

From Simulation Chips to Biomedical Simulators

Jiří Kofránek, Michal Andrlík, Tomáš Kripner, Jan Mašek

Department of Pathological Physiology, 1st Faculty of Medicine, Charles University, Prague, Czech Republic, e-mail:kofranek@cesnet.cz

Abstract

Computer simulation in physiology is an interdisciplinary problem. It is very important to provide good readability of used models. Programmer understands the architecture of models and the algorithms of programs. On the other hand physiologist usually can judge how closely the behavior of system correspond to the nature. In this text we will show how some known programming methods can help us find common language between programmers and physicians.

Simulation models

Simulation models in physiology are based on differential equations, regulation mechanisms and often on some mathematical algorithms like finding value of inverse function or solving some optimization problem.

In scientific literature there are many articles about partial simulation models programmed in classical sequential languages like FORTRAN, MODULA, ADA, etc. Those products are hardly understandable for physiologists and non-programmers. Speak nothing of linear structure of these programs, which usually excludes any updating, adaptations or expansion.

For these reasons our team found another way. We use **MATLAB & SIMULINK** graphical programming environment. This software is developed for use in control industry and for using advanced mathematical methods and has a strong toolset for constructing, developing and testing of various models.

Simulation chips

Step of high importance is creating of **simulation chips**. The idea borrows from the concept of electronic chips and electronic circuits where the circuitry lies hidden inside the packages. Simulation chip is a subsystem with exactly defined function. Mask of this subsystem contains brief description of function and meaning, and description of inputs and outputs.

This is actually a method of decomposition of a model to the parts.

It is clear that physiologists can look at this chip like a black box with some physiological function and programmers look at chip like a subprogram or algorithm that is part of a more complex system.

Example of one very simple chip can be seen on *figure 1*.

Simulation chip is a basic construction element. As physiological systems (and models too) may have several levels, it is natural that chips can be ordered hierarchically. It means,

system with several chips inside can be masked and can be represented as one “bigger” simulation chip. So construction of chips fully traces hierarchical organization of nature.

Example of hierarchical structure of chips can be seen on *figure 2*.

