MULTIMEDIA SIMULATION GUIDES TO CLINICAL PHYSIOLOGY

Jiří Kofránek, Michal Andrlík, Tomáš Kripner, Zdeněk Wünsch, Štěpán Svačina

Laboratory of Biocybernetics, Department of Pathophysiology,
1st Medical Faculty, Charles University,
Prague, Czech Republic,
e-mail: kofranek@cesnet.cz

Abstract: The authors present their way of designing multimedia educational software to support the teaching of pathological and clinical physiology, taking advantage of interactive simulation games. Simulation models, created using Matlab/Simulink® from MathWorks, are decomposed to so-called simulation chips suitable for interdisciplinary cooperation during design-time. Interactive animations utilize Flash and Director from Macromedia, the user interfaces of the simulators are implemented in Control Web from Moravian Instruments, all integrated by Microsoft Visual Studio .NET. The authors also give their arguments concerning their preference for industrial development tools over academic “open source” tools.

Keywords: Computer Simulation, Educational Aids, Medical Applications, Interdisciplinary Design, Physiological Models, Simulators.

1. MULTIMEDIA SIMULATION GUIDES

Computers displaced typewriters from offices some time ago, likewise it is obvious that computers will soon become a major teaching aid. The limiting factor of the deployment of computers in teaching is the lack of suitable educational software. Its authoring is by no means easy; the combination of hypertext, charts, audio, video and interactive animations has great pedagogical power to explain complex topics visually, on the other hand it poses great demands on the authors of such software.

These demands are even increased when the educational software incorporates a simulation model. Such software is not only a multimedia substitute of a classical paper text-book, it is a brand new teaching aid. Experiments on the simulation model (so called **simulation games**) provide a virtual world where one can play harmlessly with a virtual patient, offering a new range of possibilities for exploring complex relationships.

E.g., when explaining complex regulatory bonds in physiology, students can explore the behaviour of individual physiological subsystems being separated from their environment. They have the possibility to observe the reactions of the subsystems to changes in input variables (which are of course controlled in the living body and cannot be freely changed). Just one input can be changed, while keeping the other inputs constant, revealing the dynamics of the subsystem (so called “ceteris paribus” principle).

The subsystems then can be interconnected again; temporarily broken regulatory loops can be closed, so that one can study their role and influence in various pathological disorders and reactions to therapy. We can then gradually form individual subsystems into larger entities while temporarily disabling regulating linkages and studying its influence on the organism’s response to various pathological disorders and reactions to a particular therapy.

Experience shows that this approach leads to **better comprehension** of complex dynamic phenomena in pathogenesis of various diseases and understanding pathophysiological principles of the corresponding medical treatment (Kofránek et al. 2000, 2001a, 2001b, 2001c)

In our lectures we use multimedia features including interactive animations. These animations can be controlled by the outputs of a simulation model (e.g. animated lungs move to the frequency of breathing which is the output of the respiratory system simulation model).

