WEB MULTIMEDIA SIMULATION FOR BIOMEDICAL TEACHING

J. Kofránek[1], *P. Privitzer, M. Mateják, S. Matoušek*

[1] First Faculty of Medicine, Charles University in Prague

Abstract

We present the current state of the technology used for web multimedia educational simulator development. The main aim is to provide novel interactive multimedia application that can be used for biomedical education where (a) simulations are combined with tutorials, and (b) the presentation layer is simplified while the complexity of the model is kept beneath. The development of the multimedia teaching simulators required the cooperation of many professionals including teachers, system analysts, artists, and programmers. During the design of the multimedia simulators, tools were developed that allow for component-based creation of simulation models, creation of interactive multimedia and their final coordination into a compact unit based on the given design. The result of our works is Atlas of physiology and pathophysiology as freely available online application, which can help to explain the function of individual physiological systems and the causes and symptoms of their disorders.

Key words: e-learning, interactive simulation, internet, multimedia, web

Introduction - Schola Ludus for the 21st Century

"Tell me, I'll forget, show me and I may remember; involve me and I'll understand" – this ancient Chinese wisdom is also confirmed by modern learning methods, sometimes called "learning-by-doing", where simulation plays are widely applied. Simulation plays make it possible to test the behavior of the simulated object without any risk – for example, try to land with a virtual airplane or, as is the case of medical simulators, treat a virtual patient or test the behavior of individual physiological subsystems.

The connection of the Internet and interactive multimedia environment with simulation models provides quite new pedagogical opportunities, particularly when it comes to explaining complex interconnected relationships, active exercising of practical skills, and verifying theoretic knowledge. The old credo of John Amos Comenius "Schola Ludus" – i.e. "school as a play" [4] pioneered by this European pedagogue as early as in the 17th century finds its application in the incorporation of multimedia educational games in training courses.

Similarly as the theoretic foundation of an air simulator is based on an airplane model, medical simulators are based on a sufficiently truthful model of physiological systems in the human body.

Complex integrative simulators for medical education

Models used as the theoretic foundation of medical trainers include mathematical models not only of individual physiological subsystems, but also their interconnections, thus forming a more complex unit.

Coleman and Randal [5] created the model “Human” intended especially for educational purposes. The model (implemented in Fortran) allowed for simulating numerous pathological conditions (cardiac and renal failure, haemorrhagic shock etc.), as well as the effect of some therapeutic interventions (infusion therapy, effect of some drugs, blood transfusion, artificial pulmonary ventilation, dialysis etc.). Recently, Meyers et al. [19] made the original Coleman’s model available on the web using Java implementation. Extensive training simulator Quantitative Circulatory Physiology (QCP) [1] is an extension of the Human model. The simulator proved to be useful in the teaching practice [22]. The recent simulator HumMod (formerly called Quantitative Human Physiology - QHP) [9, 10, 11] with its more than 4000 variables apparently represents today the most extensive integrated model of physiological regulations. The model includes a menu branched abundantly, and it supports the simulation of numerous pathological conditions including the effect of any therapy. Unlike the previous simulator QCP whose mathematical background is hidden from the user in the source code of the simulator written in C++, the simulator HumMod has taken a different way. Its authors decided to separate the simulator implementation and description of the model equations in order to make the model structure clear for a wider scientific community. The model HumMod is distributed in its source form as open source (the model and the simulator are available to the public at the website http://hummod.org). Its structure is written in a special XML language and incorporates 3235 files located in 1367 directories. Thanks to this fact, the model equations and their relationships are comprehensible with difficulty, and many research teams dealing with the development of medical simulators therefore prefer to use older models of complex physiological regulations for their further expansions – for example, the classical models of Guyton of 1972 [8], and Ikeda’s models of 1979 [12]. This is the path taken, for example, by the

SAPHIR (System Approach for Physiological Integration of Renal, cardiac and respiratory control) project international research team as the source texts of QHP model seemed very poorly legible and difficult to understand to the project participants [25]. Similarly, Mangourova et al. [18] recently implemented an older Guyton's model of 1992 [2] in Simulink, rather than the most recent (poorly legible for them) version of the model QHP/Hummod of the team of Guyton's collaborators and students.

We were not discouraged and have established cooperation with the American authors. We have designed a special software tool QHPView [15] that creates a clear graphic representation of the mathematical relationships used, from thousands of files of source texts of the model. Besides others, this has also been helpful in discovering some errors in the model HumMod. Together with the American authors, we are of the opinion that source texts of the models that are the foundation of medical simulators should be publicly available given that they are the result of theoretic study of physiological regulations – then it is easy to find out to what extent the model corresponds to the physiological reality. The structure of our model called "HumMod-Golem edition" is published at the project website (http://physiome.cz/Hummod) in its source form, together with the definitions of all variables and all equations. Unlike the American colleagues, our model is implemented in Modelica, which makes it possible to provide a very clear expression of the model structure.

