migration model)
- + Alcohol Use Dynamics (stochastic model, Monte Carlo method)

Example 1
***Dynamic Systems Paradigm*: Automatic Control System (Controller of a Boiler)**

Dynamic Systems are complex objects that can be defined by sets of algebraic–differential equations. Such systems can also be depended on external events that can affect the environment and/or the structure of the system itself. Here we construct a model of a simple control system. The principle of operation is illustrated by Fig. 1.

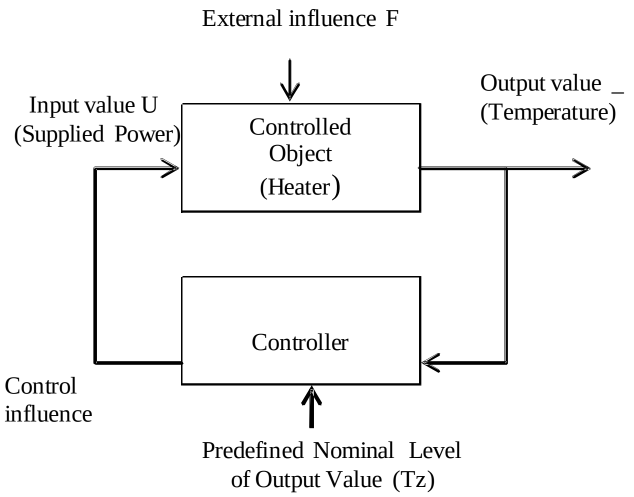

Figure1. Automatic controller principle

The boiler–controller system is defined by a set of equations:

$$d(T)/dt = 1/a *(k*U - F - T),$$
$$U=Ui +Kp*(Tz-T),$$
$$d(Ui)/dt = Ki*(Tz-T),$$

Would *Simulink* be used as a simulation platform than the block–diagram presented on Fig. 2 should be composed to solve the above set of equations. A one-to-one correspondence between formulae components and the diagram blocks is evident (very much similar to analog machine modeling procedures of the long past).

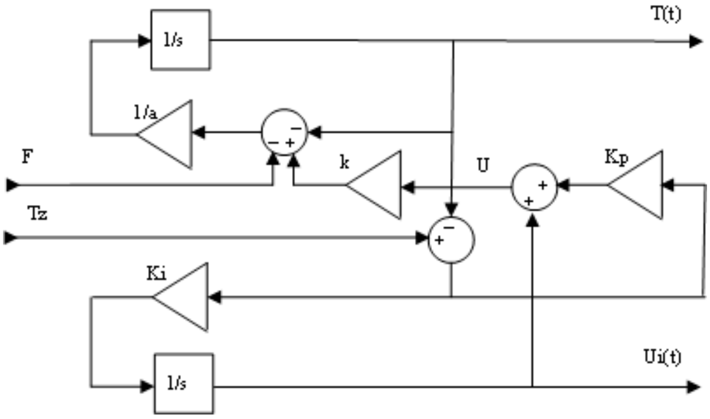

Figure 2. Block diagram for the *Simulink* model of boiler/controller

In *AnyLogic* we can construct similar model and, additionally, describe the external influence F as value that is dependent on certain discrete events – thus a hybrid model of a dynamic system is composed (Figure 3).

