

ATLAS OF PHYSIOLOGY AND PATHOPHYSIOLOGY – INTERNET SIMULATION PLAYGROUND

**Jiří Kofránek, Stanislav Matoušek, Michal Andrlík, Petr Stodulka, Zdeněk Wunsch,
Pavol Privitzer, Josef Hlaváček, Ondřej Vacek**

Charles University in Prague, 1st Faculty of Medicine
The Laboratory of Biocybernetics and Computer Aided Teaching
U nemocnice 5, 128 53 Praha 2, Czech republic

kofranek@email.cz (Jiří Kofránek)

Abstract

The Atlas of physiology and pathophysiology (<http://www.physiome.cz>) is an interactive multimedia web-based educational application that facilitates the process of physiological system functions interpretation. It helps to comprehend the principles of system malfunctioning and to identify its reasons. The connection of a multimedia environment and a simulation models enables to experience the problem in a virtual reality. The educational simulator provides a way for trying out the behaviour of a simulated object without any risk. The atlas development workflow demands the cooperation between many professionals. The model designed by a mathematician is made to fit a scenario prepared by a physiologist. Then the model is implemented connecting the model with graphical components designed by a graphic designer. To suit various needs during the Atlas preparation, we have used various technologies and approaches, namely the .NET framework, the Control Web framework, Matlab/Simulink and Adobe Flash with ActionScript v2.0. The Atlas is available for medical students (in Czech language version) from March 2007 and the official evaluation of Atlas educational contribution begins from the autumn 2007. The Atlas of physiology and pathophysiology is a free accessible application. All educational text, interactive animation and simulation models including the source code are for free to all who are interested. Any form of cooperation with other European teams resulting e.g. in translation of our Atlas is welcome.

Keywords: E-learning, Modeling in physiology, Interactive on-line simulation

Presenting Author's biography

Ondřej Vacek. Postgraduate student of Jiří Kofránek at The Laboratory of Biocybernetics and Computer Aided Teaching which is part of the Institute of Pathophysiology of the 1st Faculty of Medicine, Charles University in Prague. Graduated as MSc. in biomedical engineering at Czech Technical University in Prague (graduated in 2007). Within the scope of his studies is: regulation in physiology, its modeling and utilization in education and the designing of extensive scalable models.



1 Education as a game

The Atlas of physiology and pathophysiology is an interactive multimedia educational application, that facilitates the process of physiological system functions interpretation. It helps to comprehend principles of system malfunctioning and to identify its reasons.

Educational applications are not only multimedia supplements of traditional textbooks. In addition they constitute a new educational utility that fulfills an old credo „Schola Ludus“ (education as a game), which was published in 17th century by famous european educationist Jan Ámos Komenský [2]. The connection between a multimedia application interface and a mathematical simulation model enables students to interact with currently explained physiological problems in virtual reality, thus bringing quite new opportunities when describing complex physiological interdependencies. A simulation game enables examination of simulated object without any real danger, e.g. landing with virtual aircraft, or in this case, virtual patient treatment and examination of functions of his separate physiological subsystems. Simulation also provides a way to explain complex relations among physiological regulation systems and pathogenesis of various disorders [1].

From the pedagogic point of view, the possibility to divide the examined object is highly appreciated. In this way, particular subsystems can be examined either separately or as a part of the whole system. Selected control loops can be temporarily disconnected to demonstrate subsystem reaction to input changes. Changing one input while other inputs are set to constant values (the „ceteris paribus“ principle) enables to examine subsystem's response to single value modification.

Subsequently it is possible to reconnect temporarily disconnected control loops again and examine their general effect when certain pathologies occur or the patient is subject to particular therapy. According to our experience, this approach leads to better understanding of complex dynamics of various malfunction pathogenesis and subsequently to selection of a better and more appropriate treatment.

The development of effective educational applications, combining multimedia components with simulation games, is an ambitious and complicated task. It demands the cooperation between many professionals – experienced teachers creating the basic scenario, simulation model makers, medical doctors, graphic designers and programmers. This multidisciplinary cooperative work is supported by appropriate tools that enable the connection of interactive multimedia with simulation models according to given scenarios.

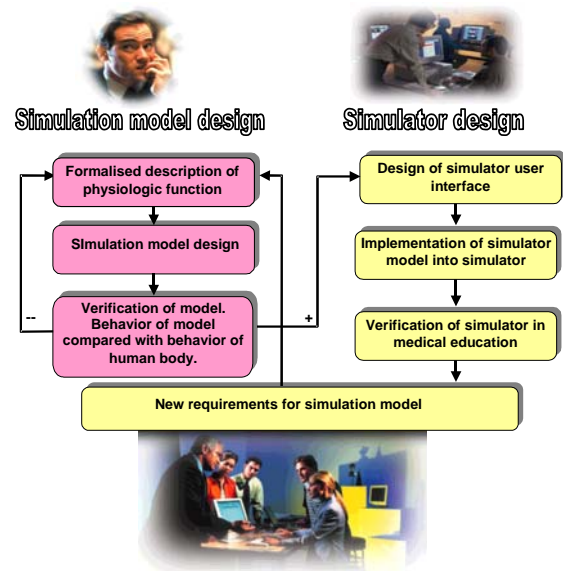


Fig. 1 Two kinds of problems were to be solved by construction of the medical simulator. Each of the problems is specific and requires using different developer's tools.