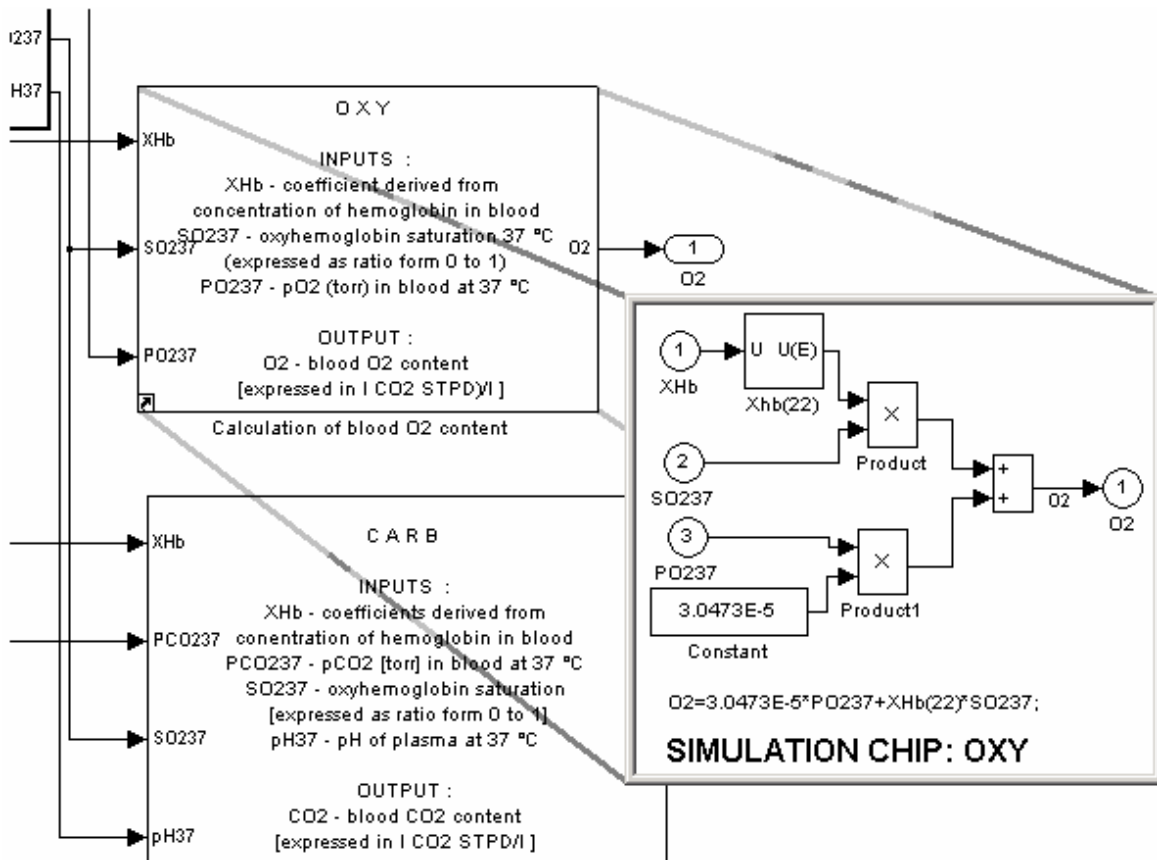


Figure 1: The simple example of simulation chip.

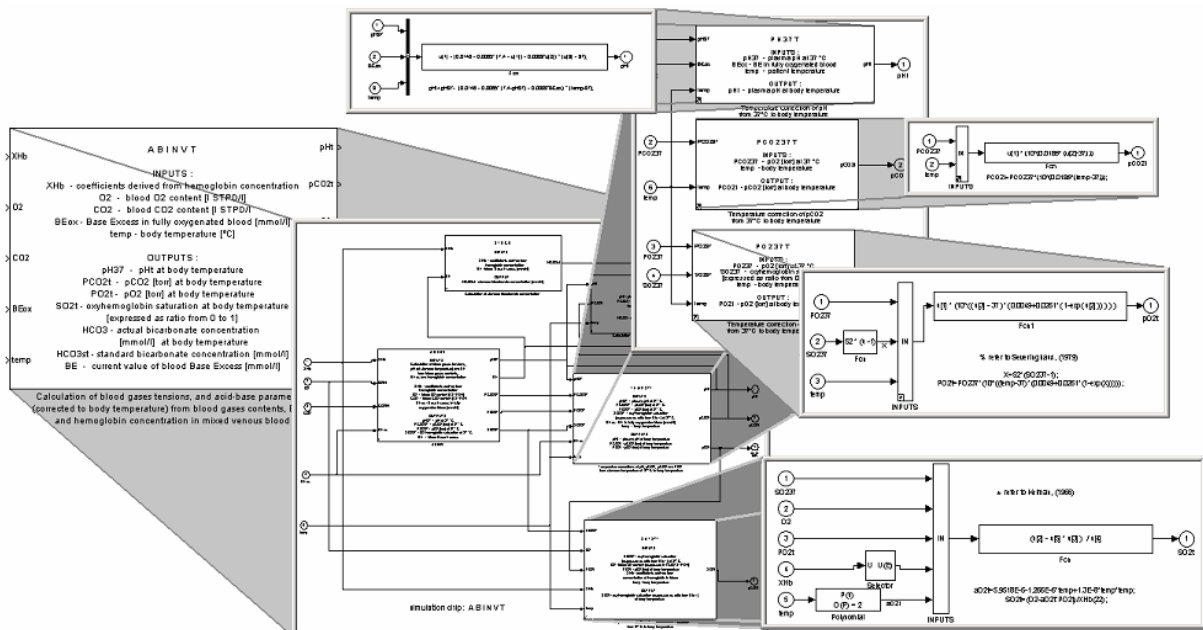


Figure 2: The sketch of hierarchical structure of chips.

Chips library

We have many different models that sometimes contain identical parts. Furthermore one system or chip can internally include the same chip in several places and contexts. From whence it follows that one simulation chip can be copied into many places in many different models. When we want to update the content of the chip, we must replace chip in all of its places (and it is very probable to forget about some of them).

Next, if we update or modify a chip, we don't know if somewhere else there exists a copy of this chip. Very simply it may happen that we will have several different versions of a subsystem in several places. It very often leads to disorder.

Therefore our team applied the idea of a large library. It is possible to create libraries of blocks (in our case libraries of chips) in SIMULINK. A model then contains no chips but just links to chips into a library. All the same, user can open chips and pass through the model without any limitations.

Let's now consider several models containing links to the same chip in a library. When we update or replace the chip in the library, then by opening or executing "Model Update" command model automatically reloads all links from libraries. So we have the latest versions of chips in a model all the time. Certainly a block in a library can contain a link to another chip, which is in the same or another library.

On opening a library it refreshes all links inside and loads the latest version of blocks. If we need to make a mutation of a linked chip in a single system only, and we don't want to modify the chip in the library, it is possible to disable or break this link. It can be used for developing new model or developing any special modification of chip.