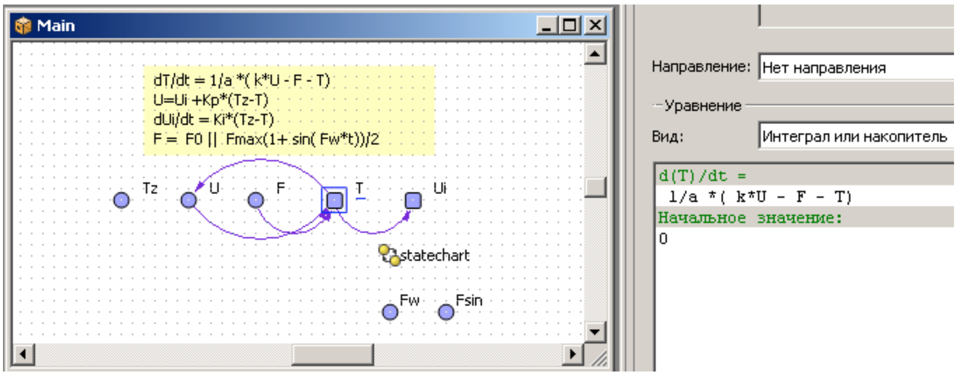

Figure3. A hybrid *Anylogic* model of a dynamic system

Our aim is to investigate with the help of this model the precision of the output value T control with dependence to configurable parameters of boiler and controller in given conditions (external influence). For convenience of manipulation with the model we design an animated GUI (Figure 4).

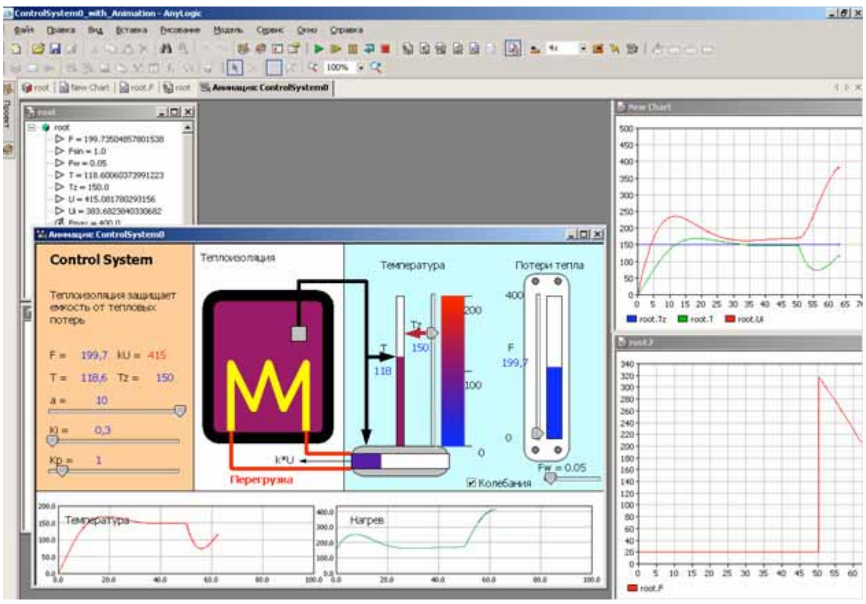

Figure4. Interaction with the hybrid Anylogic model via animated GUI

Example 2

***System Dynamics Paradigm*: Influenza Epidemics Development**

We use this example to demonstrate feasibility of graphic notation (stock-and-flow diagrams) characteristic for *System Dynamics* model definition – *Powersim* package Graphic Editor was used (Figure 5).

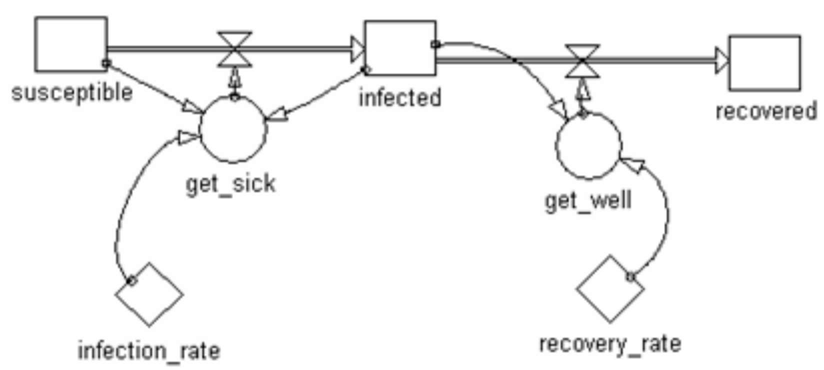

Figure5. A model of influenza epidemics – *Powersim* stock-and-flow diagram

The same model is drawn in AnyLogic as shown on Figure 6.

