What is behind the curtain of a multimedia educational games?

Jiří Kofránek
Institute of Pathophysiology, First Faculty of Medicine, Charles University in Prague, U nemocnice 5, 128 53 Prague 2, tel +420-777686868
kofranek@gmail.com

ABSTRACT
Multimedia programs with simulation components for educational purposes are not only a replacement of classical textbooks. They serve as an entirety new instruction aid making it possible to explore the studied problem in virtual reality through instruction simulation games, thus bringing quite new possibilities to explain complex problems. Atlas of Physiology and Pathophysiology (http://www.physiome.cz/atlas), designed is a web accessible multimedia-teaching tool with simulation games, which helps to explain the function of individual physiological systems, causes and symptoms of their disorders. Development of the Atlas requires cooperation of many professionals. Starting from experienced teachers whose design provides the foundation of quality educational applications, system analysts responsible for creating simulation models for educational simulation games in cooperation with professionals in their field, artists creating the visuals, and finally up to programmers who “knit” together the whole application to achieve its final form. For the inter-disciplinary collective creation to be successful, specific development tools with sufficient technical support must be used in each phase of creation; such tools allow for component-based creation of simulation models, creation of interactive multimedia and their final interconnection into a compact unit based on the given design. Creative interconnection of the various professions is the key to success.

Keywords
E-Learning, Education, Physiology, Multimedia, Simulation Games.

1. INTRODUCTION: SIMULATION AS A EDUCATIONAL GAME
An old Comenius’s motto – “schola ludus” (“school as a play”), nowadays finds its modern usage in the interactive educational programs using simulation games. The connection of the Internet, a multimedia environment, which serves as a sound and visual user interface, with simulative models enables, after connecting to the magic Internet network, to graphically “touch” the lectured on problem in the virtual reality.

Through the means of a simulation game it is possible to test, without any risk, the behaviour of the simulated object – e.g. trying to land with a virtual airplane or to heal a virtual patient. Through the simulation game we can test the behaviour of individual physiological subsystems, both in normal conditions and when there is a disorder.

Many instruction-oriented simulators of individual physiological subsystems for free pedagogical use can be found on the Internet. Thus for example, the simulator ECGsim (http://www.ecgsim.org) makes it possible to study the generation and spreading of electric potential in heart ventricles and to study the mechanism of origination of the ventricular complex QRS in various pathologies (from impulse conduction disorders to ischernias and infarctions). The educational heart simulator (http://www.columbia.edu/itc/hs/medical/heartsim/) from Columbia University allows for observing the pressure-circulation curves in heart ventricles in various cardiac pathologies (valvular defects, left-sided or right-sided failure); anaesthesiological device simulators from the University of Florida (http://vam.anest.ufl.edu/) provide the possibility to administer anaesthesia to a virtual patient and to observe appropriate physiological responses (however, more complex simulators require paid access) etc.

Complex integrative simulators of human physiology are of large importance for teaching of pathophysiology and study of pathogenesis of varied pathological conditions; such simulators include models of not only individual physiological subsystems but also their mutual connection into a more complex unit.

Coleman and Randal [2] created the model “Human” intended especially for educational purposes. The model 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. [15] made the original Coleman’s model available on the web using Java implementation. The recent simulator - Quantitative Human Physiology [6] representing probably the most complex and extensive model of physiological functions at present times. The simulator is an extension of the original large circulatory system simulator achieved by integrated connection of all important physiological systems. The model can be downloaded from the Internet. We, too, created an educational simulator “Golem” in the past, based on a complex model of integrated physiological regulations [8, 9, 10]. Our simulator “Golem” was focused on teaching of complex disorders of the internal environment.
However, experience in application of complex models (of the Golem or Quantitative Human Physiology simulator 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 simulation game. 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 [11], 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 an audio explanation synchronised with animations) with interactive simulation games with physiological subsystems, all available for free from the Internet (fig. 1).

2. BEYOND THE ATLAS CURTAIN

The creation process of the Atlas takes the form of a joint work made by creative specialists team encompassing various professions:

- Experienced teachers whose scenario is the foundation of a quality educational application;
- System analysts, responsible in cooperation with professionals of the given field, for designing simulation models for educational simulation plays;
- Artists designing the outside visual form;
- Computer science engineers (programmers) whose role is to “knit” the entire application into the resulting form.