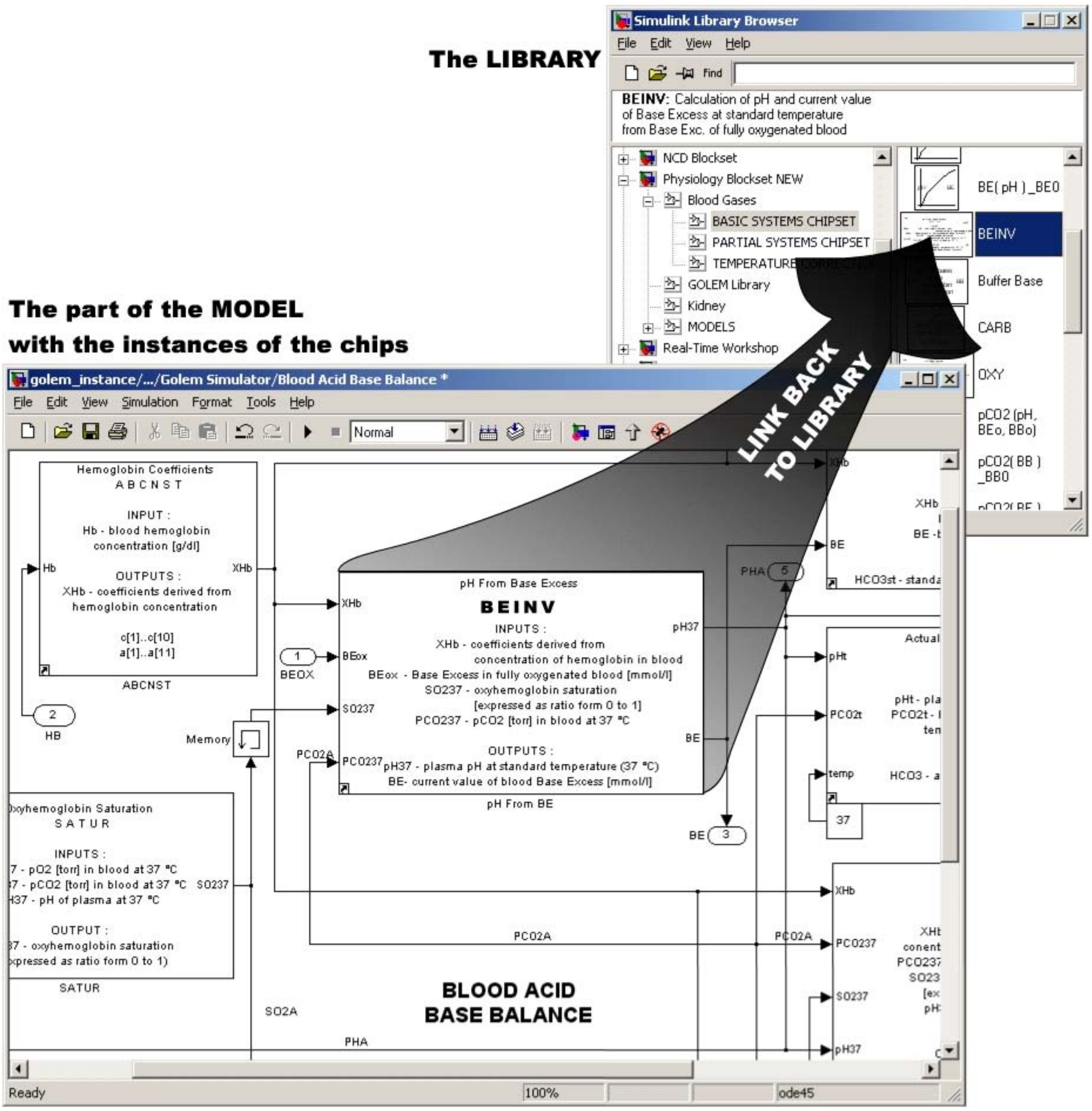

Fig. 1. The simulation chips are stored in the library (top window). The model (bottom window) contains instances of the chips. This model can be run or reconstructed as well. Every instance of any simulation chip has a link to its pattern stored in the library.

The connection of multimedia features with the simulation models forms the basis of our **multimedia simulation guides** to clinical physiology, covering the fields of acid-base and electrolyte balance, the physiology of muscles and the physiology of respiration at the moment (Kofránek et al. 2000, 2001b, 2002c). Simulation guides to the physiology of circulation and physiology of diabetes are currently in progress.

2. SCENARIOS FOR EDUCATIONAL SOFTWARE

Just as the reception of a text-book by students depends on the author's ability to explain complex material in an illustrative and comprehensive way, the key to success of multimedia educational software is a good scenario or screenplay. Without a quality scenario, all multimedia, interactive animations and simulation models are just fashion accessories. In multimedia animations, resources for the scenario comprise not only of texts, but also to the cartoon films of the "storyboard" which tells the graphic designers what animations to create.

When utilizing simulation models in educational software, one has to design the model based on a mathematical description of physical reality. As in a flight simulator with a more or less realistic model of a plane, there is a model of the human body (or some of its physiological subsystems) behind a medical simulator.