Example of a library is on *figure 3*.

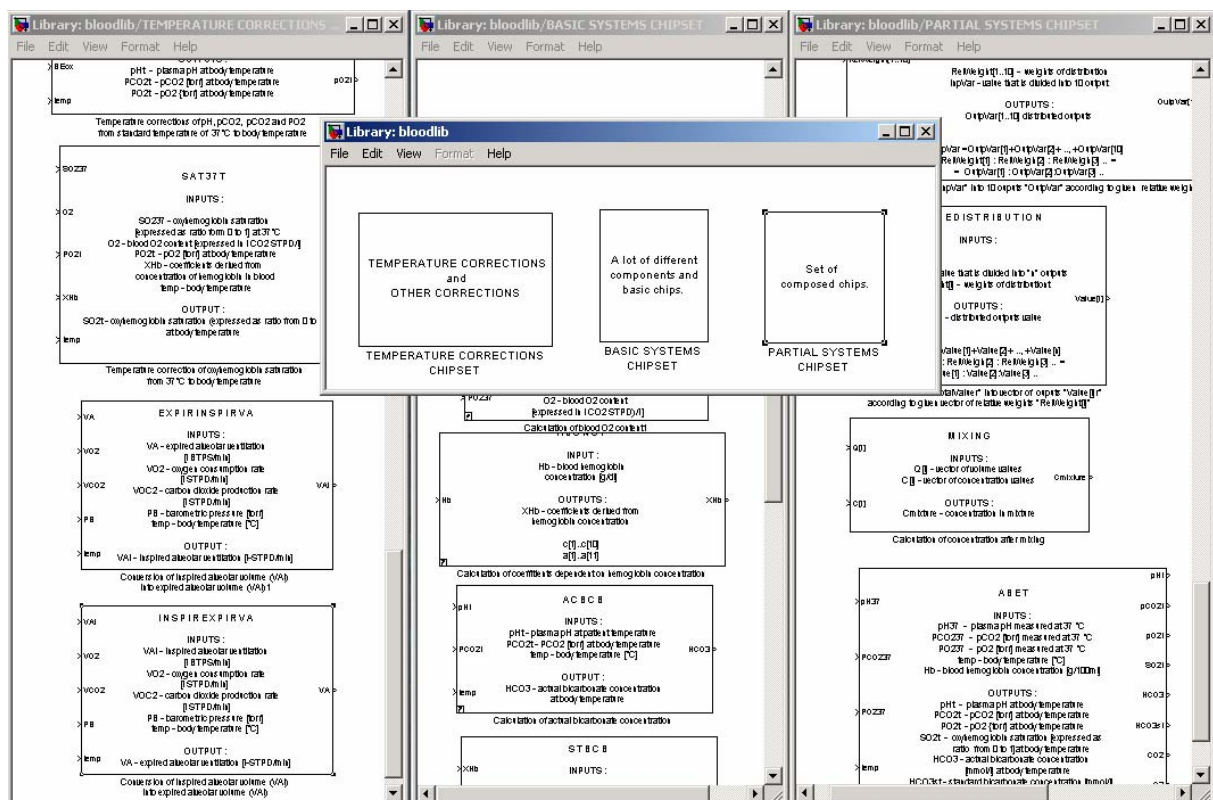


Figure 3: The library is a set of standalone chips.

Our team used this way to develop “GOLEM” simulator, for simulation of inner body environment. See more on www.physiome.cz.

Simulators

One thing is to develop a simulation model and second thing is to build an interface between the model and a user. We must consider users who don't know how to work in SIMULINK or don't want to buy all MATLAB & SIMULINK products. Or the user has no interest in the structure of the model and chips but he needs to comfortably monitor the behavior of this model.

These observations lead our team to conclusion that it is necessary to be able to run the model outside the SIMULINK. After a certain time of searching, we have chosen the Control Web developer's environment from Moravian Instruments (see <http://www.mii.cz/>).

On one hand, we need to create a comprehensive graphical interface (with the integration of interactive animations, charts etc.). On the other hand, the structural complexity of the simulation model (which is the heart of the simulator) has high demands on the numerical performance. The requirements for sophisticated graphics and the high execution speed of the simulation model are contradictory – both are demanding on the processor performance.

The tools for creating industrial controlling and measuring applications can handle this problem – these applications maintain real time computational throughput, while servicing input / output requests and updating graphics.