For the interprofessional collective creation process described above to be efficient, each stage of the process should utilize specific development tools, with sufficient technical support, making it possible to apply component-based creation of simulation models, interactive multimedia preparation, and their final interconnection pursuant to a given scenario into a compact unit. Creative interconnection of various professions and development tools is therefore a prerequisite of success. A multimedia presentation available from the Internet, (found at www.physiome.cz/atlas/info/01EN/index.htm) discusses the technologies used in the building process and methodology of creating educational simulators.

2.1 Foundation of an e-learning Educational Application – Scenario of Good Quality

The foundation of every explanatory chapter of the Atlas is represented by a quality scenario, designed by an experienced pedagogue. High attention must be paid to preparing the scenario. According to our experience, underestimating of thorough scenario preparation is paid for dearly by the necessity of...
unnecessary iteration steps in development of the educational application and in subsequent extension of the development time.

The scenario must include a detailed proposal of assignment for the artist concerning the graphic appearance of every individual page, including animations, and the interactive behaviour proposal. The final graphic appearance is then up to the artist cooperating with the author of the given chapter. At the same time, the scenario must include key points of synchronizing the audio track with beginnings of individual animations.

When the explanation is accompanied by a simulation play (in the form of “training” using the simulation model), the scenario according to which the students should manipulate with the model must be thought over carefully so that the model behaviour clarifies those relationships, which are rather difficult to explain without a simulation game with the model (see example of simulation game in figures 9-13).

2.2 Simulation model – theoretical fundament of simulation games

In 1972, an American physiologist A. C. Guyton and two co-authors published in the Annual Review of Physiology [4] an overview article, dealing with the regulation of circulatory system. The article was published in a biomedical journal and it had a completely unusual content for its readers. The main part of it was an inserted attachment with a vast scheme, which at first sight looked more like some kind of an electronic scheme (fig. 2). However, the individual blocks represented computing blocks - multipliers, dividers, integrators, summat ors, functional blocks. These blocks were connected to conductors, in which the values of various physiological quantities were running. They entered the individual blocks where a certain mathematical operation was carried out, and the output was a value of the counted physiological quantity that could be taken through the conductors to other computing blocks. Therefore the combination of interconnecting various blocks actually embodied a graphically expressed equation for counting certain physiological quantities. And the whole scheme actually represents a graphically expressed system of equations.

Guyton’s model was the first extensive mathematical description of physiological functions of interconnected subsystems of an organism, and it initiated development of physiological research, sometimes described today as integrative physiology. From this point of view, it was a certain milestone, which attempted at capturing the dynamics of relationships among the controls of circulation, kidneys, breathing, the volume and ionic composition of body fluids using a mathematical model, while applying a system view of physiological regulation.

Similarly to theoretical physics, where experimental research in physics and related sciences is being interpreted, integrative physiology attempts to formalize description of mutual regulations in physiology. Here, the main methodological tools are simulation models. The international project called PHYSIOME (http://www.physiome.org) and European project EUROPHYSIOME (http://www.europhysiome.org/) try to concentrate activities in this area [1,7].

Special software simulation environments are available nowadays for the development, tuning and verification of simulation models. One of them, for instance, is a development environment Matlab/Simulink from the Mathworks company. This environment enables to gradually put together the simulation model from the individual components – from some sort of software simulation parts, which can be mutually interconnected with the aid of a mouse into simulation networks. It is interesting that the elements that Guyton used many years ago in his vast graphic scheme (i.e. multipliers, dividers, summat ors and functional blocks) look almost identical with the elements that Simulink uses. However, unlike the graphic picture, the simulation networks in Simulink are “alive” – it is possible to connect virtual displays or oscilloscopes to the individual conductors with the aid of a computer mouse, and watch the individual levels of simulated quantities during the simulation.

It was interesting to transfer Guyton’s original graphic scheme into a simulation network in Simulink. In order to be able to do that we first had to correct some “graphic misprints” in Guyton’s original scheme. Those mistakes do not matter on a picture, however for transferring the diagram into the Simulink form, the model would not start running. There were not many mistakes; it was possible to find them all with the knowledge of physiology and systemic analysis.

It is interesting that Guyton’s diagram has been reprinted as a complex picture many times into various publications (e.g. [5,16]). However, nobody mentioned and no one cared to correct these mistakes. It is possible that the authors of the model did not really want to correct the mistakes – the ones, who did work on
We therefore face a serious problem of routine job. From Fortran into another programming language, which was a programme anything (they only needed to transfer the programme to only test the behaviour of the model, they did not have to model in Fortran programming language – so if someone wanted at their time even sent the source texts of the programmes of their who just wanted to dully copy had bad luck. All in all, the authors analysing the model did discover the picture “misprints”, the ones these experts need to understand each other. 