3. OBJECT-ORIENTED TOOLS FOR SIMULATORS

Creating the simulation models of physiological systems of the human body is **more a researcher than a developer task.** This governs the choice of developer tools for simulation models. Classical

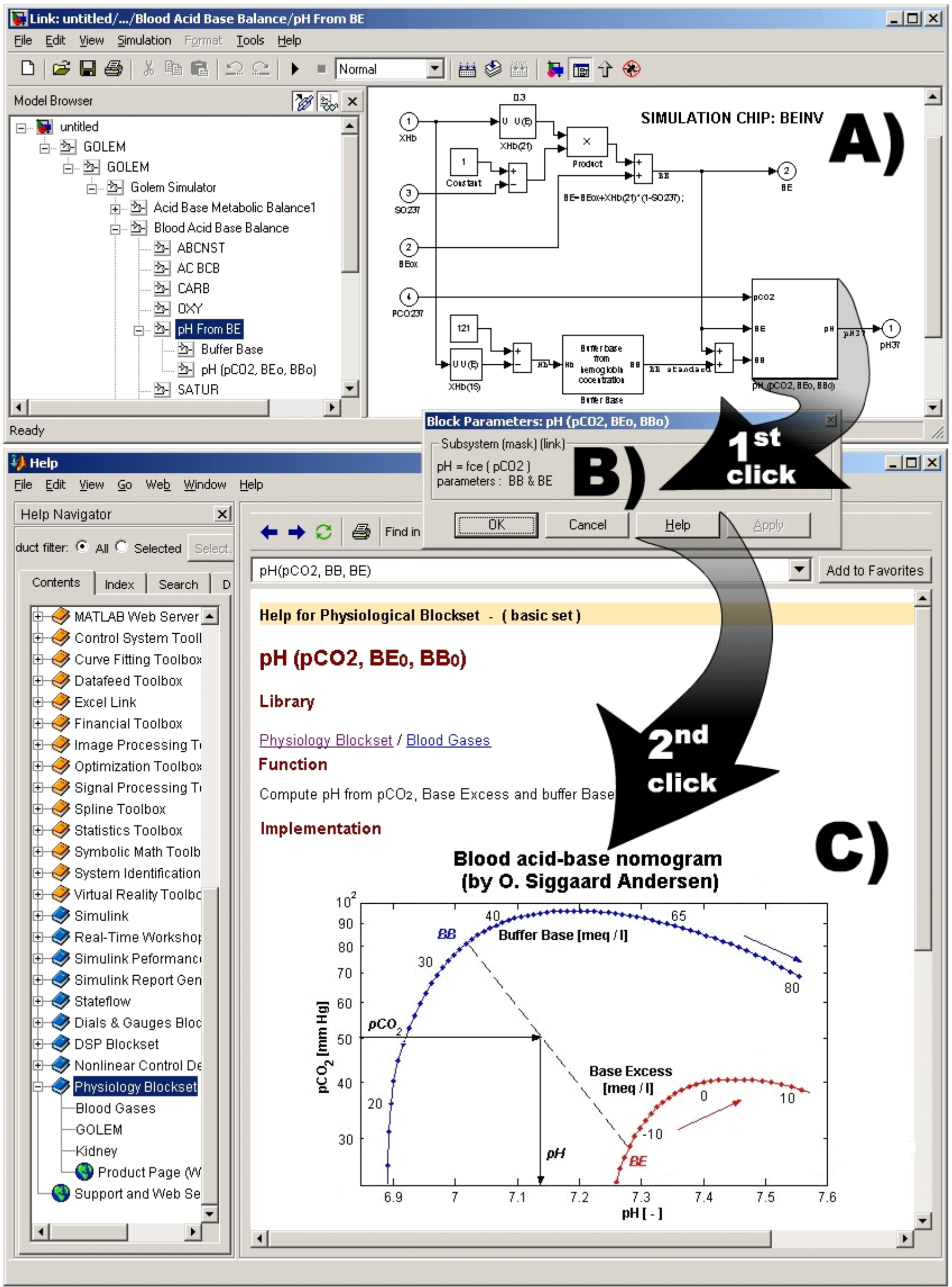

Fig. 2. A) Inside the simulation chip "BEINV" (from previous figure) are many other simulation chips (the model has a hierarchical structure). B) Using a mouse-click the user can get a window with some base information (Block Parameters) about the marked chip. C) With the next mouse-click, the user will open a dynamic help page with a complete description of the physiological meaning and mathematical implementation.

software tools like Microsoft Visual Studio or Borland Delphi are usually not sufficient.