The model Hummod has been modified and expanded particularly in the field of blood gas transfer modeling and modelling of the homeostasis of the inner environment, especially of acid-base equilibrium – considering that disorders precisely of these subsystems occur frequently in acute medicine for which our simulator and educational simulation plays have been designed. Besides others, our modifications stemmed from our original complex model of physiological regulations, namely the core of the educational simulator Golem [13].

Atlas of Physiology and Pathophysiology - simulation games on the Web

However, experience in application of complex models (of the Golem or QCP type mentioned above) in teaching shows that large and complex models are connected with a disadvantage from the didactic point of view, namely their complex control. The large number of input variables as well as the broad scale of options of observing the input variables require rather thorough understanding of the very structure of the simulation model on part of the user, as well as knowledge of what processes should be observed in simulations of certain pathological conditions. In the opposite case, a complex sophisticated model seems to the user only as a “complicated and not very understandable technical play” (similarly as if the user should face a complex airbus simulator without a prior theoretical instruction).

Instruction models (and apparently not only complex ones with hundreds of variables) in themselves therefore are not enough for efficient use in teaching. They must be accompanied by explanation of their application – using interactive educational applications at best. The possibility of using all advantages of virtual reality to explain complex pathophysiological processes arises only upon establishing connection between explanation and interactive simulation. In order to link the possibilities offered by interactive multimedia and simulation models in medical teaching, we have designed the concept of an Internet computer project, the Atlas of Physiology and Pathophysiology [13, 16], conceived as a multimedia instruction aid that should help to explain, in a visual way using the Internet and simulation models, the function of individual physiological subsystems, the causes and manifestations of their disorders – see http://physiome.cz/atlas. The Atlas thus combines explanation (using audio and animation) with interactive simulation play with physiological subsystems models, all available for free from the Internet.

Our Atlas is a part of the MEFANET network (MEdical FAculties NETwork), collecting electronic study textbooks and texts of medical universities in the Czech Republic and Slovakia Republic (http://www.mefanet.cz/index-en.php) The Atlas combines interactive lectures and chapters and simulation games with models of physiological systems. During the creation of the model user interface, used as a base for simulation games, the atlas looked more like an atlas of animated pictures from the regular, printed Color Atlas of Physiology [23] or the printed Color Atlas of Pathophysiology [24], rather than abstract regulation schemes used during biomedical classes. However, contrary to the printed illustrations, pictures creating the multimedia user interface in simulators are „alive“ and interactive – changes in parameters or variables will change the picture look as well. Thanks to interactive illustrations we may create simulation games which, better than regular still pictures or simple animations, explains the dynamical relations in physiological systems and help students to understand the causes or reasons that are involved in the development of pathogeneses in various diseases.

The Atlas uses the possibilities offered by the connection between interactive multimedia and simulation models, combines the simulations with tutorials, and simplifies the presentation layer while keeps the complexity of the model beneath. It is conceived as a multimedia instruction aid that should help to explain, in a visual way and using simulation models, the function of individual physiological subsystems and the causes and manifestation of their disorders. The Atlas thus combines an explanation of the physiological subsystems using audio and animation with interactive simulation (interactive for presentation layer, simulation for underlying model); it can be accessed at http://www.physiome.cz/atlas/index_en.html.

During the time of the Atlas development, it has been proposed new technologies allowing component-based creation of simulation models, creation of interactive multimedia, and their interconnection into a compact unit. The prerequisite for development of the Atlas was also the creation of a number of mathematical models of physiological systems and appropriate tools which enable to facilitate the design and sharing of the multimedia interactive educational simulators.

Atlas project is open-based, meaning its results are available for all those interested. During the course of its development, we welcome cooperation with all who would like to take part in its gradual building process.