A simulation model is a mathematical model created into the form of a computer program. A physiologist and a system analyst cooperate on creating and testing simulation models. Therefore these experts need to understand each other.

We therefore face a serious problem of *inter-disciplinary cooperation*. So-called *simulation chips* are some of the tools easing the mutual understanding.

Modern tools for creating simulation models, e.g. Simulink enable graphical representation of each subsystem as an icon, which can be connected by a mouse click. A simulation model can be put together using individual connecting components - so called software simulation chips. These software chips are individual modeled subsystems, and each chip includes in the front mask of the chip descriptions of its inputs and outputs (possibly, in case of interactively called help, it even contains a detailed description including mathematical relations).

The core of simulation chips can be also built with simulation chips, or directly with individual computer elements of a simulation tool that represent individual mathematical relations. Connected networks resemble electronic networks. Instead of a power stream, there is an information stream (see fig. 3 and 4).

The user does not need to know the inner structure of simulation chips. The user only needs to know the meaning of individual pins representing the model’s inputs and outputs. Simulation chips therefore operate in the similar way as electronic chips. Generators of constant or changing signals can be added to individual inputs for tools for simulation models and individual outputs of simulation chips can be connected to virtual measurement devices, so such simulation chip can be tested.

Simulation chips can be connected into networks of relationships - the advantage of simulation networks assembled in such a way is, that the structure will be understandable even for an experimental physiologist. With the help of the virtual tools he can also test the behaviour of the simulation model. Simulation chips are therefore very suitable tool for interdisciplinatory communication.

### 2.3 Causal and acausal simulation software tools

Today, Mathworks development tools (Matlab and Simulink) rank among well-established industrial standards. For many years, we were developing and debugging the mathematical models in standard tools of the company Mathworks – in the *Matlab/Simulink* environment, one of the today’s well-proven industrial standards. We have created a special library of formalized physiological relationships in this environment, Physiology Blockset, freely accessible from our website (www.physiome.cz/simchips).

In Simulink, we have implemented a mathematical model, used as a foundation for the Golem simulator [8,9,10]; we have also implemented extensive Guyton’s models in this environment [12].

As a rule, Simulink operates using connected blocks. Signals are transmitted through links between individual blocks; the signals serve to transfer values of individual variables from the output of one block to inputs of other blocks. *Input information is processed* in the blocks to output information. Interconnection of blocks in Simulink therefore reflects rather the calculation procedure than the very structure of the modelled reality. This is the so called *causal modelling*.

Recently, new, the so called “acausal” tools to create simulation models were developed. An essential innovation brought by acausal modelling tools is represented by the declarative (and thus acausal) notation of models, when *individual parts of the model are described directly as a system of equations and not as an algorithm to solve the equations*.

Acausal modelling tools work with connected components in which equations are defined. The equations do not mean assignments (i.e. saving of the result of the assigned instruction computation in the given variable) but a definition of relationships among the variables (as usual in mathematics and physics). These components (which represent class instances with equations) can be *connected* through exactly defined interfaces – *connectors*.
The important fact is that by connecting the components, connection of the systems of equations in individual components with each other actually happens.

A typical representative of acausal modelling tools is thus represented precisely by the new object-oriented programming language Modelica [3]. It was originally developed in Sweden and now, it is available both as an open-source version (developed under the auspices of the international organization Modelica Association, \( \text{http://www.modelica.org/} \)), and in two commercial implementation (from the company Dynasim – Dassault Systems, called Dymola, and from the company MathCore, called MathModelica).

In accordance with modern trends, we have therefore expanded the development tools used to create mathematical models until present (i.e. Matlab/Simulink) by tools that use the acausal modelling language – Modelica. Nevertheless, we immediately found considerable advantages compared to creating the models in the Matlab/Simulink environment, brought by this new tool, especially in making of rather extensive models [13,14]. This led us to fundamental reassessment of the present strategy, and as for further development of the simulator, we opted for the Modelica programming language as the fundamental tool to create simulation models.