Specialized tools for simulations and mathematical modelling include Matlab/Simulink® from Mathworks, used by our developer team. These tools allow for easy creation, tuning, identification and optimization of simulation models. Simulink provides comfortable graphical design and simulation of mathematical models. Elements of a model can be grouped and hierarchically ordered, giving rise to subsystems with user defined inputs and outputs. This graphical representation has a self-documenting power, while there are of course other tools for sophisticated documentation of a simulation model in Simulink (Hanselman and Littfield 2001, Tewari 2002)

The building of simulation models in Simulink is hierarchical and component-oriented. Subsystems of models can be stored in libraries and reused. The subsystems resemble electronic integrated circuits, with information rather than current flowing between their pins (user defined inputs and outputs). The inner structure of these so-called **simulation chips** can be hidden from the system architect; knowledge of physical quantities on the respective input and output pins is sufficient (see fig. 1).

A simulation chip can have other interconnected chips inside. The lowest level chips include the actual mathematical relationships. A large simulation chip can have a quite complex structure. The simulation chips are “alive” in Simulink. One can connect signals to the inputs of the chips while

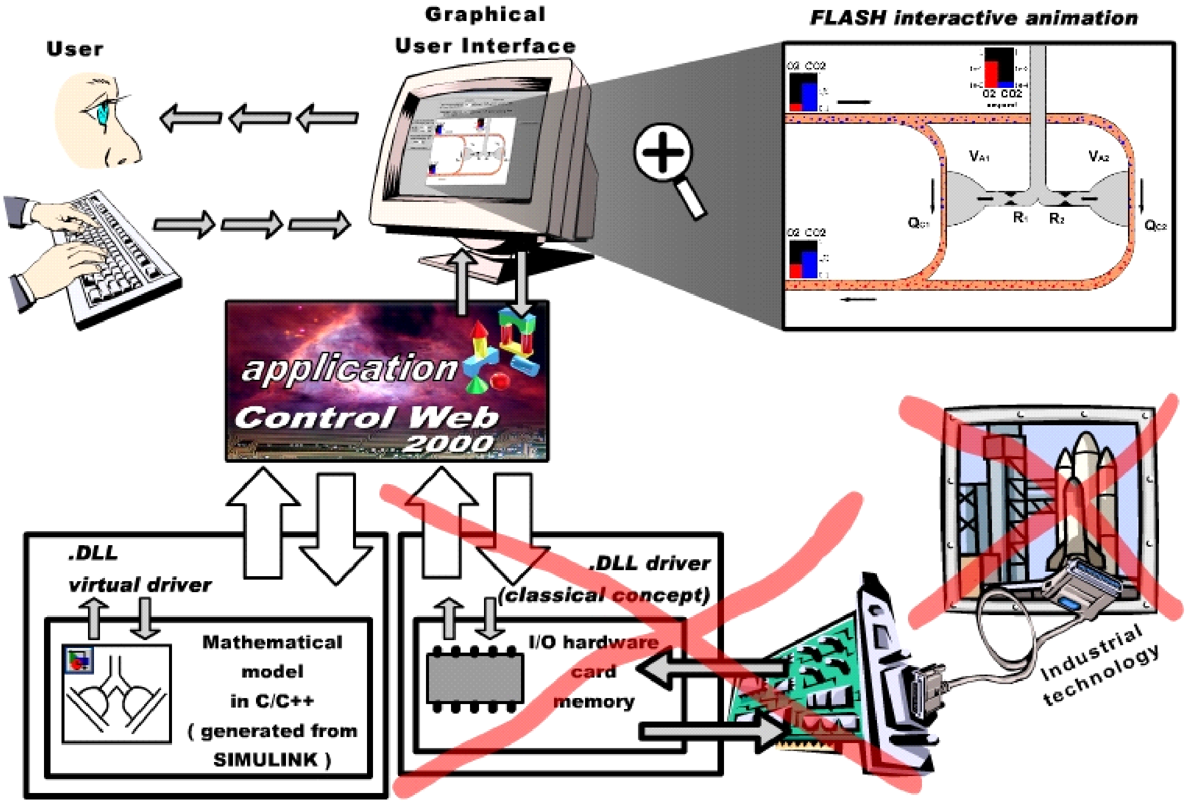

Fig. 3. The Control Web application communicates with the mathematical model through a virtual driver. In industry the Control Web application would communicate with the measuring/controlling card through a similar driver and this card would be interconnected with any industrial technology. Next, the FLASH interactive animations can be built into the user interface of the Control Web application.

measuring and displaying the outputs either numerically or graphically. Final versions of the simulation chips can be stored in libraries and easily reused (Kofránek et al. 2002a).

4. COMMUNICATION THROUGH SIMULATION CHIPS

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 a **clinician** who cannot deal with the integro-differential description of a 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.

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.

Thus simulation chips can be the means of communication among specialists of different professions.

Besides, simulation chips can represent an accurate, up-to-date electronic documentation of simulation models. The mask of a chip contains a brief description of all inputs and outputs and the inside of a chip graphically represents the network of mathematical expressions. A help window with a detailed description is also available (see fig. 2).

5. FROM SIMULATION CHIPS TO SIMULATORS

Simulation tools from MathWorks are intended for specialists and are not particularly useful for common users wanting to “play” a bit with a simulation model. Although it is perfectly possible to program a user-friendly interface to a model in Matlab/Simulink®, its licensing policy makes it not a good choice for educational simulators. Moreover, the benefits of a comfortable design environment are outweighed by the great demands on computational power when running large-scale simulation models. On less powerful computers the simulations might be inadequately slow.