2 Simulation model – theoretical fundament of simulation games

Two types of problems are usually to resolve when creating simulation games (fig. 1):

Simulation model designing – a theoretical physiology based research work describing formalized physiological relations expressed in mathematical formulas. These issues were solved within the scope of research projects.

Designing of final multimedia simulator – designing of an educational application using simulation games is an application of theoretical results. The foundations of the simulator are verified mathematical models.

It has proved effective to use appropriate development tool for each step of this process. For simulation models design, we're using Simulink and Matlab from Mathworks. Using these tools enables us to compound the final model from separate components, so called software simulation chipsets.

Taking the advantage of the features of Simulink we have adopted the concept of *simulation chips* that represent the building blocks of our simulation models.

The simulation chips represent the subsystems of simulation model. Simulation chips resemble electronic integrated circuits, with information rather than current flowing between their pins (user defined inputs and outputs). The inner structure of these simulation chips can be hidden from the system architect; knowledge of physical quantities on the respective input and output pins is sufficient.

In biomedical models simulation chips correspond to certain physiological subsystems, they have defined

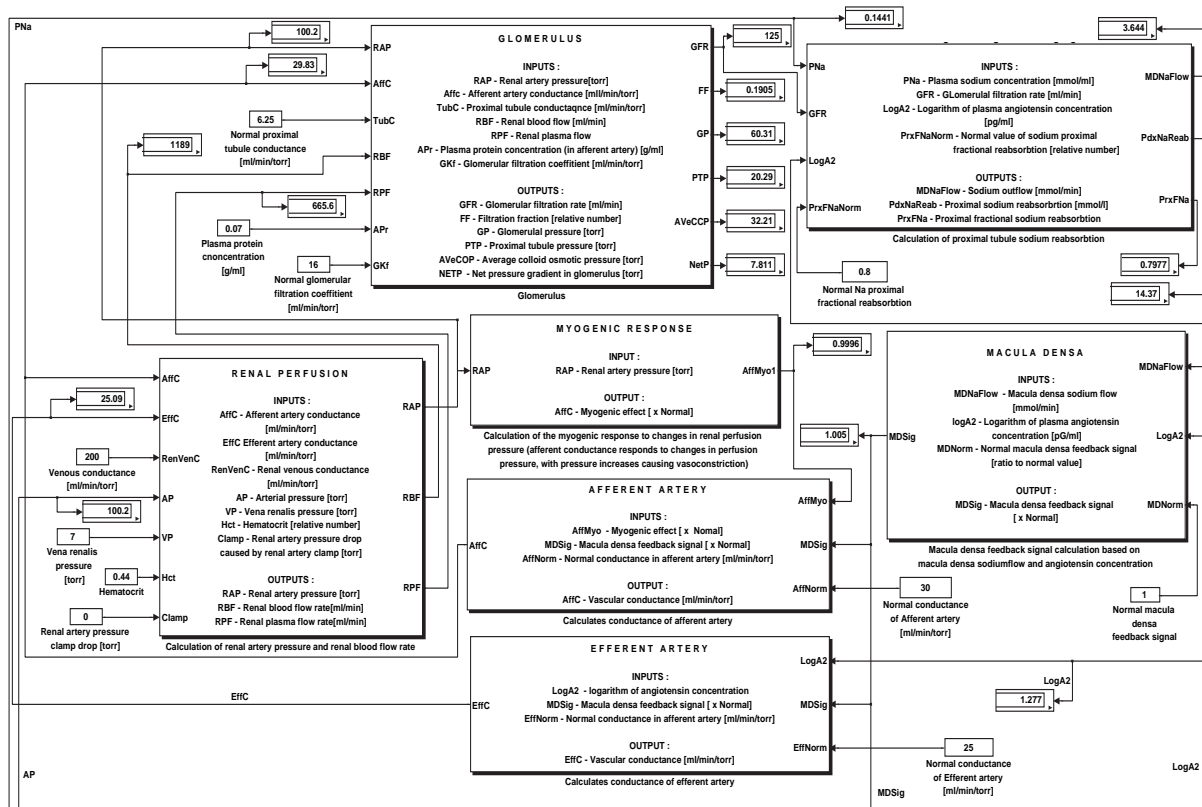


Fig. 2 Part of the of large-scale kidney model designed in Simulink using interconnected "simulation chips". This graphical representation of the large-scale model reveals the relationships among physiological subsystems represented as "simulation chips". Interconnected chips depict the core structure of the simulated physiological regulations. Schematic representations of the model with "simulation chips" are understandable for physiologists. A physiologist doesn't have to understand in detail the inner structure of the chip, but he knows what behaviour can be expected from the physiological subsystem represented by the network of chips.

inputs and outputs through which they communicate with other simulation chips (fig. 2).