2.4 Multidimensional Animations as Puppets on the Strings of Simulation Models

In order to achieve a professional resulting appearance of the application, the very animations should be created by an artist – the results are far better than if the animations are created by a programmer talented in graphics. However, this would mean to devote certain efforts to training of artists who must manage working with tools to create interactive graphics. However, there is a critical lack of artists with such an education on the labour market.

We tried to resolve the lack of interactive graphics professionals years ago already by establishing narrow working cooperation with Václav Hollar School of Arts where we have built our external workplace – Interactive Graphics Laboratory. Our activities focused on training the teachers at first (and later students, as well) of the school in the field of using modern computer graphics tools, and thanks to our joint efforts we have created a new, three-year, Vocational College specialization “Interactive Graphics” at this school (at present, there are graduates from two year-classes already). We provide especially teaching of “Interactivity Mastering” specialization at the Vocational College, as well as guidance of the students’ practical experience. The students (and also graduates today) of this Vocational College are those who provide the prevailing part of artistic appearance of our educational applications.

Until today, we have used especially Adobe products in designing interactive graphics – especially Adobe Flash and Adobe Flex. Lately we have started to turn to the Microsoft development environment, which offers very suitable tools allowing for good cooperation between programmers and artists – Microsoft Expression Blend (on part of the artists) and Visual Studio 2008 (with the WPF framework on part of the programmers).

In order to increase efficiency of creating the graphic layer, we have developed an auxiliary software tool – Animtester, which enables the designers – graphic designers to create such animation “puppets” and debug them without the need of any further programming (see Fig. 5 and 6). Thus created “puppets” can be then connected directly to model outputs and there is no need to add any separate program inter-layer for data propagation, as would be the case if using Flash animations.

2.5 Development of web accessible educational application with simulation games

Development of the educational simulator is demanding programming work, linked to the results of the mathematical model development and to the created elements of interactive graphics. In accordance with the designed scenario, graphic elements of the user interface must be “knitted” together with the mathematical model programmed in the background.

In the past, we have used the development environment Control Web to create simulators; this environment was originally designed to make industrial applications (control, management, control centres design) using a PC. Control Web provides numerous tools to create a complex user interface, allows for connecting Flash animations to the interface and to control it.

---

Fig. 5. Animation of the beating heart. The model outputs have an effect on the heart beat phases, opening and closing of the valves etc. Auxiliary control elements of Animtester are found over the very animation, enabling the graphic designer to debug individual subanimations.

Fig. 6 Created animation can be subsequently directly connected to model outputs in educational application and model can take full control of this animation.
according to values of variables on the background. A Control Web application in its classical industrial deployment form communicates through a software controller of the hardware control and measurement card with the industrial technological device (see www.mii.cz). Using the Control Web environment to create simulators, we have programmed a special software controller in whose core a simulation model is programmed. Control Web was thus “cheated”: It did not communicate with some industrial technology through the relevant software input/output channels but with a simulation model in the controller (see Fig. 7).

In order to facilitate development of “virtual measurement/control card” controllers containing a simulation model and not to have to write such a controller for each model “manually” in the C programming language, we have developed a special program that enables us to generate the source text of the relevant virtual controller in C directly from a Simulink diagram. This has allowed for simple and quick modification of the controller for Control Web upon making various adaptations and new versions of the simulation model.

For example, in Control Web we have created the simulator Golem [8,9,10] and as for the Atlas of Physiology and Pathophysiology, the kidneys simulator [11].

Currently, we are using classical programming tools to design the simulators. As far as simple Flash simulators are concerned, these are programmed directly in ActionScript, thus the programming language for Flash applications. However, the ActionScript language development environment is not sufficient for more complex simulators. Therefore we use the Microsoft Visual Studio .NET programming environment in our laboratory, which, especially its latest version, provides extensive possibilities for programming work. In this environment, we are no more limited by “preset” elements of the user interface as is the case of Control Web, and moreover we can use all the power of a modern software application development tool; however, on the other hand, we must program ourselves many elements of the user interface for the application under development.

In order to make writing of the simulators easier (and not to have to program an already debugged simulation model “manually” in Visual Studio .NET), here, too, we have developed a special software tool to automatically generate the simulation model from Simulink in the form of a component for the .NET environment (see Fig. 8).

To facilitate conversion of mathematical models from the Modelica language environment into Microsoft .NET, we are in the process of making (as part of the international project Open Modelica) our own solver – Modelica.NET.

Besides interconnection with the model creation tools, easy connection to graphic components of the user interface under development is important, as well. Flash components can be incorporated into the simulator in the process of creation through an Active X component.