Consider the similarity of a physiological model to a complex industrial device. When controlling such a device, data from its sensors have to be continually read, and appropriately displayed and stored, and the supervisor has to be able to easily control all the inputs to the technology. The requirements for running a simulation model are the same.

The graphical interface of a simulator has to be independent of the simulation kernel; the graphics, however complex they may be, must not affect the speed of simulation.

Our choice of Control Web, an industrial application development tool from Moravian instruments, as a design tool for multimedia simulators, possesses all the beneficial properties mentioned above.

6. CONTAINERS FOR SIMULATION MODELS

The Control Web environment is intended to develop industrial control and visualization applications running in real-time on the Microsoft Win32 platform. Normally, Control Web communicates with the technology via the drivers of various measuring and controlling interface cards (Bílý et al 1999).

Similarly, one can use Control Web to provide a user interface to a simulation model. Only a **special driver** has to be provided, with an embedded simulation model. The input channels of the driver then carry instant values of the model's variables to the display controls on the computer screen like they were measured signals from the technology. In the other direction, the driver outputs send the user commands to the simulation model in the same way they would send them through the interface card to the controlled device (see fig. 3).

To simplify the development of the above-mentioned driver of a "virtual input/output card" containing the simulation model, so that one doesn't have to write a new driver in C++ for every model, a special utility has been developed. It **automates the creation of the driver from the Simulink model** description. Now it is possible to develop and tune the simulation model in Simulink, then convert it efficiently and quickly to a Control Web driver. Any changes to the Simulink model can be easily translated into a new driver (Kofránek et al 2002b, 2002c).

Among other undisputable advantages of Control Web is the support for distributed solutions both for intranets and the Internet. Here lies an opportunity for a distributed educational application with the numerically demanding calculation being carried out on a server while clients provide only visualization.

Another prospective environment for hosting multimedia simulators is the Microsoft .NET platform (Richter 2002). Simulation models can communicate with the objects of the user interface created with Microsoft Visual Studio .NET. We have also developed a **tool to automate the conversion of a Simulink model to the .NET environment**. A Control Web simulator can be placed in the .NET environment too and it can easily communicate with other .NET components. There are also new possibilities for e-learning web applications brought by the ASP .NET technology, which is a part of the new environment from Microsoft (Otey and Griffey 2002).