The inside of a chip contains all mathematical expressions needed to implement its physiological function, but this is hidden from those team-members that do not need it; still they can analyze and understand the structure and the behaviour of the simulation model.

Simulation chips can be organized in Simulink libraries so that they can be easily reused among models, moreover they represent themselves a means of documentation thanks to their graphical nature, as well as they can become part of the Matlab help system.

Developing simulation models in biomedical sciences usually results in *team-work*. On one side, there is a *system analyst* – an expert in the formalization and creation of simulation models, a theoretical physiologist who describes a physiological system formally and tests the simulation model. On the other side, there is a classical *experimental physiologist* or *clinicians* who cannot deal with the integro-differential description of chips *depict the core structure of the large-scale* physiological system, but

certainly can evaluate the functionality of a simulation model.

Usage of simulation chips throughout the simulation model design can, in our experience, facilitate the understanding between these two groups of people. Physiologists can look at this chip as like a black box with some physiological functions and programmers can look at the chip as like a subprogram or algorithm that is part of a more complex system.

An experimental physiologist doesn't have to understand in detail the inner structure of a simulation chip. But he knows what behaviour can be expected from the physiological subsystem represented by the chip or a network of chips. The graphical representation of the model reveals the relationships among the chips. Importantly, schematic representations with simulation chips are understandable for physiologists.

The development project results led to the whole library of physiological models for Matlab/Simulink as theoretic basis for subsequent development of educational simulation models [5,6]. The library is a periodically updated open-source product available at our internet site (www.physiome.cz/simchips).

3 Model design for web-based simulation games

Since Mathworks tools are primarily intended for engineering specialists, they're not suitable for a common user, who wants only to „play a little“.

Although it is possible to create a rather friendly user interface to control the created model in Simulink environment, this interface is too complicated for purposes of use of the simulation model in medical education and moreover, it requires further purchase of licenses. Besides, one pays with higher hardware requirements for a comfortable environment, which is assigned especially for creation (not for operation) of

simulation models. Less powerful computers run the simulation extremely slow.

Therefore it was necessary to separately create another simulator along with its user interface, based on a debugged and verified structure of a simulation model. The implemented simulation model can then run faster and the possibilities of user control are considerably more comfortable for non-technicians (i.e. medical students and doctors) than those in Matlab or Simulink environment.

For simulator creation, we use the **Microsoft .NET** platform. So that it is not necessary to create the

