# Web-based Educational Simulators for Teaching Pathological Physiology

Petr STODULKA, Pavol PRIVITZER, Jiří KOFRÁNEK
Laboratory of Biocybernetics and Computer Aided Teaching, Institute of Pathological Physiology, First Faculty of Medicine, Charles University in Prague
e-mail: petr.stodulka@lf1.cuni.cz

This extended abstract shortly introduces the problem of designing educational simulators of physiological processes, used technologies and their development process.

The educational experience in pathological physiology at the First Faculty of Medicine, Charles University in Prague shows that physiology and pathological physiology have rather close to physics, biochemistry and other technical disciplines that medical students are not used to. Deep understanding of physiological systems and regulations is very hard to achieve only by studying textbooks without convenient learning tools. Hence, our goal is to incorporate computers into the process of education. A big part of our e-learning activities resides in the development of applications that demonstrate the behavior of non-trivial physiological systems, their dynamics and regulation in schematic, easily understandable, yet still precise manner. It is important to keep the form familiar for the students (similar to textbooks illustrations) and to allow them to interactively affect the simulation, experiment with it and explain the complexity by experiencing the issue in a virtual reality. Simulations allow demonstrating complex dependencies in physiological regulation systems and causal chains in pathogenesis of various diseases.

The development of educational physiological simulators is an interdisciplinary work that requires a cooperation of experts from many domains (see [1]). Teachers have to identify the problem domain and work out a scenario for the simulator. Modelers in cooperation with physiologists and clinicians develop and verify a model. Artists create graphics and animations and programmers put it altogether. As the models are developed on specialized platforms (special modeling languages and environments) and the animations are created in special animation programs, one of the crucial tasks is to ensure that these components will work together.

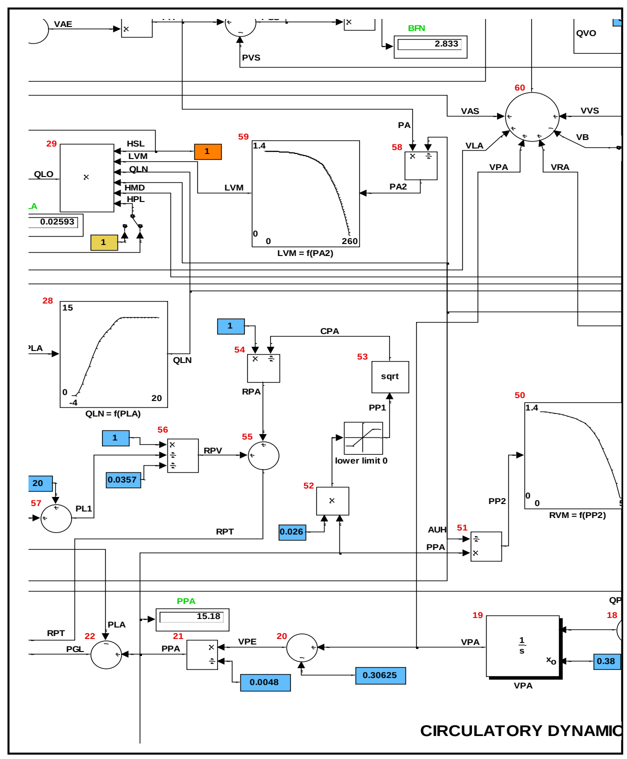


**Fig. 1** Illustrative cut out from the model of blood circulation implemented in Matlab Simulink.

We build our simulators on formalized mathematical models of physiological systems that are developed in specialized modeling languages and environments like Matlab Simulink or Modelica (see Fig. 1). The result is a system of (mostly) differential equations together with support logic of the model. Since the models are usually rather large and complex, transferring them into the program source code manually would be too difficult or even impossible. Not to say that such code would be error prone and hardly maintainable. Therefore we have developed tools for automated processing of models into a form easily usable in general programming environment (in our case the Microsoft .NET Framework). The Simulink model can be exported in the form of the C/C++ code using another Matlab tool RealTime Workshop. This code then gets postprocessed: it is modified and wrapped to be in C++ .NET compliant form. That is compiled to a .NET Assembly which contains the model as a class that exposes the inputs and outputs of the model as its properties and offers methods that control operations of the model. The result is completely independent of Matlab and does not use any of its libraries. It has to be emphasized that the result is a wrapper around native C/C++ code and therefore is considered to be non-managed code. As for Modelica models, we are currently working on a compiler that would compile Modelica language into a .NET Common Intermediate Language. We change the back-end of the OpenModelica Compiler (open-source implementation of the Modelica language) to produce CIL instead of a C code. Together with rewriting the simulation runtime code into a (managed) .NET code and with our implementation of ODE solver in .NET we will get fully managed .NET models created in Modelica. To be able to use models created by third parties, we have created a rough converter of the J-Sim language to Modelica as well as CellML to Modelica converter. CellML is used at the Auckland Bioengineering Institute (University of Auckland) and J-Sim is used at the Department of Bioengineering, University of Washington for the Physiome project.