7. CONTAINERS FOR MODEL-CONTROLLED INTERACIVE ANIMATIONS

It is advantageous to use interactive animated pictures, connected to the simulation model, for the visualization of simulation games.

We have exploited Macromedia Flash for these animations. Flash offers the creation of interactive, scripted components that can be controlled from our simulators. Flash objects can be viewed directly from web pages or can be incorporated in any application supporting the ActiveX interface (Armstrong 1996).

Both graphical and programmer's interfaces (Action Script) contribute to the success of Flash (Davis 2002, Reinhardt and Down 2002).

Our development team opened a specialized laboratory of interactive graphics at a graphical school and invested a lot of effort in teaching the professional graphic designers to work with Macromedia Flash.

Interactive graphics created with Flash are used to visualize the behaviour of simulation models. Animations are controlled based on the model's outputs, e.g. a schematic picture of a vein can dilate or constrict, an alveolus can "breathe" deeply or more shallow etc.

Components with interactive graphics can be placed into Control Web's ActiveX container as well as into the Microsoft .NET environment. In multimedia educational applications, these interactive graphic components are connected with the simulation model and other components of the user interface.

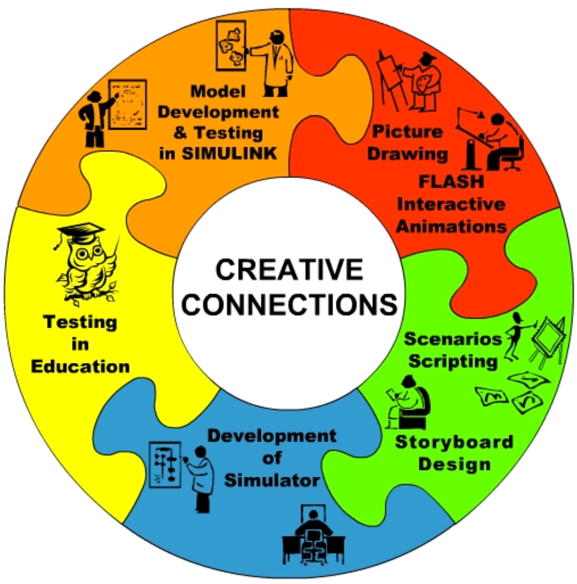

Fig. 4. Educational software development requires various professions.

8. CREATIVE CONNECTION

Creation of modern educational software represents a challenging and complicated project, requiring team cooperation of various professionals (see fig. 4):

- ***Skilled teacher*** - prepares the scenario (including the basic design of pictures and interactive animations) and tests the final products as a teaching aid
- ***System analyst*** – an expert that designs, formalizes and tunes the simulation models, in cooperation with a physiologist. The means of mutual communication are simulation chips of Matlab/Simulink environment.
- ***Graphic designer*** - designs and constructs graphic components for interactive animations in Macromedia Flash/Director
- ***Programmer*** - utilizes Control Web or Microsoft .NET as a container for the simulation model (in Control Web via a driver of a virtual input/output card, in Microsoft .NET as a special assembly) and connects it with the interactive animations and other multimedia features and programs the actual educational application
- And last but not least – the **student,** for whom the whole product is intended and whose comments after testing the programme are of high interest to the teacher and the developers

To keep the whole interdisciplinary design cycle fast and efficient, it is necessary to use specialized development tools with sufficient technical support at every stage of the design. Only such tools allow us to tie the respective parts of the simulator together according to the given scenario, forming a compact

