Simulation Applications in Medical Education

Pavol Privitzer, Jan Šilar, Tomáš Kulhánek, Marek Mateják, Jiří Kofránek
Charles University in Prague, First Faculty of Medicine, Institute of Pathological Physiology, Laboratory of Biocybernetics, U nemocnice 5, 128 53 Praha 2, Czech Republic

Abstract. Abstract - Simulation applications are effective tools in medical education. They can explain the function of individual physiological systems in a very visual and interactive way. Development of simulation applications requires cooperation of many domain experts. Teachers design intended educational scenarios, physiological modellers create simulation models, computer graphic designers designs visual appearance and programmers assemble all components together into one application. Each domain expert has their specific development tools. These tools produce different outputs that are not easily composable into one application. To resolve this problem a special integration tools and frameworks are needed.

Keywords. Education, medicine, physiology, simulation, WEB, domain expert, frameworks, software development

Introduction – Integrative Physiology and HumMod Model

Physiological models are composed of individual physiological subsystems and their interconnections, thus forming more complex units.

Coleman and Randal [3] created the model named "Human" intended especially for educational purposes. The model (implemented in Fortran) allowed simulation of 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. [12] 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 education [14]. The recent simulator HumMod (formerly called Quantitative Human Physiology - QHP) [6, 7, 8] with its more than 4000 variables represents today the most extensive integrated model of physiological regulations. It supports the simulation of numerous pathological conditions including the effect of many therapeutic scenarios. 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 approach. Simulator implementation and equation description are separated in order to make the model structure readable for a wider scientific community. The model HumMod is distributed as open source (available at http://hummod.org). Its structure is written in a special XML language and incorporates 3235 files located in 1367 directories. Due to this fact, the model equations and their relationships are difficult to comprehend.

We have established cooperation with the American authors. We have designed a special software tool QHPView [10] that creates a clear graphic representation of the mathematical relationships, from the XML source files of the model. Besides other benefits, this has also been helpful in discovering some errors in the HumMod model.

In our group we have developed version of the original HumMod model called "HumMod-Golem edition"[16] implemented in Modelica language. This model 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 modeling 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 [9].

1. Modelica Language

We use Modelica language to describe the mathematical model. This language is equation based, thus individual parts of a model are described by equations. The language is quite natural and close to the way of describing the model on a paper, which is great advantage.

The well known signal based approach, used e.g. by Simulink, describes the system by set of filters (addition, multiplication, integration...) that are successively applied on the input variables is less descriptive.

Modelica uses interconnected components in which equations are defined [4, 15].

2. Framework – Modelica to .Net Translation, Simulation Runtime, Application Runtime

In our case, the simulator generally has a three layer architecture known as MVC (model-view-controller) [2, 11]; the layers include the user interface with interactive animations, the control layer and the simulator core. The simulation core is obtained by converting model in Modelica to target language (C/C++ or C#).

We have created software tools to convert the debugged models automatically. The control layer connects the simulation core with the interactive animations of the user interface and assures correct application logic. The user interface is created in cooperation of the simulator programmer with a graphic designer and a teacher.

Typically, Modelica based tools can generate simulator kernels in the C/C++ language only.

However, simulators that contain parts of the code in C++ are not allowed to run in the internet browsers due to security requirements. As active participants on the Open Source Modelica Consortium [13], we have designed and implemented a code generator templating language that enables multi-targeting of the compiler output [5], and we have developed templates for C# code generation from Modelica models. This solution allows automatic conversion of a Modelica formulated model into a C# simulation code and we can produce pure .NET code able to run even under strict security requirements.

3. Tools for graphic designers – Blend and Adobe Creative Suite

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 - see figure 1. 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.

