HumMod - Large Scale Physiological Models in Modelica

Jiri Kofranek Marek Matejak Pavol Privitzer
Institute of Pathophysiology, First Faculty of Medicine, Charles University
U nemocnice 5, 128 53 Praha 2, Czech Republic
kofranek@gmail.com matejak.marek@gmail.com pavol.privitzer@lf1.cuni.cz

Abstract

Modelica is being used more and more in industrial applications, but Modelica is still not used as much in biomedical applications. For a long time we have mostly been using Matlab/Simulink models, made by Mathworks, for the development of models of physiological systems. Recently, we have been using a simulation environment based on the Modelica language. In this language, we implemented a large scale model of interconnected physiological subsystems containing thousands of variables. Model is a richly hierarchically structured, easily modifiable, and “self-documenting”. Modelica allows a much clearer than other simulation environments, to express the physiological nature of the modeled reality.

Keywords: simulation; physiology; large-scale model

1 Introduction

It is simply amazing how fast the Modelica simulation language adopted various commercial development environments. Modelica is being used more and more in industrial applications. However, Modelica is still not used as much in biomedical applications.

The vast majority of biomedical simulation applications are still done in casual, block-oriented environments. These include referencing database development environments for biomedical models (such as the JSIM language - http://physiome.org/model/doku.php or CellML language - http://www.cellml.org/).

A frequently used environment in biology and medicine is Matlab/Simulink – monographs dedicated to biomedicine models are usually equipped with additional software used in this environment, but so far without the use of new acasual or non-casual Simulink libraries, such as [24, 28, 32].

However, already in 2006, Cellier and Nebot [5] pointed out the benefits of Modelica, when used for clear implementation of physiological systems descriptions and interpretations. The classic McLeod‘s circulation system model was implemented by PHYSBE (PHYSiological Simulation Benchmark Experiment) [25, 26, 27]. The difference is clearly seen, if we compare the Cellier model implementation [5] with the freely downloadable version of the PHYSBE model implementation in Simulink http://www.mathworks.com/products/demos/simulink/physbe/.

Haas and Burnhan, in their recently published monograph, pointed out the benefits and large potential of the Modelica language used for modeling medically adaptive regulatory systems [9]. The most recent, Brugård [4] talks about work on the implementation of the SBML language (http://sbml.org/) in the Modelica language. This would enable us in the future, to simply run models, whose structure is described in the SBML language, on development platforms, based on the Modelica language.

2 Web of physiological regulations

Thirty-nine years ago, in 1972 Guyton, Coleman and Granger published an article in the Annual Review of Physiology [9] which at a glance was entirely different from the usual physiological articles of that time. It was introduced by a large diagram on an insertion. Full of lines and interconnected elements, the drawing vaguely resembled an electrical wiring diagram at first sight (Fig. 1). However, instead of vacuum tubes or other electrical components, it showed interconnected computation blocks (multipliers, dividers, adders, integrators and functional blocks) that symbolized mathematical operations performed on physiological variables. In this entirely new manner, using graphically represented mathematical symbols; the authors described the physiological regulations of the circulatory system and its broader physiological relations and links with the other subsystems in the body – the kidneys, volumetric and electrolyte balance control, etc. Instead of an extensive set of mathematical equations, the article used a graphical representation of mathematical relations. This syntax allowed depicting relations between individual physiological variables graphically in the form of interconnected blocks representing mathematical operations. The whole dia-

