

Tools development for physiological educational simulators

Jan Ruzs¹, Jiří Kofránek²

¹Faculty of Electrical Engineering, Czech Technical University in Prague

²Department of Pathophysiology, 1st Faculty of Medicine, Charles University in Prague
ruszj1@fel.cvut.cz, kofranek@gmail.com

Abstract — One of our main goals is to incorporate the usage of computers into the process of education. The aim of the work lies in the development of applications that would demonstrate non-trivial physiological systems behavior, their dynamics and regulation. In the process of the development of simulators, our main focus is creating the model and creating high quality animations to visualize the simulation results. In this paper, we describe our approach to usage of simulators, the development of these applications, their layered architecture and technologies we use. We use Matlab/Simulink for creating physiological models, .NET framework or Control Web as the main platforms and Adobe Flash for controllable animations. Simulation models are used for biomedical-engineering-students education, as well as a theoretical basis for a multimedia simulator for medicine students at Charles University in Prague and biomedical engineering students at Czech Technical University in Prague.

1 INTRODUCTION

The development of personal computers and internet in the end of last century brought new opportunities for a practical use of simulating models. Simulating model doesn't have to be just a tool for scientific research but also a very useful educational tool. Educational programs with simulating components aren't just a multimedia replacement of traditional textbooks. They are new educational tools, where old J. A. Komenský's credo "Schola Ludus" (School of game) finds its modern usage [1]. Connection of multimedia environment (sound and visual interface) with simulating models enables students to become familiar with problem in virtual reality and brings entirely new possibility for explanation of difficult problems. In simulating game, behavior of simulated object can be tested without any risks. You can try e.g. land with virtual airplane, cure virtual patient or test behavior of particular physiological subsystems. Simulating games can clearly explain complex relations in physiological regulating systems and causal chains in pathogenesis of all sorts of diseases.

2 PRINCIPLE OF DEVELOPING SIMULATORS

Development of effective educational program, which combines multimedia with simulation games [2-4] isn't easy and expects a working team of many

professions – experienced pedagogue creating basic scenario, creators of simulation models, medics, artists and programmers. This interdisciplinary collective production is being effected by using useful tools, which enable a component creation, connection of simulating programs and interactive multimedia according to given scenario to compact complex.

There are two types of problems which we have to consider during the creation of simulation model. First is the theoretical research itself, which basis in formalization of physiological relations expressed by a mathematical model. In our case, problem was solved within the branch of research grants and research purposes. Second problem is creating an own multimedia simulator, or more precisely creating an educational program which uses simulation games – is the practical application of the theoretical results which is related to the results of the research. The principle of creating the simulator is in the creation or verification of mathematical models.

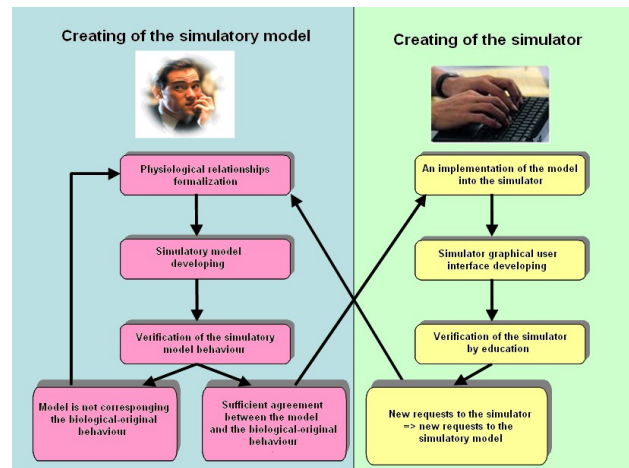


Figure 1: Two types of problems during educational models development

We are using more software technologies to build simulators, because we haven't found a single environment that would satisfy all our needs regarding effective visual expressiveness of physiological models, richness and maturity of the resulting simulators user interface and the ability to produce standalone and web-accessible applications. Generally we use either Microsoft .NET or Control Web as the base platforms we build upon. The models are created in Matlab Simulink and the animations created in Adobe Flash are used as a user interface [5]. We have spent significant amount of time on development of our own unique tools that allow the

Simulink model to be used in a .NET or Control Web application [6]. Flash animations are inserted into the form of ActiveX objects.