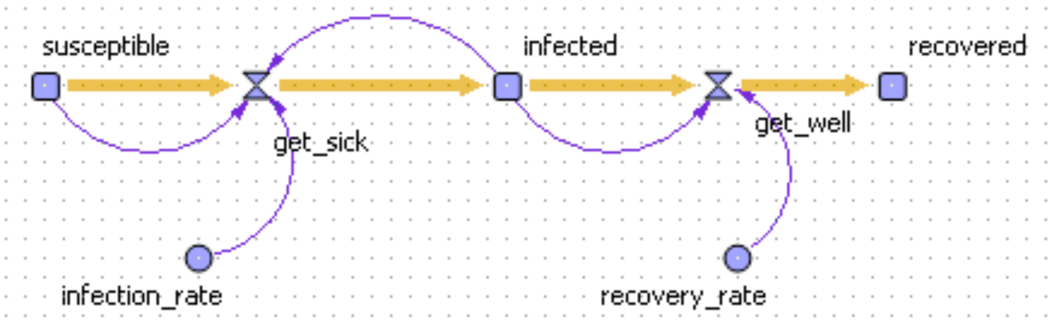

Figure 6. A model of influenza epidemics depicted as a diagram in *AnyLogic*

Experiments with the model and analysis are performed via animated GUI (Figure 7).

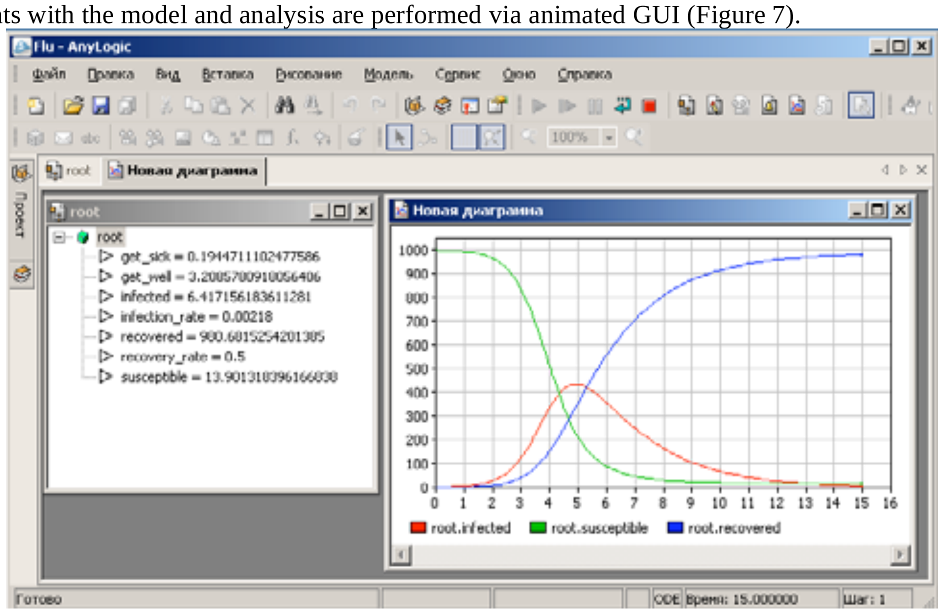

Figure 7. Results of the influenza epidemics model analysis in *AnyLogic*

Example 3

***Event-driven Discrete Systems Paradigm*: Simulation of Transport Layer Internet Protocol (TCP)**

Computer networks are, perhaps, the most complicated artificial distributed systems. Analysis of networks behavior under different internal and external conditions and optimization of communication protocols are of great importance. In the past this was achieved solely with the help of hardware network analyzers. Much later appeared software protocol analyzers – such as `tcpdump` which became a learning tool for a whole generation of networking specialists [10]. Recently, due to security constrains, experiments with real local and wide-area networks are hard to carry out. Thus, simulation becomes in most cases the only instrument that can be effectively used for research, educational goals and for training of specialists.