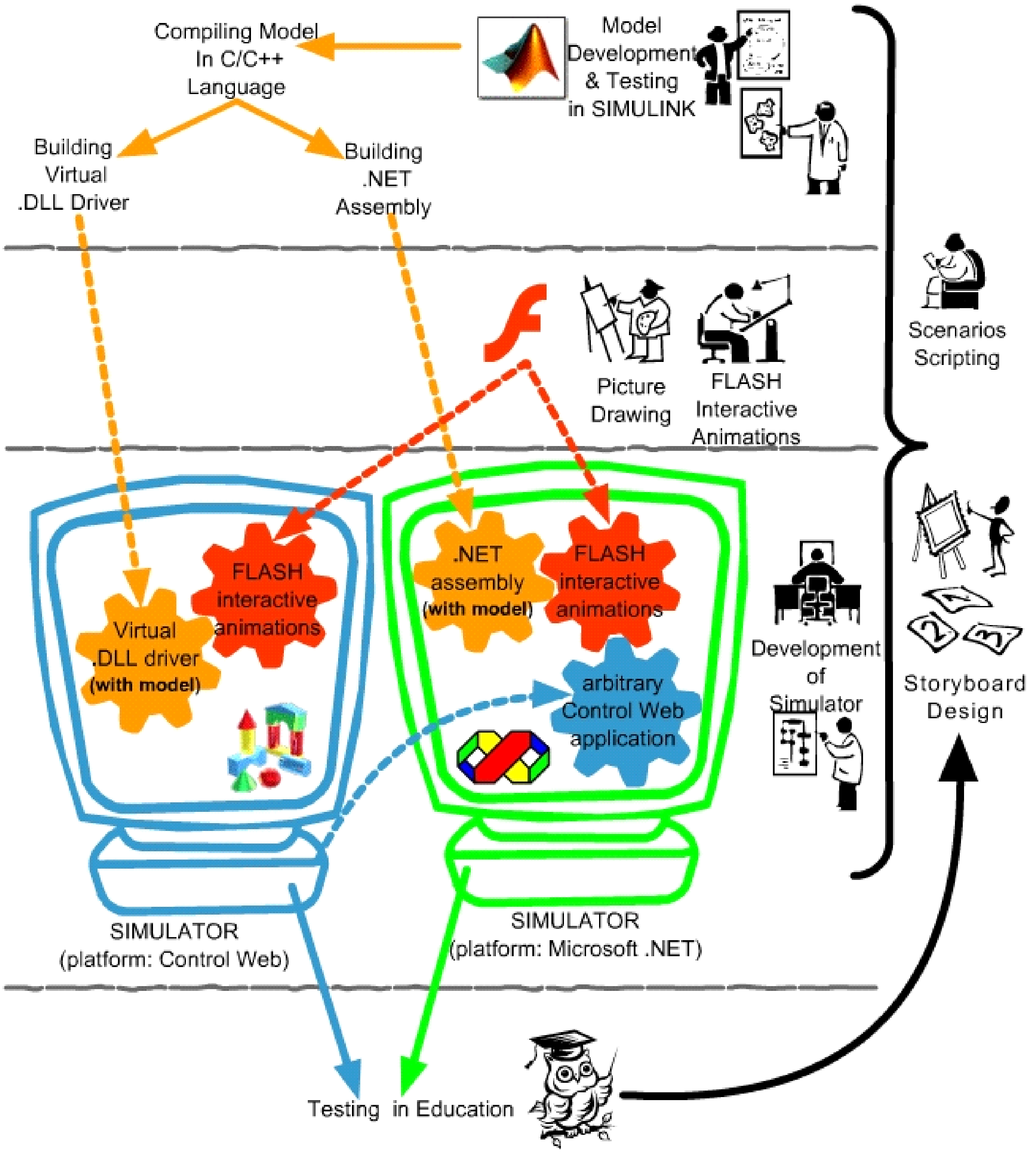

Fig. 5. In the development cycle every specialist needs specialized development tools with sufficient technical support. This diagram shows the design cycle of an educational simulator and shows how our team works at the same time.

union. To our knowledge and in our experience, the call for reliable support from the software producer is crucial, that's why *we prefer proven commercial development tools to "open source" tools of the academic domain.*

We use Matlab/Simulink for the generation of simulation models, Flash for multimedia animations and Control Web or Visual Studio .NET for integration of simulators, interactive animations, texts and other components together.

9. FROM ART TO INDUSTRY

The time of only small groups of educational software enthusiasts is coming to a close. The authoring of multimedia educational programmes is getting closer to an industrial procedure.

This doesn't mean that the invention and teaching experience of the scenario creators or the creativity of a graphic designer or the skills of a programmer are becoming less important. Working with many clever intercommunicating tools can further promote the authors' creativity and productivity (see fig. 5).