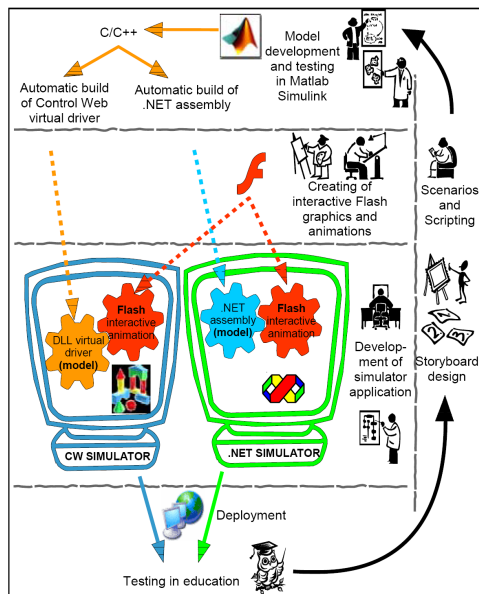


Figure 2: Development cycle of simulator application

2 MATLAB/SIMULINK

We use Simulink from *Mathworks* company as a modeling tool providing graphical design and simulation of mathematical models. Simulink allows us to gradually create models using single components connected to each other by a simulating net, which can remind us of electronic circuits. We can graphically express very complicated mathematical relations in Simulink. The created net is not only a well-arranged illustration; it is at the same time a functional simulating model. To each element we can add a virtual display or oscilloscope. After starting the simulation we can see the numerical or graphical form of the values flowing thru the connectors. Elements of a model can be grouped and hierarchically ordered, giving rise to subsystems with user defined inputs and outputs. We have adopted the concept of *simulation chips* that represent the building blocks of our simulation models. Another Matlab toolbox, Real-Time Workshop (RTW), exports the model as a C/C++ code. This code consists of a routine that executes the model logic, a structure that holds the model data and an ODE solver. We have developed two new RTW targets (we call them Wizards) that transform this code to be compatible with the .NET or Control Web. For the .NET the code is wrapped in a C++ .NET class that exposes the model's inputs and outputs as its interface. For Control Web the code is transformed into a C++ class that implements the interface making it possible to be used as a virtual driver in a CW application. Thus, the

final output of the Wizards is a Matlab independent dynamic library.

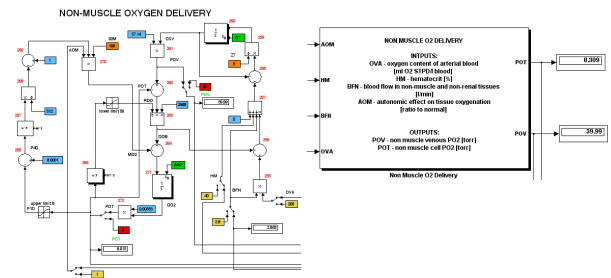


Figure 3: Example of "non-muscle oxygen delivery" subsystem created in Simulink as a graphic chart (left) and same subsystem as a simulation chips (right).

3 THE .NET PLATFORM

The .NET represents a modern and strong platform that allows rapid and comfortable application development with support for visual design of the GUI, creation of custom visual controls and the web deployment. The .NET platform is nowadays most supported for Windows, but open source projects such as Mono or Portable.NET promise the future of portable Simulator based on this platform. Verification relationships formalization

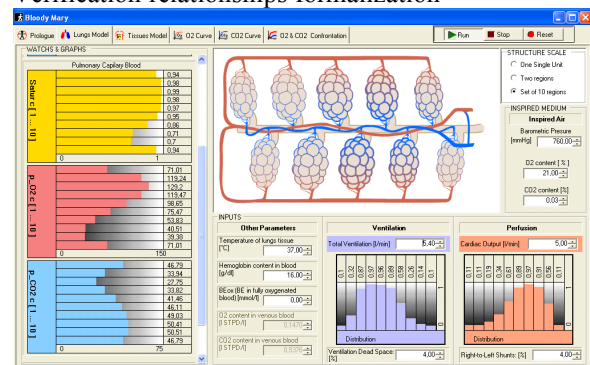


Figure 4: Example of educational simulator created in .NET platform

4 CONTROL WEB

Control Web of the Moravian instruments company is a platform mainly intended for development of industrial visualization and control applications on the WIN32 platform. It is optimized for realtime controlling and visualization. Its visual framework is component based and gives a great efficiency and rapid application development. We used the Control Web platform mainly for its advantages such as rapid prototyping and easy visual interconnection of model and visual controls.