Quite a number of various dedicated modeling languages and simulators have been developed for the needs of telecom and networking industries. Among the most widely known software products are

Estelle (ISO) – Extended state transition Language,
SDL (ITU-T) – System Definition Language,
Lotus (ISO),
PDL – Protocol Definition Language,
NS2 – Network Simulator 2 (Laurence National Laboratory, Berkley).

Here we present a very simple simulator of the Transport Layer Internet Protocol (TCP) that is used in a laboratory class of the Distributed Computing & Computer Networking Chair (Engineering Cybernetics Faculty, Saint–Petersburg State Polytechnic University) as a TCP features demo tool.

Educational advantages are gained due to
experiments in virtual networking laboratory, rather than on the actual wide-area network,
visualization of the protocol behavior under different conditions,
adjusting protocol configuration (configurable options), timers, functions.

The following features of TCP are implemented in the model:
state transition diagram (automata for connection establishment and connection termination procedures),
sliding window protocol, acknowledging and retransmission mechanisms,
congestion avoidance algorithm, fast retransmit and fast recovery algorithms,
timers (RTO, 2MSL, keepalive, and persist timers),
TCP send/receive variable buffer sizes, configurable options.

The *Anylogic* model of the end-to-end interaction between TCP modules on both end-systems is very simple (Fig 8).

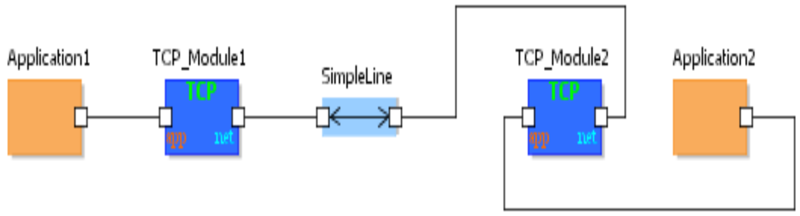

Figure 8. A simplified schematic of end-to end interoperation on the transport layer