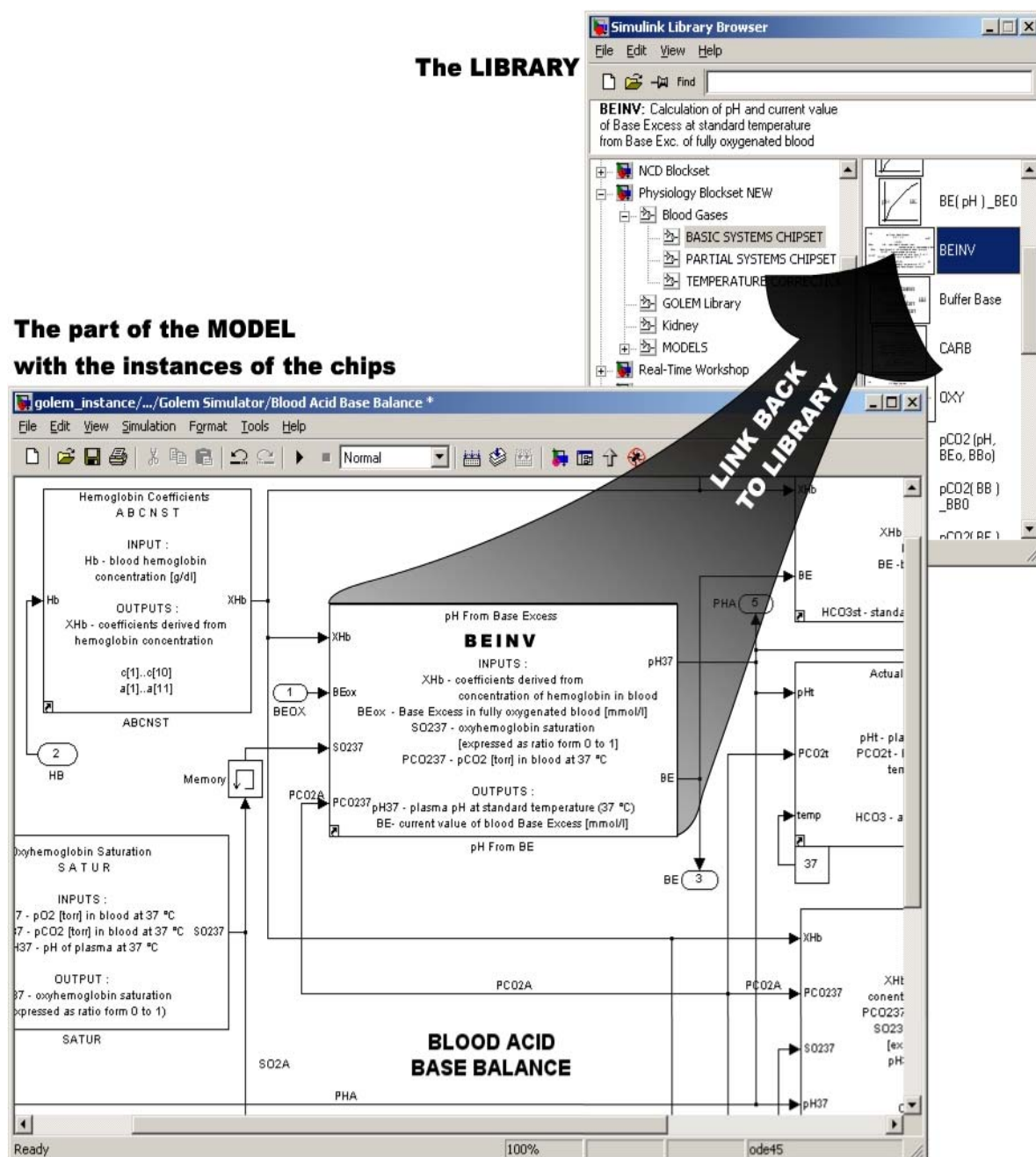


Fig. 3 "Simulation chips" are stored in libraries (top window). A model (bottom window) contains instances of the chips. Note the 'quick reference' of the function, inputs and outputs of the simulation chip in its mask. The library of simulation chips is open-source product available at our internet site (www.physiome.cz/simchips).