10. REFERENCES

Armstrong, T. (1996), *Designing and Using ActiveX® Controls,* Hungry Minds, Inc., Book and CD ROM Edition ISBN 1558515038.

Bílý, R., P. Cagaš, R. Cagaš, D. Hladůvka, M. Kolařík, J. Sobotík, M. Zálešák, Z. Zgarba (1999) *Control Web 2000 - Object Oriented Industrial Information and Control System.* (in Czech, English on-line version see www.mii.cz). Computer Press, Prague. ISBN 8072262520.

Davis, J. (2002). *Flash to the Core.* New Riders Publishing; 1st edition, ISBN 0735712883.

Hanselman D.C. and B. Littlefield (2001). *Mastering MATLAB 6: A Comprehensive Tutorial and Reference*, Prentice Hall, New York. ISBN 0-13-019468-9.

Kofránek, J., T. Velan and P. Janicadis. (2000). "Computer Simulator of Body Fluids and Acid-Base Disorders as an Efficient Teleeducation Tool." In: *Modelling and Control Biomedical Systems 2000. IFAC Symposium*. (E. Carson, E. Salzeider (Eds.)). Pergamon, London 2000, 233-242.

Kofránek, J., H., Snášelová, Lu Danh Vu, P. Janicadis, T. Velan (2001a): Virtual patients behind the screen. ESM 2001, *European simulation multiconference,* Ed.: E.J.H.Kerckhoffs and M.Šnorek, Prague 2001, pp. - 1000-1008.

Kofránek J., Z. Wúnsch, H., Snášelová, Lu Danh Vu, Š. Svačina (2001b) Multimedia simulation games in medical education. In: *ESM 2001, European simulation multiconference*, Ed.: E.J.H.Kerckhoffs and M.Šnorek, Prague 2001.995-999.

Kofránek, J., T. Velan, P. Janicadis, R. Kerekeš, (2001c): Diagnostic and treatment of virtual patients with Golem – multimedia simulator of physiological functions. In: *Simulation in the Health and Medical Sciences 2001*. (editors: J. G. Anderson, M. Katzer), Phoenix, Arizona 2001. ISBN: 1-56555-222-9, Society of Computer Simulation International, Phoenix 2001, pp. 157- 164.

Kofránek, M.Andrlík, T. Kripner, J. Mašek, T. Velan (2002a) "Simulation chips for GOLEM – multimedia simulator of physiological functions." In: *Simulation in the Health and Medical Sciences 2002.* (James G. Anderson, M. Katzper (Eds.)). Society for Computer Simulation International, Simulation Councils, San Diego, 2002, pp. 159-163.

Kofránek Jiří, Michal Andrlík, Tomáš Kripner, Jan Mašek. (2002b) 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

Kofránek, J., Andrlík, M., Kripner, T.,Mašek, M. (2002c): From art to industry in design of biomedical simulators. Experience of the Golem simulator project. In: *SCI 23002, The 6th World Multiconference on Systemics, Cybernetics and Informatics. Orlando, Florida, USA. Proceedings. Volume XIII. Concepts and Applications of Systemics, Cybernetics and Informatics III.* (N. Callaos, G. Whymark, W. Lesso (Eds)). Internatrional Institute of Informatics and Systemics, Orlando, Florida, USA, 2002, pp. 249-256. ISBN 980-07-8150-1.

Otey, M. and W. Griffey (2002) *ADO.NET: The Complete Reference.* Osborne/McGraw-Hill, , ISBN 0072228989.

Reinhardt, R. and S. Dowd (2002). *Flash MX Bible*. John Wiley & Sons; Book and CD-ROM edition, ISBN 0764536567.

Richter, J. (2002). *Applied Microsoft .NET Framework Programming.* Microsoft Press. 1 edition. ISBN 0735614229.

Tewari, A. (2002): *Modern Control Design With MATLAB and SIMULINK* , John Wiley & Sons, ISBN 0471496790.

ACKNOWLEDGEMENT

This research was partially supported by the Research and Development Initiative MSM 111100008 (physiome.cz) and by BAJT servis Ltd.