The educational aspect remains in the user experience: the visualization of the simulator is just as important as the model itself, if not even more. We combine common user controls as buttons or sliders with animations that not only display the simulated processes, but also allow interactions. The animations are driven by the model and can express both quantitative information (heart rate, blood flow etc.) and qualitative information (overall condition of the patient by the expression of her face). The interactive animations are created using Adobe Flash or Microsoft WPF. Flash animations are inserted into the .NET world in the form of ActiveX components (see Fig. 2). The animations display properly only if Flash player is installed on the computer and the communication between the application and movie clips is realized through the ActiveX interface by sending FSCommands and FSCommandEvents that are not strongly typed and type safe. On the contrary, the WPF technology gives much the same display possibilities as Flash but is itself a part of the .NET Framework and as such eliminates the technology gap between the application logic and animations. This allows a better control of animations, better response to user events and a more straightforward and safe data flow. Moreover, WPF offers rich typography and text processing capabilities as automatic formatting to columns, paging, zooming and text searching. The text may contain mathematical equations (which are hard to achieve in HTML), graphics, animations or any multimedia and even simulators themselves.

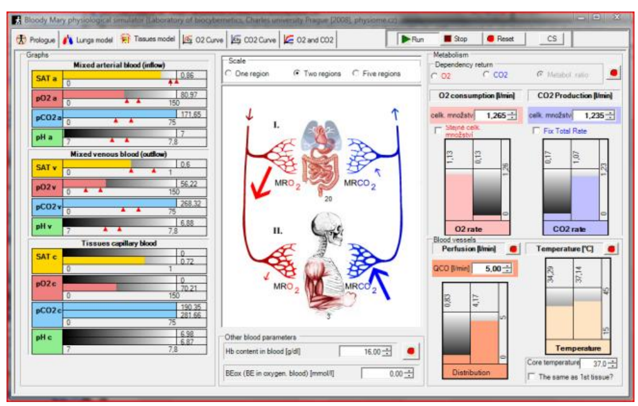

**Fig. 2** The Bloody Mary simulator of blood gases exchange. An example of complex simulator user interface combining graphic user controls with Flash animations.

Under certain conditions the applications created on the .NET 3.5 Framework may be run directly from the internet in the browser window; these are so called XBAP applications (see Fig. 3). The biggest restriction on XBAP applications is that they must not use any non-managed code. This excludes Simulink from creating XBAP simulators. The XBAP application must use only WPF user interface and they run in a security sandbox without installing. They are executed straight from the internet offering the user an impression of browsing web pages (XBAP applications are even able to use browsers 'Back' and 'Forward' capabilities, they may contain link to other web pages etc).

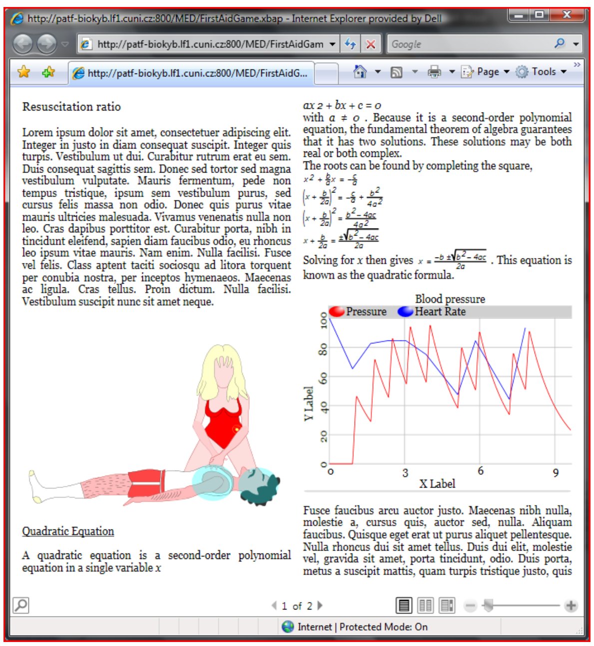


**Fig. 3** Example of XBAP in-browser application. Text divided into columns containing equations and an embedded simulator. The simulator in the left column allows the user to click on the head or chest of the lying person to play the resuscitation game. The graph displays the actual output values of the underlying physiological model.

We dispose of technologies and know-how for development multimedia web-based educational simulators based on simulation models. Simulators from all stages of our research can be seen on our Online Atlas of Physiology and Pathological Physiology [2]. Many of them are now technologically outdated and may serve as a documentation of our progress. The technologies still evolve and we are planning to extend the Atlas with more textbook-like chapters containing simulators and simulations games as a part of the reading, with examinations using simulators and to accent the social aspect of using the web atlas.

## REFERENCES

[1] 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)

[2] Jiří Kofránek, Stanislav Matoušek, Michal Andrlík, Petr Stodulka, Zdeněk Wünsch, Pavol Privitzer, Josef Hlaváček, Ondřej Vacek. **Atlas of physiology and pathophysiology - internet simulation playground**. EUROSIM 2007, Ljubljana, Slovenia

[3] Fritzson P. **Principles of Object-Oriented Modeling and Simulation with Modelica 2.1**, ISBN 0-471-47163-1,Wiley-IEEE Press, 2003

[4] Guyton AC, Coleman TA, and Grander HJ. **Circulation: Overall Regulation.** Ann. Rev. Physiol., 41, s. 13-41, 1972