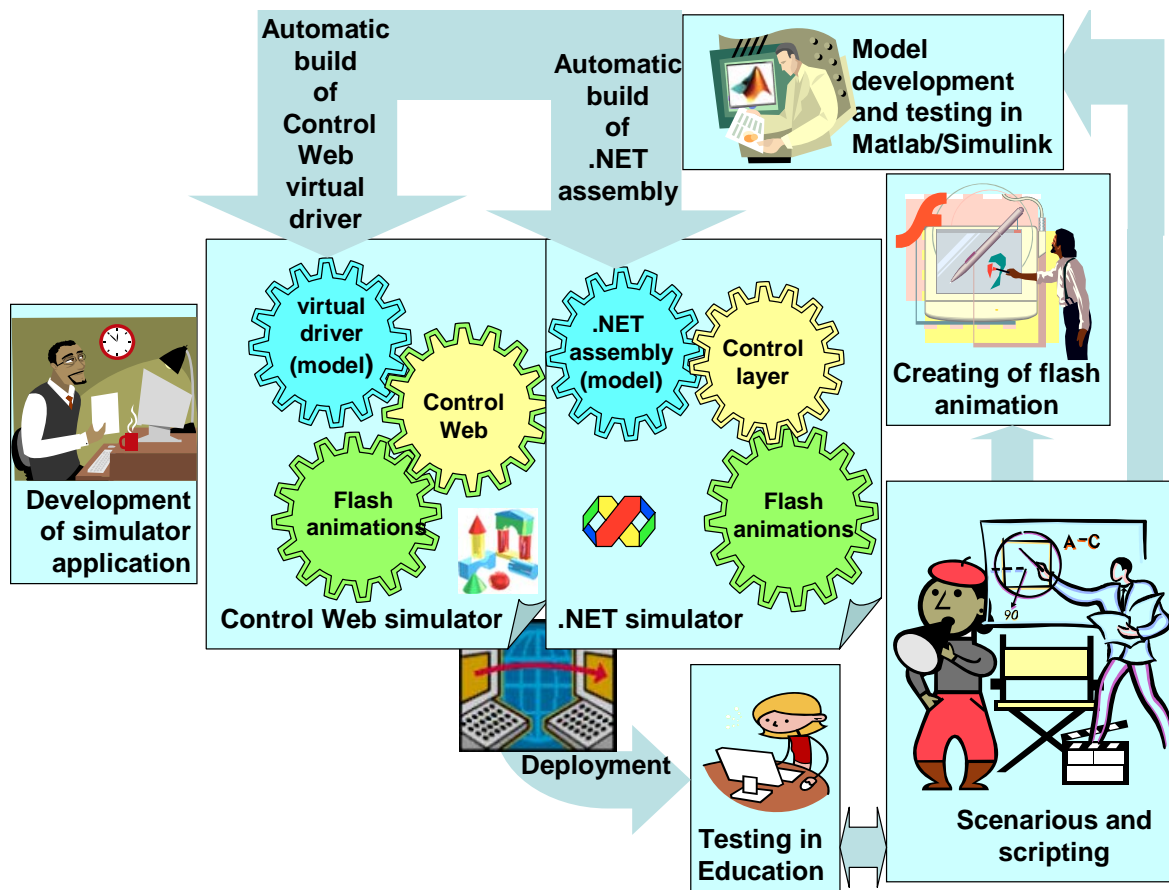


Fig. 4 Development cycle of educational simulator application

Simulink model in Visual Studio .NET manually once again, we have created a special software tool which automatically converts the Simulink model into the target platform - this tool automatically generates a simulation model in .NET assembly from Simulink (see fig. 4).

Another platform in which the simulation games can be run is the environment **Control-Web**, developed by the Moravian instruments Inc. The Control-Web is a development system originally meant for being able to easily create a visual design of controlling and measuring devices for industry with use of a PC, equipped with a special measuring/controlling card. We use this application as an environment for running the simulation models, where the generated controller of the virtual control card in the computer is by the Control Web system perceived as an interconnection with the industrial interface – but in fact, it contains a simulation model. The Control Web is somehow „fooled“ this way – if the system is sending some control signals into the industrial device, it is in fact sending input values into the model (fig. 5). If the application created in the Control-Web environment reads any measured signals from the industrial device, it is in fact reading the output signals from the model.

This way we manage to use the total visualization potential of this device for creation of multimedia presentations of simulation models for educational

purposes. This way, we have created for instance educational applications „Internet kidney“ or „Mechanical properties of a skeleton muscle“ in our Atlas.

4 Interactive animations – puppets controlled by simulation models

For creation of user interfaces for an educational simulator, it is very useful to visualize the simulator as an animation. That is why we interconnect the simulation model with a multimedia animation created with the help of Macromedia Flash. The animations can be then controlled by outputs from the implemented simulation model and the meaning of numerical values can be represented graphically – i.e. a schematic picture of a vein can dilate or compress, an air sack can breathe more or less deeply, a device pointer can move and continuously display values of an output variable of the model, which is being read from the running simulation model in the background.

Conversely, various inputs can be entered into the simulation model with the help of visual components (various kinds of buttons, knobs, draw bars etc.).

In case of more complex architecture, the interconnection logic between the flash animation and the simulation model is rather complicated, and therefore it is more advisable to insert a control layer

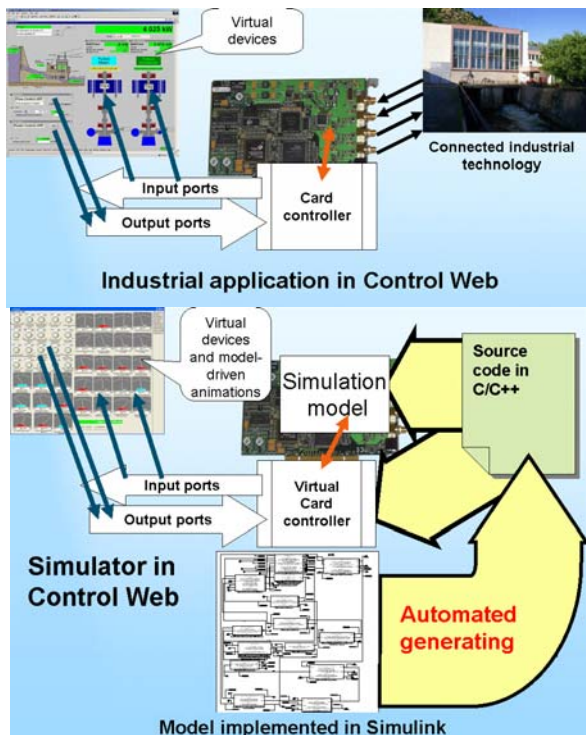


Fig. 5 The Control Web application communicates with the mathematical model through a virtual driver. In industry the Control Web application would communicate with the measuring/controlling card through a similar driver and this card would be interconnected with any industrial technology. Next, the FLASH interactive animations can be built into the user interface of the Control Web application.

between the layer of visual components and the layer of the simulation model. This new layer in one single point controls all the communication logics between the user interface and the model, where also the respective content is being stored.