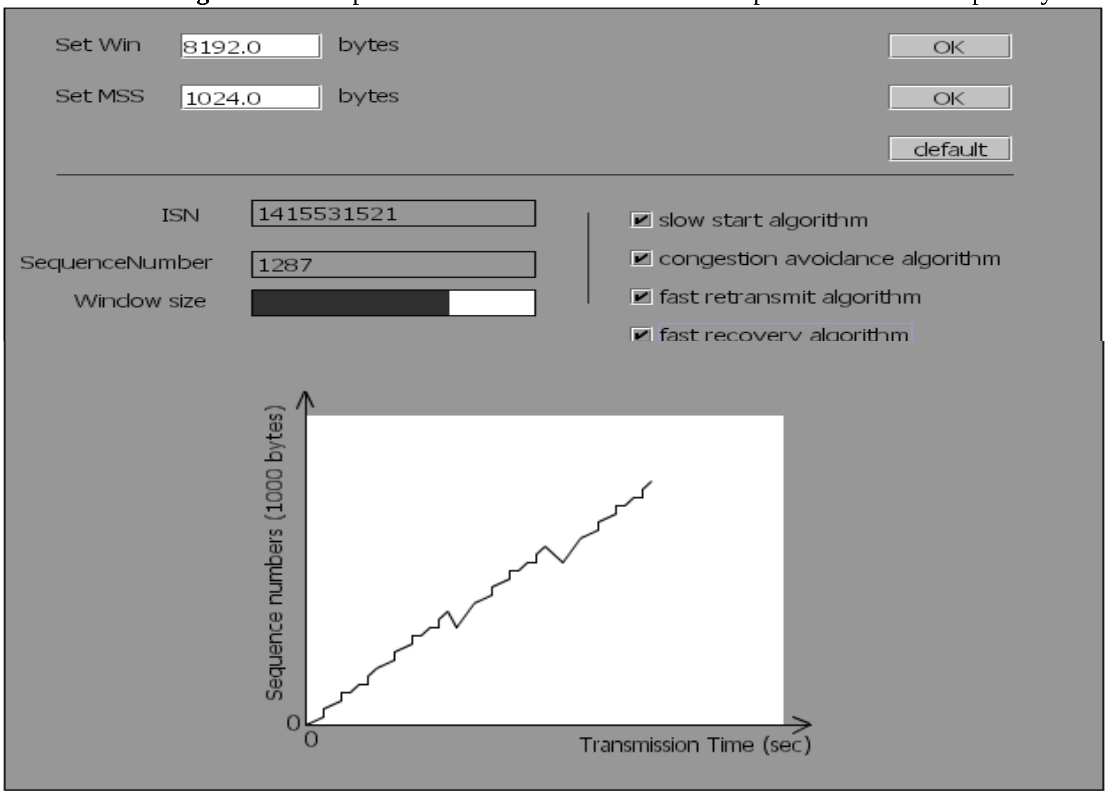

Figure 9. The graphic user interface to simple *Anylogic* model of TCP

Experiments with the model allow to simulate connection establishment and exchange of segments during transmission of a file under various conditions with different configurable parameters and options (Figure 9).

Modeling objectives and research possibilities with more complicated transport layer protocols simulators include
optimization of the protocol algorithms (e.g. congestion avoidance for TCP),
experimental assessment of newly proposed TCP options,
experimental evaluation of possible TCP extensions,
experimental comparison with alternate transport layer protocols under development.

A case study performed in a laboratory of the South Ural Technical University (the city of Chelyabinsk) presented in [11] reveal a comparison of developing a transport layer protocols Simulator that meets above requirements on two competitive platforms: Network Simulator 2 (Berkley) and AnyLogic 5. Reported results are in favor of the latter. [http://2005.edu-it.ru/docs/4/4-10.Dombrovskii.doc]

Example 4
***Agent-based Simulation Paradigm*: Heat Bugs Behavior (a simple migration model)**

This model has been initially implemented on the *Swarm* agent-based simulation platform.
Bugs can move on a surface that has cellular structure. Each bug emanates certain amount of heat. Besides, each bug has certain preferences for the temperature of the immediate surroundings and a sensor that enables to determine direction in which there is the nearest preferable spot. Certain mathematical laws describe heat propagation across the territory, cooling effect (loss of heat) for each cells and movements of bugs. The collective behavior of bugs is hard to predict. We can run a series of interesting experiments with this virtual community of bugs watching their collective and individual behavior as shown on Figure 10.