The new .NET environment version also introduces entirely new possibilities of creating graphic components. Thanks to the new WPF (Windows Presentation Foundation) technology, complex graphic components can be created directly in the .NET platform, which include animations, vector graphics, 3D elements etc. (similarly as in Adobe Flash or even with potentially greater possibilities). It is important that the graphic user interface under development is directly integrated with the .NET platform, which removes the need of bridging the heterogeneous worlds of .NET and Adobe Flash in the simulators development (see Fig. 8).

New development tools of Microsoft provide a very perspective environment for development of simulators, and for the future, they represent our main development platform. Moreover, the new tool Silverlight shall make it possible to develop simulators, which can run directly in the Internet browser (even on
computers with different operating systems – it is only necessary to install the relevant plugin in the browser).

3. RESULTS: WEB-BASED EDUCATIONAL APPLICATION

The Atlas of Physiology and Pathophysiology is currently designed as a web-based application that can be run in an Internet browser (a Flash player installed in the browser is a prerequisite). Some simulation models require Microsoft .NET framework installed on the computer (if this part is not installed, its installation is offered before installing the first simulator, which requires .NET).

Explanatory chapters of the Atlas are designed as audio lectures accompanied by interactive multimedia images. Every animation is synchronized accurately with the explanatory text.

However, the Internet-based Atlas of Physiology and Pathophysiology is much more than just an animated explanation with an audio track. The foundation of didactic efficiency is represented by **explanation accompanied by a simulation game**. Simulation models forming part of the Atlas are implemented as Flash applications and need not be installed separately or (in more complex models) their separate installation is required directly from the Internet browser. More complex models require somewhat more complicated control – a suitable scenario is therefore important, according to which the model can be used in the simulation play as an instruction aid to explain more complicated physiological relationships.

![Fig. 9 Simulation game with the blood gases transport model to explain the consequences of ventilation – perfusion non-uniformity failures. Initial condition.](image)

![Fig. 10 Setting of different ventilation distribution shall cause decrease of PO2 and increase of pCO2 in mixed arterial blood.](image)

![Fig. 11 Slight increase of the breath frequency means achievement of pCO2 normalization in mixed arterial blood; however, PO2 still remains low. Different shape of O2 and CO2 dissociation curves is the cause – see the following figure.](image)

![Fig. 12 Comparison of total concentrations and partial pressures O2 and CO2 in hyperventilated, hypoventilated alveoli and in mixed arterial blood.](image)

![Fig. 13 Perfusion limitation by poorly ventilated alveoli limits admixing of hypoxygenated blood from hypoventilated alveoli, thus causing increased partial pressure of O2 in mixed arterial blood.](image)
Some simulators combine the model as well as the explanatory part, other simulators can be run separately and scenarios used in their control are designed as part of relevant explanatory chapters. The complex model of blood gases transport can be given as an example; this model shall be used as an instruction aid in explanation of physiology and pathophysiology of oxygen and carbon dioxide transport. Simulation game with the blood gases transport model to explain the consequences of ventilation – perfusion non-uniformity failures.

Initial condition is shown in fig. 9. Setting of different ventilation distribution shall cause decrease of PO2 and increase of pCO2 in mixed arterial blood (fig. 10). Slight increase of the breath frequency means achievement of pCO2 normalization in mixed arterial blood; however, PO2 still remains low (11). Different shape of O2 and CO2 dissociation curves is the cause – see comparison of total concentrations and partial pressures O2 and CO2 in hyperventilated, hypoventilated alveoli and in mixed arterial blood (figure 12). Comparison of total concentrations and partial pressures O2 and CO2 in hyperventilated, hypoventilated alveoli and in mixed arterial blood are shown in fig. 13.

4. CONCLUSION

Educational applications using simulation play, available through the web, represent a new educational aid, very efficient from the didactic point of view in explaining complex pathophysiological processes.

However, their process of creation is not very easy – it requires multidisciplinary teamwork cooperation and use of suitable development tools. Their making is a combination of research and development work. Research work consists in formalizing physiological reality by designing mathematical models, while development work is the very creation of multimedia simulators, which make use of the mathematical models designed.

5. ACKNOWLEDGMENTS

Development of the Atlas and educational simulation models takes place under the support of the MSMT (Ministry of Education, Youth and Sports) grant 2C06031 – “e-Golem” and the company Creative Connections s.r.o.

6. REFERENCES