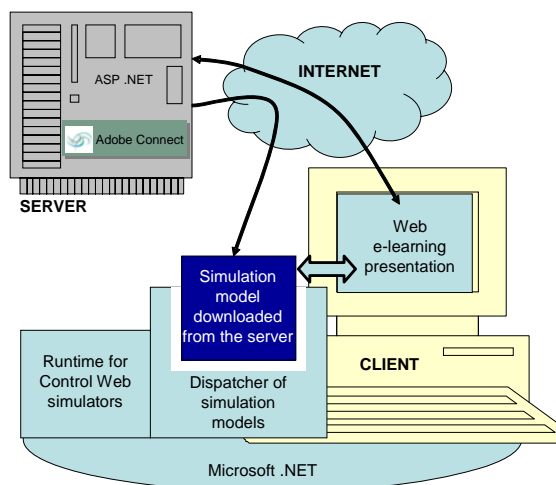


Fig. 7 Deployment of Control Web and .NET simulators

This organization is necessary in case of more complex models and simulators whose user interface is represented by a number of virtual devices on more

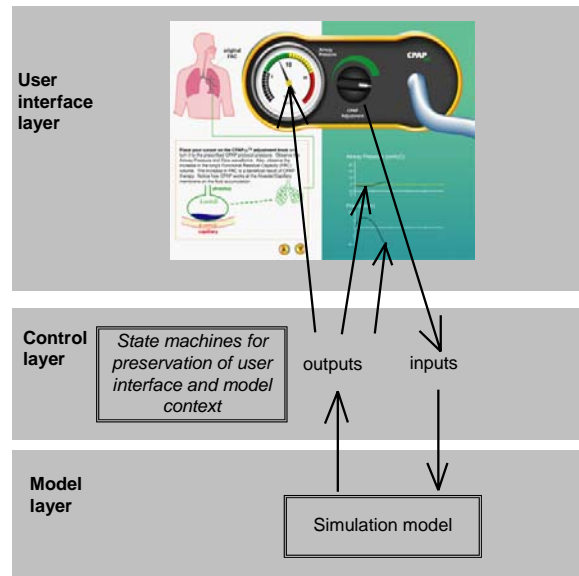


Fig. 6 Three layers architecture of simulators

mutually connected screens. The advantages of such an organization excel especially at modifications both of the model and the user interface. In technical publications, a so-called UCM architecture (User interface – Control layer – Model layer) of simulators is being discussed [3,4].

For this purposes, the state machines capable of remembering the respective context have proved well. The Matlab/Simulink environment offers execution of state machines through a special toolbox (Stateflow) which is immediately interconnectable with a simulation model in Simulink (fig. 6).

It is also possible to execute the state machine on the side of the user interface – directly in Flash. This machine then directly communicates with visual objects in Flash.



Fig. 8 Main screen of Atlas of physiology and pathophysiology. This atlas contain explanation multimedia e-learning components combined with simulation games

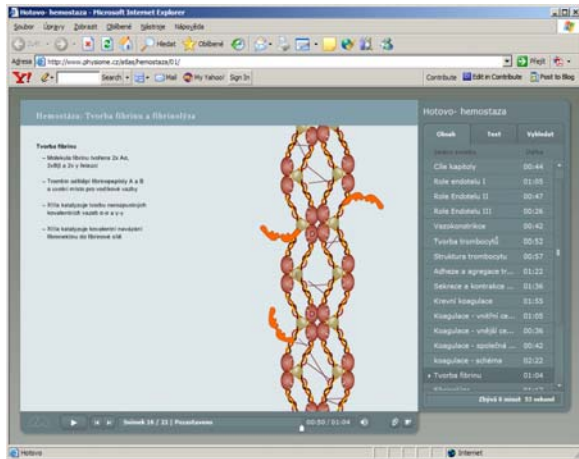


Fig. 9 The usage of multimedia components during the explanation of coagulation failures