From a mathematical model to a web-based multimedia simulator

Several key points should be considered when creating web-based multimedia enabled interactive simulators, such as those that are part of the Atlas of Physiology and Pathophysiology:

1. The underlying mathematical model needs to be formulated based on known physiological relationships. The model (i.e. set of equations that simulate the behaviour of the underlying object) is often implemented based on the verified models published in biomedical literature, but it can also represent original theoretical scientific work. In the past, simulation models were typically created in the same environment as the simulator itself (e.g. in the languages Fortran, C++ or Java). Today, special modelling tools can be used. Our team has been using the Matlab/Simulink (The Mathworks Inc.) environment on a long-term basis. We have created a Simulink library of formalized physiological relationships, named Physiology Blockset (available on http://www.physiome.cz/simchips). Recently, we have started implementing and creating mathematical models in an environment based on the language Modelica [15]. An essential innovation introduced by Modelica lies in its declarative and thus acausal definition of

models. Individual parts of the model are described directly as equations and not as an algorithm to solve the equations. Modelica uses interconnected components in which equations are defined [6, 26].

2. The simulator itself is created based on the underlying mathematical model and the teaching goals; it is more programming rather than modeling work. In our case, the simulator generally has a three layer architecture known as MVC (model-view-controller) [3, 17]; the layers include the user interface with interactive animations, the control layer and the simulator core. The simulation core is obtained by converting a debugged model from the modeling tool (Simulink or modeling tool based on the Modelica language) to the simulator development environment (ActionScript, Microsoft Visual Studio, etc.). The conversion can be done manually in the case of simpler models, but manual conversion of more complex models would be a tedious and error-prone job. We have created two software tools to convert the debugged models automatically (see Table 1). The control layer connects the simulation core with the interactive animations of the user interface and assures correct application logic. This layer is not needed in simpler simulators. The user interface is created in cooperation of the simulator programmer with a graphic designer and a teacher. It can evoke pictures from image-based medical textbooks.
3. In order for an educational simulator to look professional, a graphic designer should be the author of the user-interface animations. Our team has developed special tools that allow testing of the animation properties and subsequent connection of these interactive animations with the other model layers (e.g. Animtester made for Microsoft Expression Blend). Hence, the artist is required to be proficient not only in standard technologies (e.g. Adobe Flash or Microsoft Expression Blend), but also in these testing tools. We have put substantial effort into the training of our artists in both areas.
4. An ideal means of educational material deployment has been the Internet. Uncomplicated user accessibility and relative ease of updates are among its advantages over portable electronic media. However, if a larger number of users get connected one-by-one to a more complicated simulation model then its placement on the server may bring problems with the server performance. In this situation, it is more effective to use the computing power of the client computers for running the model. We therefore used a technology, where the simulator is automatically downloaded, transparently installed and run securely in a Sandbox.

Web simulator creation technology

Three technology chains (and pertaining work-flows) have been used during the development of the Atlas. They are illustrated in Table 1, while Table 2 summarizes the advantages and disadvantages of each option.

Simulators based on a relatively simple mathematical model were implemented using the Flash Player platform, which allows them to be run directly in the browser window. Same platform was used for the explicatory lectures of the Atlas. The simulation kernel of the simulator was created in the Action Script language of the development tools Adobe Flash and Adobe Flex. Same tools were used for the design of the interactive animations.

More complex simulators were implemented using the platform .NET. We developed a software tool that allows automatic conversion of the mathematic model implemented in Matlab Simulink into language C#, and thus creating the simulator kernel. The created simulators can be installed on the client computer with one click using the technological solution ClickOnce [20]. The ClickOnce technology provides a way of running an application by clicking on an internet link. After confirming the security level, the application is automatically downloaded, transparently installed and run in a Sandbox.

Table 1: Atlas of Physiology and Pathophysiology: Technologies used in the development of interactive Simulator

Simulator platform	**Modeling environment**	**Model conversion to simulator development tool**	**Simulator development tool**	**Animation development tool**	**Simulator development tool**
Adobe Flash	Simulink	Manual	ActionScript	Adobe Flash	Internet browser (Flash plugin)
.NET	Simulink	Authomatic	Microsift Visual Studio	Adobe Flash	Intalled localy fron the internet
Silverlight	Modelica	Authomatic	Microsift Visual Studio	Microsof Expression Blend	Internet browser (Silverlight plugin)

Table 2: Atlas of Physiology and Pathophysiology: Technologies used in the development of interactive Simulator

Simulator platform	**Positives**	**Negatives**
Adobe Flash	No need for simulator installation, simulator runs in the internet browser with Flash Player plugin. Runs under various OS. Rich support for visual aspect of the user interface.	Manual conversion from Simulink to ActionScript. Simulation kernel is erelatively slow.
.NET	Authomatic generation of the simulation kernel form Simulink. Fast simulation kernel enables creation of computationally demanding simulators.	Simulator runs under Microsoft Windows only. Need for simulator installation on the client.
Silverlight	No need for simulator installation, simulator runs in the internet browser with Silverlight plugin. Runs under various OS. Rich support for visual aspect of the user interface. Automatic generation of the simulation kernel from a Modelica based environment. Declarative model description (using equation based modeling approach) in Modelica based environment. Common platform of the user interface, animations and the simulation kernel. Animtester tool provides division between programming and the graphical design of the interactive animations.	Silverlight plugin is less widespread than Flash Player plugin.