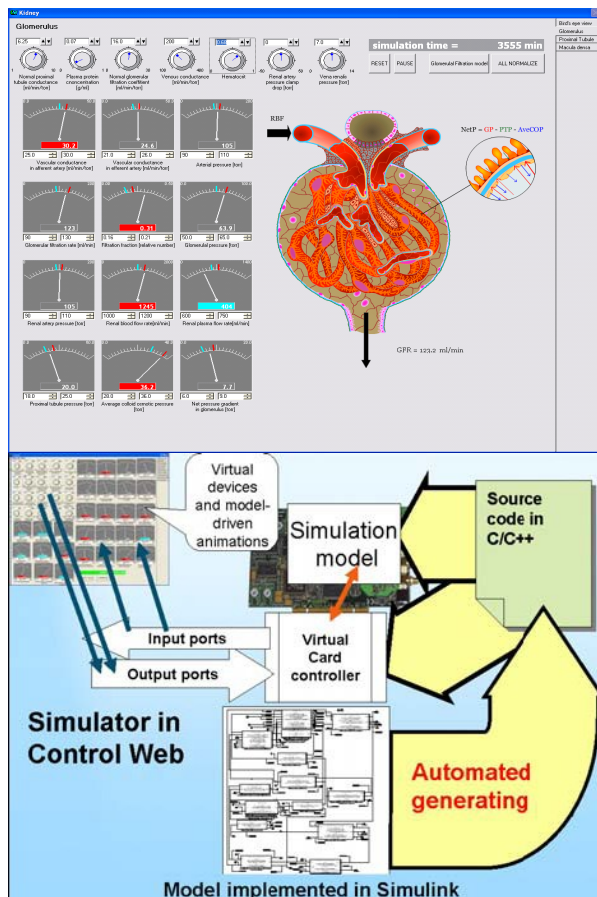


Figure 5: Example of a renal function simulator (up) in Control Web and virtual device generation (down) for Control Web

5 FLASH

Although it is possible to create the presentation layer of the application using standard or custom made .NET (or CW) visual components, for more sophisticated visualizations we use movie clips created in Adobe Flash. Fundamental feature of these animations is the possibility to control them programmatically (using Action Script) and thus affect the presentation. It is also possible to get the user input from a Flash clip and that way create animated user controls.

As the execution speed of the Flash and Action Script is increasing over the time, it is a good platform for implementation of simulators based on less complex models. Pure Flash based simulators are very portable and in-browser what is the most desirable deployment scenario for our simulation games. Some of our Flash based simulators can be seen in our Atlas of Physiology and Pathophysiology [6]. In our complex simulators based on the .NET or Control Web the Flash animations are embedded as ActiveX objects. Their behavior is than controlled by the underlying model layer.

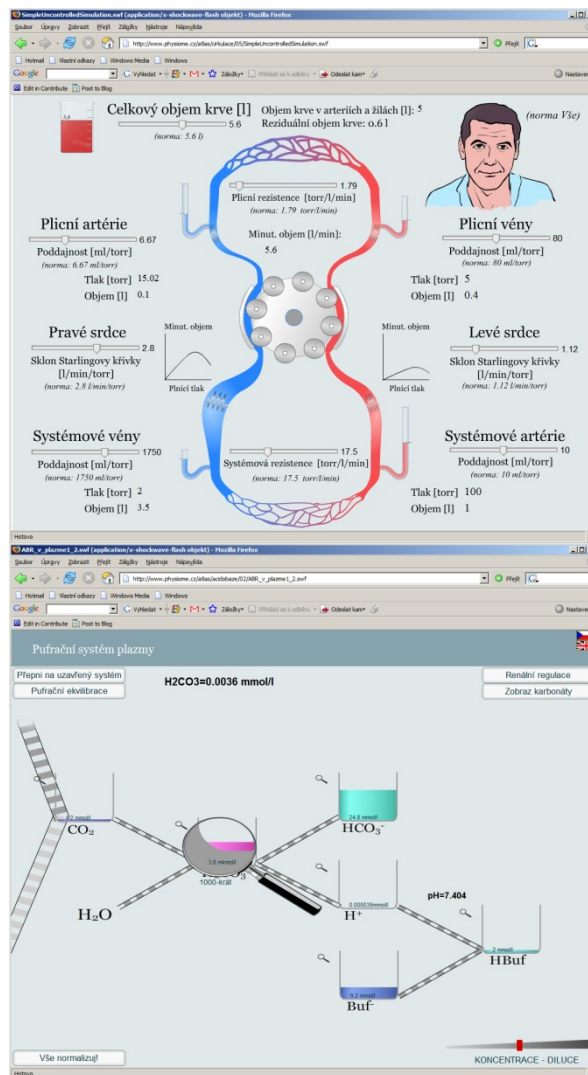


Figure 6: Demonstrating of right heart failure (up) and plasma buffer system (down) in Flash

6 LAYERS

In case of more complicated architecture the logic of connection between the flash animation and the simulation model can be relatively complex; therefore it is preferable to input an operation layer between layer with visual elements and the layer with simulation model. This operation layer takes care of the communication logic between user interface and the model. Between model layer and GUI layer is useful paste a control layer, where are directed all messages and events in the virtual GUI machines and where are simultaneously solved all communications with the model. The layer solves all context of imaging data and appropriate requirements on communication with the model by using finite state machines [7]. Logic is concentrated to one place, which essentially saves time through modification of GUI or model changes.