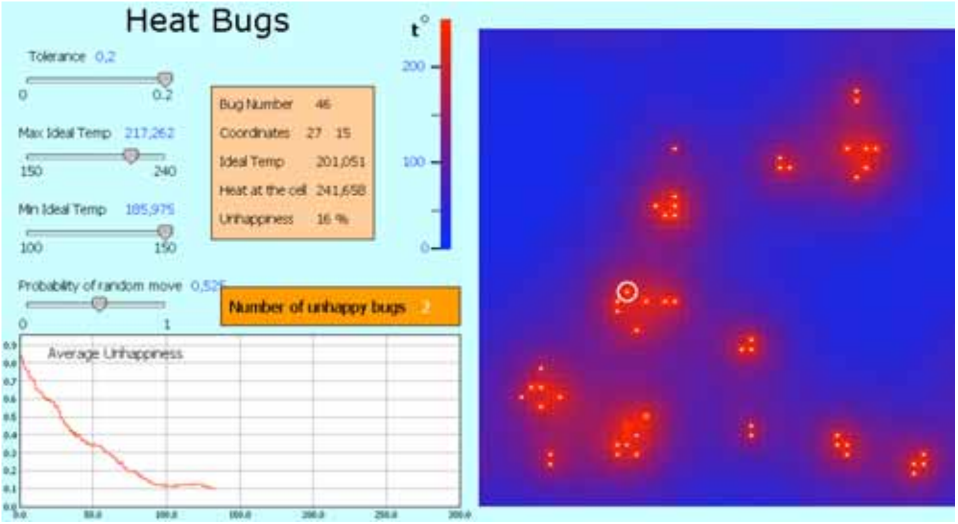

Figure 10. Community of heat bugs "under a microscope" (bug #46 is selected for investigation)

Conclusions

Evolution of simulation techniques for the past half century kept in step with the progress of available computing resources (processors' power, memory space), programming methods and tools (object oriented programming, Java, UML), computer graphics and multimedia support (GUI, 2D and 3D animation).

Recently, learning methods and supporting informational technologies in higher education can be significantly enhanced for a large variety of disciplines with the help of modern modeling and simulation methods.

Today we witness a qualitative breakthrough in simulation technologies – a new generation tools offer novel unique opportunities. A multi-purpose simulation tool is presented in the report as an example of such new opportunities. The tool has been already recognized and is used in many educational organizations both in Russia and in other countries, including several Universities in Japan.

References

1. Buckner G.D. *Simulink: A Graphical Tool for Dynamic System Simulation*. Technical Report Dept of Mechanical and Aerospace Engineering North Carolina State University, 2002.
2. Forrester J.W. *Urban Dynamics*. Productivity Press, 1969.
3. Sterman J.D. *Business Dynamics. System Thinking and Modeling for a Complex World*. Mc.Graw Hill, 2000. — 982 p.
4. Gordon G.A. *A General Purpose Systems Simulator*. IBM Systems Journal. 1962.

5. N. Schieritz, P.M.Milling. *Modeling the Forest or Modeling the Trees. A Comparison of System Dynamics and Agent-Based Simulation*, Mannheim University, 2000.
6. Banks J., ed. *Handbook of Simulation, Principles, Methodology, Advances, Applications, and Practice*, J.Wiley & Sons, Inc., 1998. — 849 p.
7. Sheldon M. Ross. *Simulation*. Academic Press, 3rd edition, 2002. — 274 p.
8. Hill D.R. *Object–Oriented Analysis and Simulation*, Addison Wesley, Inc., 1996.
9. Karpov Yu.G. *Imitacionnoye modelirovaniye system. Vvedeniye v modelirovanye na AnyLogic 5.* BHV Peterburg publisher, Saint-Petersburg, 2005 — 400 p. (*Published in Russian, this title is transliterated from Cyrillic*)
10. Stevens W.R. *TCP/IP Illustrated, Volume 1: The Protocols*, Addison-Wesley, 1994. [Available in Russian as translation: BHV Sankt-Peterburg Publishing, St.Petersburg, 2003. – p. 672 c.]
11. Dombrovskii K.A. *Simulation of Multiservice Communicational Networks*. The 1st International Conference "Modern Informational Technologies and Scientific/Engineering Education", Moscow State University, Moscow, 20 – 21 September, 2005, Proceedings, pp. 540 – 545 *(Published in Russian*)..
12. Mukhin O., Mukhin K. *Knowledge Representation by Interactive Formal models and Virtual Worlds for E-Learning. Technology and Tools for education by Means of Rapid Prototyping*. APRU Distance Learning & the Internet Conference, Vladivostok, October 4–7, Proceedings, pp 85–93, 2005
13. Lorenz E.N. *Deterministic nonperiodic flow*. Journal Atmosf. Sci., 20, pp.130–131, 1963.
14. Dijkstra E.W., Feijen W.H.J., van Gasteren A.. *Derivation of a termination detection algorithm for distributed computations.* Information Processing Letters, vol.16, pp.217-219, 1983.