The simulators implemented in the platform Silverlight are the result of our most recent work-flow and technological chain. These simulators can be computationally demanding and yet can be run in the browser window under various OS; the prerequisite is the Silverlight plugin in the Internet browser. The underlying mathematical model is first implemented in the modelling tool based on the language Modelica. Typically, Modelica based tools can generate simulator kernels in the C++ language. However, simulators that contain parts of the code in C++ are not allowed to run in the browser window due to security requirements.

As active participants on the Open Source Modelica Consortium [21], we have designed and implemented a code generator templating language that enables multi-targeting of the compiler output [7], and we have developed templates for C# code generation from Modelica models. This solution allows automatic conversion of a Modelica formulated acausal model into a C# formulated simulator kernel and we can produce pure .NET code able to run even under strict security requirements. Besides these improvements, the technology allows simulators of a more compact structure. The animations are created by graphic designers in Microsoft Expression Blend – a tool that communicates well with the rest of the platform. The already mentioned Animtester tool provides considerable support for cooperation of graphics and programmers. Its interface separates (and connects) the graphic design and simulator programming. The artist can create complex animations comfortably and the animations can be controlled easily. The programmer specifies the animation control by connecting it to relevant simulator modules.

The Atlas is composed of explicatory chapters and web-based simulators. The explicatory chapters of the Atlas are designed as audio lectures accompanied by interactive multimedia images. Every animation is synchronized accurately with the explanatory text. Some simulators combine the model with the explicatory part. The simulator of mechanical properties of muscles is an example.

Other simulators can be run separately and scenarios used in their control are planned as part of relevant explanatory chapters. The complex model of blood gases transport is an example; this model can be used as an instruction aid in explaining the physiology and pathophysiology of oxygen and carbon dioxide transport (e.g. to explain the consequences of ventilation-perfusion mismatch). This simulator can be downloaded from our Atlas using the following link: http://physiome.cz/atlas/sim/BloodyMary.

Conclusion – from enthusiasm to technology and multidisciplinary cooperation

The times of enthusiasts who created the first educational programs at the turn of the 80ies, excited about the new potential of personal computers, has long been gone. Today, the design of good-quality educational software capable of utilizing the potential offered by the development of information and communication technologies is not built on the diligence and enthusiasm of individuals. It is a demanding and complicated process of a creative team of specialists from various professions: Experienced teachers whose scenarios provide the foundation of a good-quality educational application; system analysts responsible, in cooperation with professionals of any given field, for the creation of simulation models for educational simulation plays; artists who design the external visual form of the simulator; and finally, information science specialists (programmers) who "stitch up" the whole application to its final form.

For such interdisciplinary cooperation to be efficient, numerous developmental tools and methodologies are needed for every stage of development; such tools and methodologies make the work of individual team members easier and help them to overcome interdisciplinary barriers. Considerable efforts must be devoted to the process of creating and mastering the tools, but it pays in the end. The process of educational program design thus acquires ever more features of engineering design work.¨

Acknowledgement

Work on development of web multimedia simulators has been supported by the grants MSM 2C06031 („eGolem"), MSM 0021620806, MPO FR—TI3/869 and SVV-2011-262507.

References

[1] cellML. (2010). Description of Guyton 1992 Full cardiovascular circulation model. [Online] http://models.cellml.org/exposure/cd10322c000e6ff64441464f8773ed83/Guyton_Model_1-0.cellml/view

[2] Collins, D. (1995). Designing object-oriented user interfaces, Redwood City, CA: Benjamin Cummings, 1995.

[3] Comenius, J. A. (1656) Schola ludus seu Encyclopaedea Viva. Sarospartak, 1656.

[4] Coleman,T.G. & Randall, J.E. (1983). HUMAN. A comprehensive physiological model, The Physiologist, 26: 15-21.

[5] Fritzon, P. (2003). Principles of object-oriented modeling and simulation with Modelica 2.1. Wiley-IEE Press. ISBN 0-471-47163-1.

[6] Fritzson, P., Privitzer, P., Sjolund, M. & Pop, A. (2009). Towards a Text Generation Template Language for Modelica, Proceedings 7th Modelica Conference, Como, Italy, 193-207.

[7] Guyton, A. C., Coleman, T. G., & Grander, H. J. (1972). Circulation: Overall regulation. Ann. Rev. Physiol., 41: 13-41.