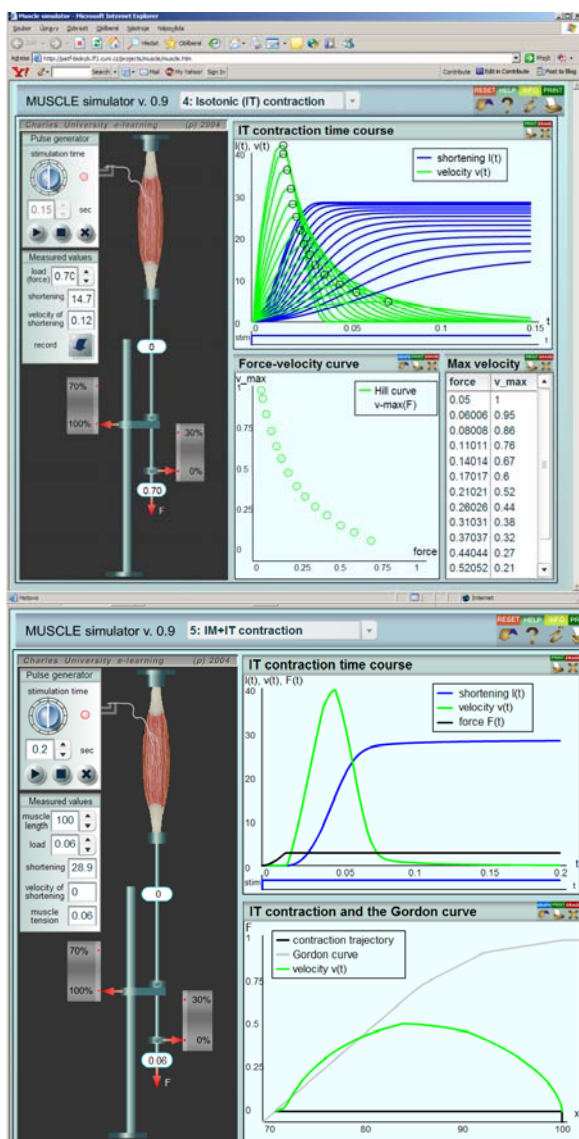


Fig. 10 Muscle simulator implemented in Adobe Flash using Action Script language in educational program on macroscopic and molecular aspects of muscular contractility. The time-course of isotonic muscle contraction (shortening and its velocity) is generated and Hill's curve reconstructed from the results.

5 Simulation games on the Internet

Another task to solve was to find a way to embody simulators into the Internet e-learning applications. There are several ways to do that. One is to run model on a server and send only the outputs to the client's workstation. But that means one model instance started for each connected client, apart from slow time response. That's why we're using client-based models.

Simple models may use Flash ActionScript, but complicated models must be provided with more sophisticated simulator background.

Our solution is depicted in fig. 7. First step is the .NET framework installation, in case user doesn't already have it. Then the special client application „the simulation model dispatcher“ is loaded. The dispatcher's component is the Control Web runtime that ensures the simulator execution. Internet e-learning application provides files necessary for the dispatcher to execute the simulation. The dispatcher acts similarly to a picture viewer, but instead of viewing a picture it runs a simulation.

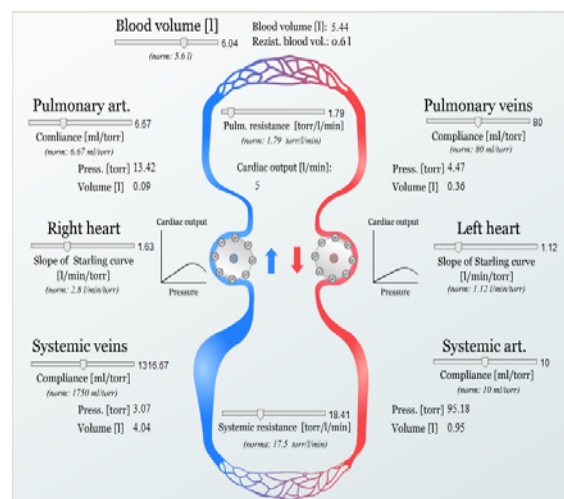


Fig. 11 Right heart failure demonstrated using simple model of circulation implemented in Adobe Flash using ActionScript language

6 Multimedia e-learning using simulation games

Simulation games are a part of e-learning multimedia educational lessons, whose foundation is a scenario created by an experienced pedagogue. The scenario created by the pedagogue consists of an explanatory text and supporting figures and animations. The animations are being created in the Adobe Flash environment by a pedagogue closely cooperating with an artist. Afterwards, the text is narrated and synchronized with the start of particular animations and with the links to the simulation games. All these components are put together in educational lessons [7].

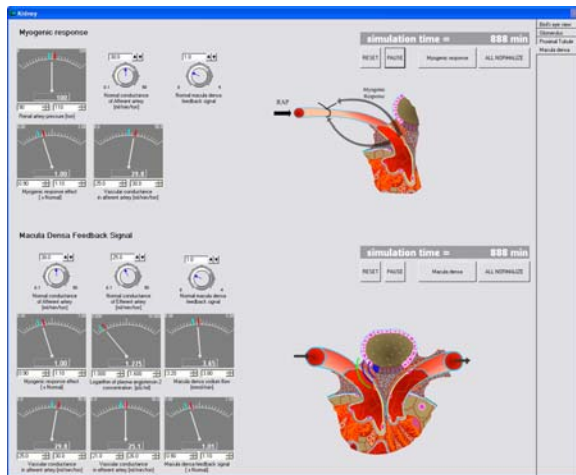


Fig. 12 An example of a simulator user interface combining Control Web controls (graphs and knob) with Flash animations (vas afferent and efferent arteriole in kidney glomerulus) in educational program on kidney functions.

For creation and completion of multimedia educational lessons a software environment of Adobe Connect (formerly Macromedia Breeze) server is being used.

7 From enthusiasm to technology and interdisciplinary cooperation

Though the application of computers in education has become the main theme of many conferences and papers and the potentials of hardware and software tools have enabled the design of demanding interactive multimedia, no extensive expansion of medical e-learning application has happened.