A little trick had to be used when incorporating the SIMULINK model into the core of the simulator. Instead of the measuring/controlling adapter driver by which Control Web communicates with a controlled device in an industrial application, we have programmed a virtual driver for a (nonexistent) measuring/controlling adapter. This virtual device driver mimics the behavior of the simulation model. The Control Web application now “thinks” that it sends some control signals through the driver to an industrial device, but in reality the signals are inputs to the simulation model. Similarly, instead of measured output signals from the periphery, the Control Web application registers the values of the output variables of the model (See *figure 4*). This enabled us to employ the abundant options of the Control Web developer's environment for the generation of our own simulator.

The generation of the simulator is also facilitated by the capabilities of a special MATLAB toolbox - Real Time Workshop (RTW). RTW is intended for development of real-time systems running on many software and hardware platforms. As an intermediate step, RTW translates a SIMULINK model into C-language.

Through customization of the C source code, different targets are supported, depending on the programmer's needs. An important achievement of our team is a creation of a new RTWtarget, Control Web driver target, which serves the purpose of making a Control Web virtual device driver with an embedded SIMULINK model.

Gradual generation of the simulator is sketched in *figure 5*.

Simulation chips – standardized communication tool

In through all the development stages of the simulator the simulation chips proved to be very useful. They promote structured design of a model, serve as an up-to-date documentation of the model and succeeded as a standardized communication tool between technicians and physiologists.

From simulation chips to biomedical simulator

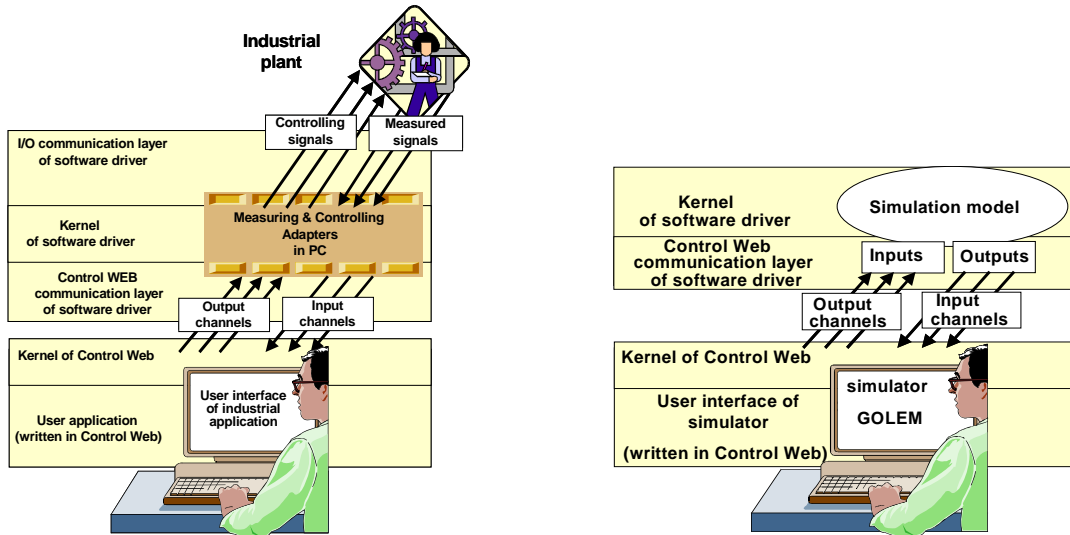


Figure 4: Communication of the system Control Web with driver measuring/controlling adapters during creation of industrial applications and inserting of simulating model to the driver of the “virtual card” during creation of simulator Golem in the system Control Web.