[8] Hester, R. L., Coleman, T., & Summers, R. L. (2008). A multilevel open source model of human physiology. The FASEB Journal, 22: 756.

[9] Hester, R. K., Summers, R. L., Iliescu, R., Esters, J., & Coleman, T. (2009). Digital Human (DH): An integrative mathematical model of human physiology. Proceedings of MODSIM World Conference NASA/CP-2010-216205, 129-134. NASA, 2009

[10] Hester, R. L., Ilescu, R., Summers, R. L., & Coleman, T. (2010). Systems biology and integrative physiological modeling. Journal of Physiology, published ahead of print December 6, 2010, doi:10.1113/jphysiol.2010.201558, pp. 1-17.

[11] Ikeda, N., Marumo, F., & Shirsataka, M. (1979). A Model of overall regulation of body fluids. Ann. Biomed. Eng., 7: 135-166.

[12] Kofránek, J., Anh Vu, L. D., Snášelová, H., Kerekeš, R., & Velan, T. (2001). GOLEM – Multimedia simulator for medical education. V L. Patel, R. Rogers, & R. Haux (Editor), MEDINFO 2001, Proceedings of the 10th World Congress on Medical Informatics., 1042-1046. London: IOS Press.

[13] Kofránek, J., Matoušek, S., Andrlík, M., Stodulka, P., Wünsch, Z., Privitzer, P., Hlaváček, J., & Vacek, O. (2007). Atlas of physiology - internet simulation playground. V B. Zupanic, R. Karba, & s. Blažič (Editor), Proceedings of the 6th EUROSIM Congress on Modeling and Simulation, Vol. 2. Full Papers (CD ROM), MO-2-P7-5: 1-9. Ljubljana: University of Ljubljana

[14] Kofránek, J., Mateják, M., & Privitzer, P. (2010). Web simulator creation technology. Mefanet Report, 3: 32-97. Accessible on-line http://www.physiome.cz/references/mefanetreport3.pdf.

[15] Kofránek, J., Matoušek, S, Rusz, J., Stodulka, P., Privitzer, P., Matejak, M. & Tribula, M. (2011). The Atlas of physiology and pathophysiology: Web-based multimedia enabled interactive simulations, Comput. Methods Programs Biomed.(2011), doi:10.1016/j.cmpb.2010.12.007.

[16] Krasner, G. E. & Pope, S. T., (1968) A cookbook for using the model-view controller user interface paradigm in Smalltalk-80, Journal of Object-Oriented Programming, 1: 26-49.

[17] Mangourova, V., Ringwood, J., & Van Vliet, B. (2010). Graphical simulation environments for modelling and simulation of integrative physiology. Computer Methods and Programs in Biomedicine, Article in press, doi:10.1016/j.cmpb.2010.05.001, 10 pp.

[18] Meyers,R.D. & Doherty, C.L.. (2008). Web-Human physiology teaching simulation (Physiology in health, disease and during therapy), Available online: http://placid.skidmore.edu/human/index.php).

[19] Microsoft, (2008). ClickOnce Deployment for Windows Forms Applications. [Online], Available: http://msdn.microsoft.com/en-us/library/wh45kb66.aspx.

[20] Open Source Modelica Consortium (2010), [Online], Available: http://www.openmodelica.org/index.php/home/consortium/

[21] Rodriguez-Barbero, A., & Lopez-Novoa, J. M. (2009). Teaching integrative physiology using the quantitative circulatory physiology model and case discussion method: evaluation of the learning experience. Advances in Physiology Education, 32: 304-311.

[22] Silbernagl, S., & Despopoulos, A. (2009). Color Atlas of Physiology. 6th edition, Georg Thieme Verlag, Stuttgart, 2009, ISBN 978-3-13-54006

[23] Silbernagl, S., & Lang, F. (2009). Color Atlas of Pathophysiology 2nd edition, Georg Thieme Verlag, Stuttgart, 2009. ISBN 978-3-13-116552-7

[24] Thomas, R. S., Baconnier, P., Fontecave, J., Francoise, J., Guillaud, F., Hannaert, P., Hernandez, A, LeRolle, V., Maziere, P., Tahi, F, & White, R. (2008). SAPHIR: a physiome core model of body fluid homeostasis and blood pressure regulation. Philosophical Transactions of the Royal Society, 366, pp. 3175-3197.

[25] Tiller M.,M. (2001). Introduction to physical modeling with Modelica. Kluver Academic Publishers, Boston, 2001. ISBN 978-9-7923-9367-4.