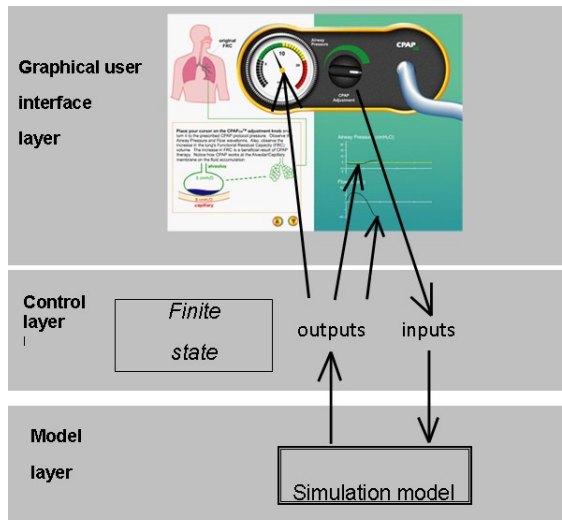


Figure 7: Architecture layers along simulators developing

7 CONCLUSION

We have shown our motivations for using simulators as simulation games in medical education. Our main improvement is usage of rich graphical user interface as a gaming element in simulators based on complex models. We use Matlab/Simulink for physiological modeling, Adobe Flash for controllable animations and the .NET or Control Web as the base environment. We have developed a unique way of exporting the Simulink model into the .NET framework and into the Control Web platform.

Acknowledgments

This work was supported by research MSM 0021620806 and by aid grant MŠMT 2C06031 and BAJT servis s.r.o company and by the research program “Transdisciplinary Research in Biomedical Engineering” No. MSM6840770012 and by the grant „Analysis and modelling biological and speech signals”, GAČR č. 102/08/H008.

References

- [1] Comenius Johann Amos (1592-1670): *The Great Didactic*. ed. by M. W. Keatinge (PDF files at Roehampton), available on: <http://onlinebooks.library.upenn.edu/webbin/book/lookupid?key=olbp34684>
- [2] Clark Aldrich's web page: http://clarkaldrich.blogspot.com/2007/03/immersive-learning-simulation_14.html
- [3] Kofránek J, Andrlík M, Kripner T, Stodulka P.: *From Art to Industry: Development of Biomedical Simulators*. The IPSI BgD Transactions on Advanced Research 1 #2[Special Issue on the Research with Elements of Multidisciplinary,

Interdisciplinary, and Transdisciplinary: The Best Paper Selection for 2005], 63-68. 2005. New York, Frankfurt, Tokyo, Belgrad. 2005. Journal (Full)

- [4] Kofránek J, Andrlík M, Kripner T.: *Multimedia educational simulators in pathophysiology – how to design and why to use them*. In: Gamal Attiya and Yskandar Hamam editors. Proceedings of the 5th EUROSIM Congress on Modeling and Simulation. Full Papers CD *Volume.*; Marne la Vallee, Paris, France: Eurosim - Francosim - Argesim; 2004. p. Simulation in Education 22-27.
- [5] Atlas of physiology and pathophysiology available on: <http://physiome.cz/atlas>
- [6] Stodulka, P., Privitzer, P., Kofránek, J., Tribula, M., Vacek, O. *Development of WEB accessible medical educational simulators*. In. Proceedings of the 6th EUROSIM Congress on Modeling and Simulation, Vol. 2. Full Papers (CD). (B. Zupanic, R. Karba, S. Blažič Eds.), University of Ljubljana, ISBN 978-3-901608-32-2, MO-3-P4-2, 1–6. 2007. Available on: [http://patf-biokyb.lf1.cuni.cz/wiki/_media/clanky/Harel, D.: Statecharts \(1987\): A visual formalism for complex systems. Science of Computer Programming, vol. 8, 1987, 231-274.](http://patf-biokyb.lf1.cuni.cz/wiki/_media/clanky/Harel,%20D.:%20Statecharts%20(1987):%20A%20visual%20formalism%20for%20complex%20systems.%20Science%20of%20Computer%20Programming,%20vol.%208,%201987,%20231-274.)