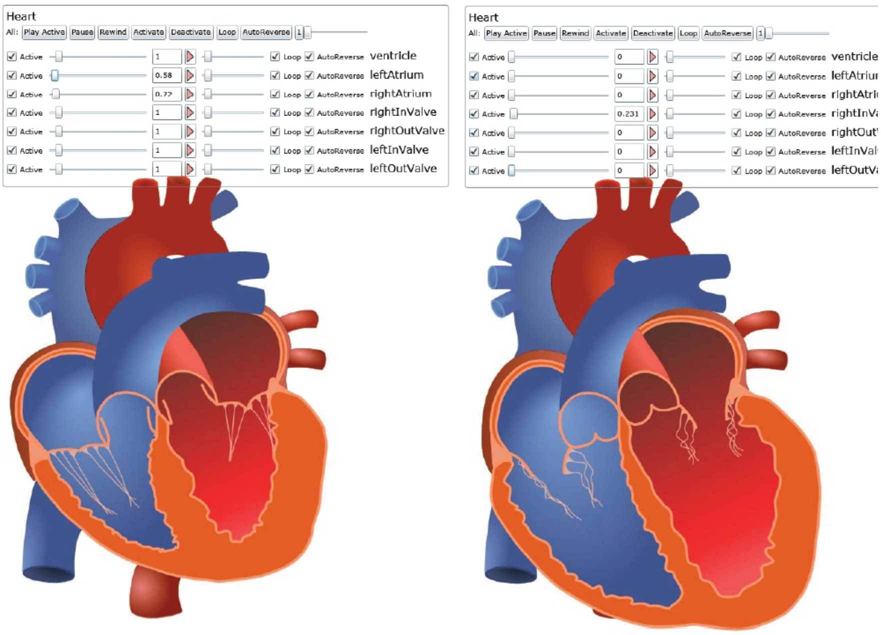

Fig 1: Beating heart.

The animations are created by graphic designers in Microsoft Expression Blend – a tool that communicates well with the rest of the platform. We have created tool named Animtester that provides considerable support for cooperation of graphics and programmers. 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 resulting simulation applications are implemented on the Silverlight platform. 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.

Discussion

The design of good-quality educational software capable of utilizing the potential offered by the development of information and communication technologies is demanding and complicated process of a creative team of specialists from various professions: Experienced teachers, system modellers, artists and finally information science specialists (programmers).

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.

References

[1] Abram, S.R., Hodnett, B.L., Summers, R.L., Coleman T.G. & Hester, R.L. (2007). Quantitative circulatory physiology. An integrative mathematical model of human mathematical model of human physiology for medical education, *Advanced Physiology Education,* 31: 202-210.

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

[3] Coleman,T.G. & Randall, J.E. (1983). HUMAN. A comprehensive physiological model, The Physiologist, 26: 15-21. [4] Fritzon, P. (2003). *Principles of object-oriented modeling and simulation with Modelica 2.1.* Wiley-IEE Press. ISBN 0-471-47163-1.

[5] 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. [6] Hester, R. L., Coleman, T., & Summers, R. L. (2008). A multilevel open source model of human physiology. *The FASEB Journal*, 22: 756.

[7] Hester, R. K., Summers, R. L., Ilescu, 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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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).

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

[14] 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.

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

[16] J. Kofranek, M. Matejak, and P. Privitzer. Hummod-large scale physiological models in modelica. Proceedings 8th Modelica Conference, Dresden, Germany, March 20-22, 2011, 713-724, Linkoping Electronic Conference.

[17] J. Kofranek, P. Privitzer, M. Matejak, and S. Matoušek. Use of Web Multimedia Simulation in Biomedical Teaching, Worldcomp 2011, Proceedings of the 2011 International Conference on Frontiers in Education: Computer Science & Computer Engineering, Las Vegas, July 18-21, 2011, (H. R. Arabia, V. A. Cincy, L. Deligianidis, Eds.), ISBN 1-60132-180-5, CSREA Press, Las Vegas, Nevada, 2011, 282-288, online: http://world-comp.org/p2011/FEC3582.pdf