The reasons are following:

- First, e-learning application development is fundamentally more expensive as to human and material costs.
- Second, high-quality e-learning application development requires team cooperation of experienced teachers, doctors, mathematicians, physics, programmers and design artists.
- Finally, appropriate developing tools have to be available to allow the interconnection of aforementioned professionals. Also, to master the design tools requires certain time and effort.

If the simulation model enabling interactive simulation games operates in the application background, computational demands are raised rapidly. Developing team has to involve professionals capable of designing such models (doctors, mathematicians, physicists and computer programmers).

We consider the creation of our interdisciplinary team to be one of the greatest achievements, since it

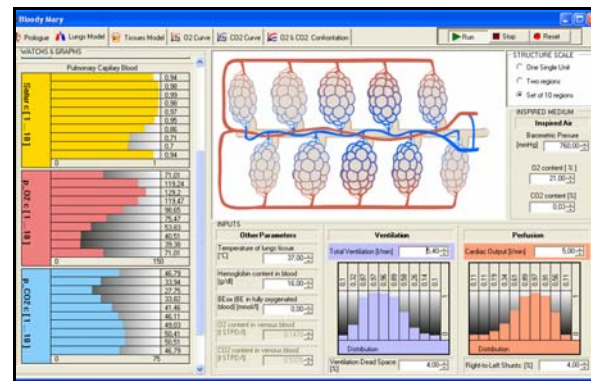


Fig. 13 An example of a simulator user interface combining custom MS .NET controls (bar graphs) with Flash animations (alveoli) in educational program on ventilation-to-perfusion abnormalities.

involves medical doctors, mathematicians, computer programmers and design artists.

The Atlas of physiology and pathophysiology is a free application. All educational text, interactive animation and simulation models including the source code are for free to all who are interested. All supportive applications developed in our Laboratory, enabling automatic conversion of Simulink models are free as well.

Any form of cooperation with other European teams resulting e.g. in translation of our Atlas is welcome.

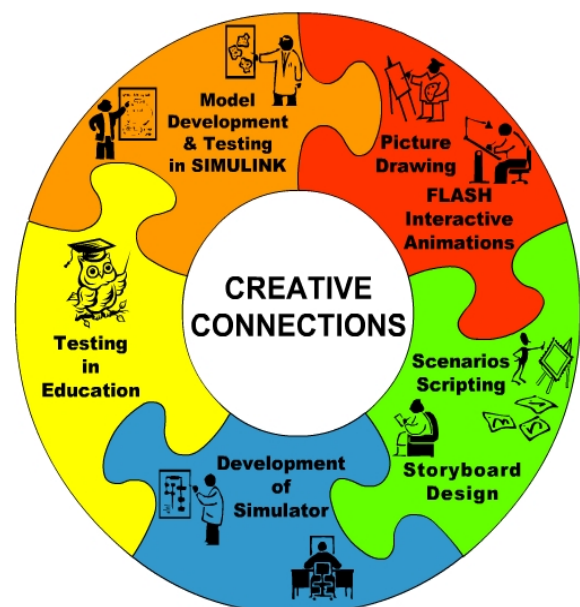


Fig. 14 Various professions and activities involved in the educational software development

8 References

- [1] Coleman, T.G. and J.E. Randall: HUMAN. A Comprehensive Physiological Model. *The Physiologist* **26**, pp. 15-21. 1982
- [2] 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/lookup?key=olbp34684>
- [3] Harel, D.: Statecharts: a visual formalism for complex systems. *Science of Computer Programming*, **8**, 231-274. 1987
- [4] Kayne, J., Castillo. D.: *Flash MX for interactive simulation*. Thomson Delmar Learning, 2003, ISBN 1-4018-1291-0.
- [5] Kofránek, J., Anh Vu, L. D., Snášelová, H., Kerekeš, R. Velan, T. (2001): "GOLEM – Multimedia simulator for medical education." In: *Studies in Health Technology and Informatics.*, vol. 84. MEDINFO 2001, Proceedings of the 10th World Congress on Medical Informatics. (V.L. Patel, R. Rogers, R. Haux (Eds.)), IOS Press,
- [6] Kofránek J, Andrlík, M., Kripner, T. Mašek. J.: From Simulation chips to biomedical simulator. In Amborski K, Meuth H, (eds.): *Modelling and Simulation 2002, Proc. of 16th European Simulation Multiconference*, Germany, Darmstadt, 3.-5. 6. 2002. ISBN 90-77039-07-4, pp. 431-435
- [7] 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, 62-67. 2005. New York, Frankfurt, Tokyo, Belgrad. 2005.

Acknowledgement

Medical simulators project is supported by aid grant MŠMT 2C06031 and BAJT servis s.r.o company.