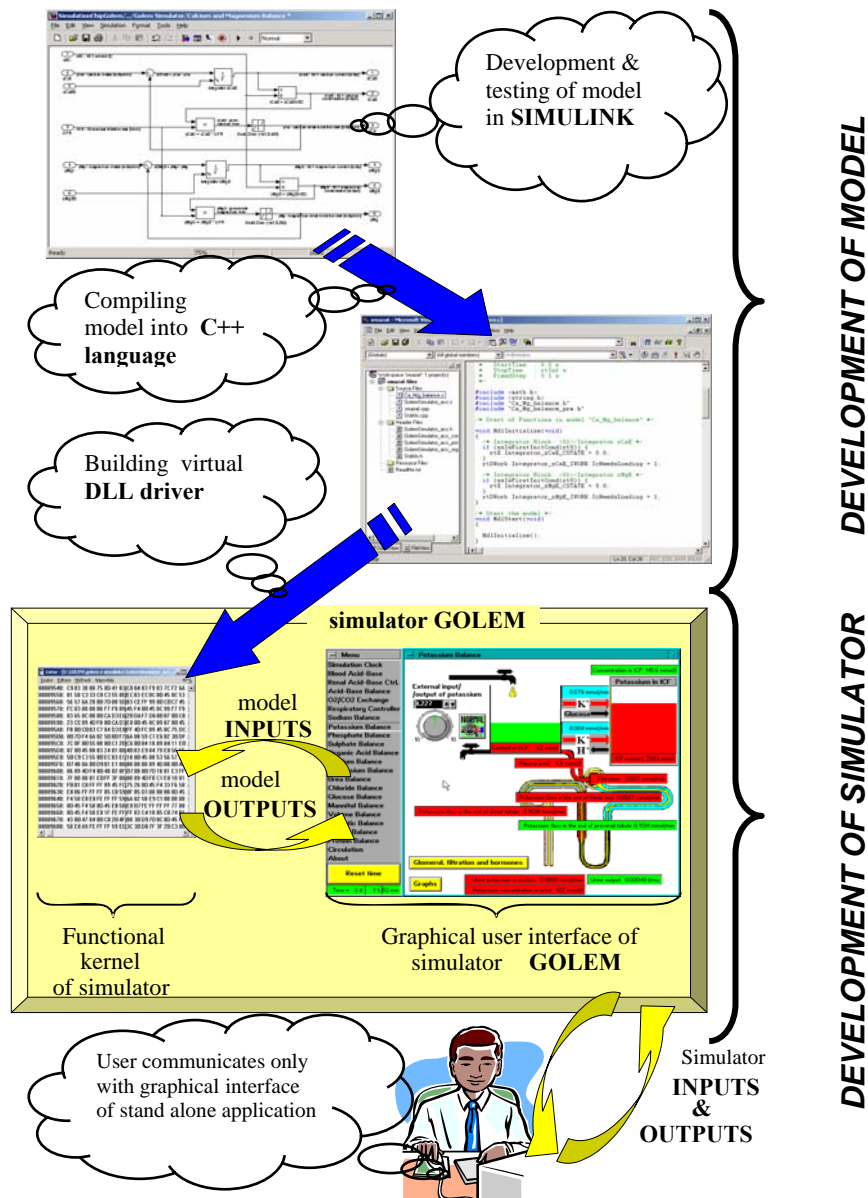


Figure 5: Gradual generation of the simulator

References

1. Current structure of Golem simulator: <http://www.physiome.cz>
2. Kofránek, J., T. Velan and R. Kerekeš. 1997. "Golem: A Computer Simulator of Physiological Functions as an Efficient Teaching Tool." In: Legacy for 21st Century. Proceedings of the First World Congress on Systems Simulation., (Teo, Z.M., W.C.Wong., T.I.Oren, R. Rimane (Eds.)), IEEE Singapore Section, Singapore., Singapore, 407-411.
3. 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.
4. Kofránek, J., T. Velan, P. Janicadis and R. Kerekeš. 2000. "Golem – medical multimedia training simulator of body fluids and acid-base disorders." In: The Proceedings of 2000 Summer Computer Simulation Conference. (William Waite, Abe Nisanci.(Eds.)) Society for Computer Simulation International. Vancouver 2000, pp. 737-741.
5. Kofránek, J., T. Velan, P. Janicadis and R. Kerekeš. 2001. "Diagnostic and treatment of virtual patients with Golem – multimedia simulator of physiological functions." In: Simulation in the Health and Medical Sciences 2001. (James G. Anderson, M. Katzper (Eds.)). Society for Computer Simulation International, Simulation Councils, San Diego, 2001, pp. 157-164.
6. Kofránek, J., L. D. Anh Vu, H. Snášelová, R. Kerekeš and T. Velan. 2001. "GOLEM – Multimedia simulator for medical education." In: Studies in Health Technology and Informatics., vol. 84. MEDINFO 2001, Proceedings of the 10th World Congress on Medical Informatics. (V.L. Patel, R. Rogers, R. Haux (Eds.)), IOS Press, Amsterdam, Berlin, Oxford, Washington DC, 2001, pp. 1042-1046.
7. Kofránek J., H., Snášelová, Lu Danh Vu, P. Janicadis, T. Velan: Virtual patients behind the screen. ESM 2001, European simulation multiconference, Ed.: E.J.H.Kerckhoffs and M.Šnorek, Prague 2001, pp. - 1000-1008.
8. Kofránek, J., M.Andrlík, T. Kripner, J. Mašek, T. Velan: 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