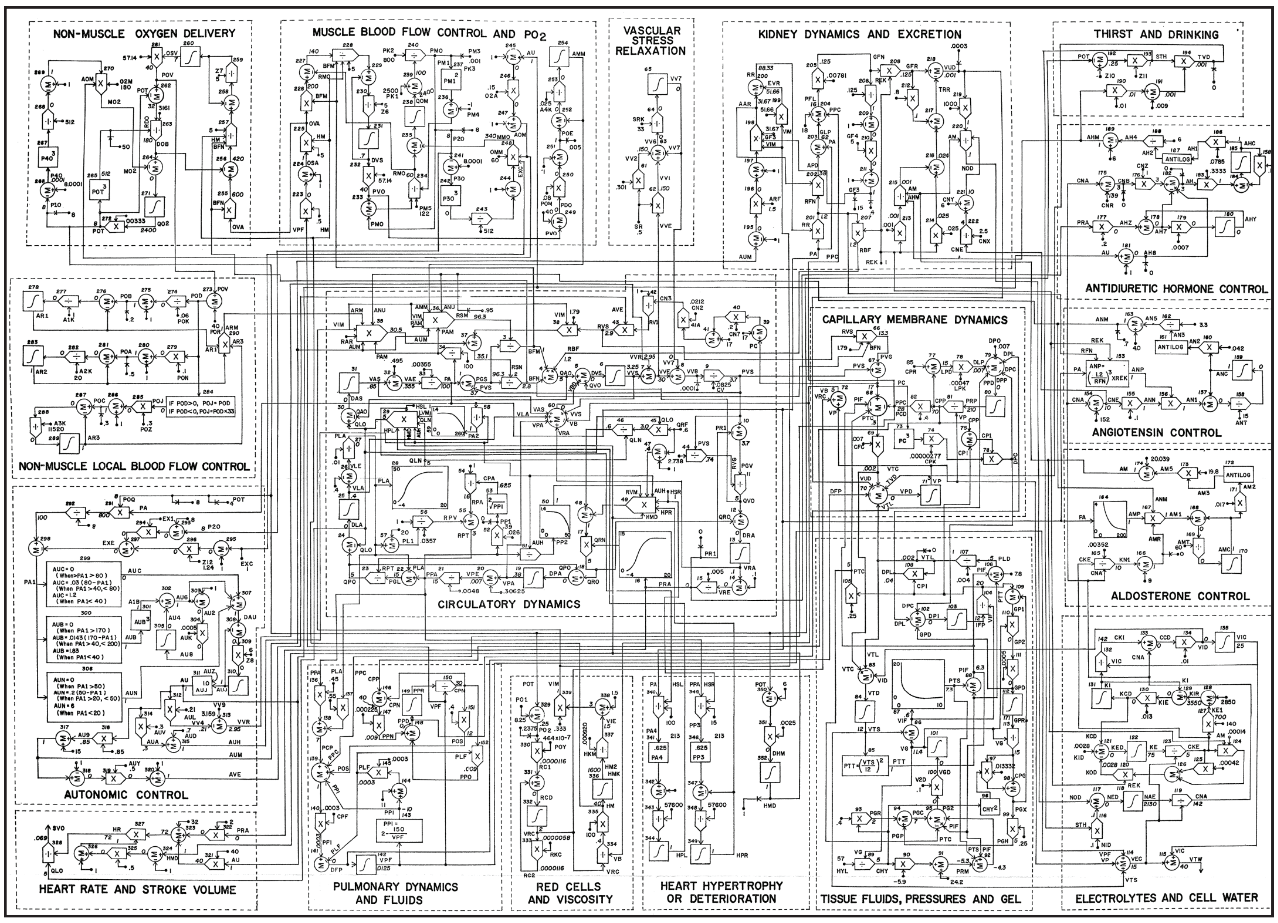

NON-MUSCLE OXYGEN DELIVERY
MUSCLE BLOOD FLOW CONTROL AND PO2
VASCULAR STRESS RELAXATION
KIDNEY DYNAMICS AND EXCRETION
THIRST AND DRINKING
ANTIDIURETIC HORMONE CONTROL
NON-MUSCLE LOCAL BLOOD FLOW CONTROL
CAPILLARY MEMBRANE DYNAMICS
ANGIOTENSIN CONTROL
CIRCULATORY DYNAMICS
ALDOSTERONE CONTROL
AUTONOMIC CONTROL
HEART RATE AND STROKE VOLUME
PULMONARY DYNAMICS AND FLUIDS
RED CELLS AND VISCOSITY
HEART HYPERTROPHY OR DETERIORATION
TISSUE FLUIDS, PRESSURES AND GEL
ELECTROLYTES AND CELL WATER

Figure 1: Guyton's blood circulation regulation diagram from 1972.

gram thus featured a formalized description of physiological relations in the circulatory system using a graphically represented mathematical model.

Guyton's model was the first extensive mathematical description of the physiological functions of interconnected body subsystems and launched the field of physiological research that is sometimes described as "integrative physiology" today. Just as theoretical physics tries to describe physical reality and explain the results of experimental research using formal means, "integrative physiology" strives to create a formalized description of the interconnection of physiological controls based on experimental results and explain their function in the development of various diseases.

From this point of view, Guyton's model was a milestone, trying to adopt a systematic view of physiological controls to capture the dynamics of relations between the regulation of the circulation, kidneys, the respiration and the volume and ionic composition of body fluids by means of a graphically represented network.

Guyton's graphical notation was soon adopted by other authors – such as Ikeda et al. (1979) in Japan [13]

and Amosov et al. (1977) in the former USSR [2]. However, the graphical notation of the mathematical model using a network of interconnected blocks was only a graphical representation – Guyton's model and later modifications (as well as the models of other authors that adopted Guyton's representative notation) were originally implemented in Fortran and later in C++.

Today the situation is different.

Now, there are specialized software simulation environments available for the development, debugging and verification of simulation models, which allow creating a model in graphical form and then testing its behavior. One of these is the Matlab/Simulink development environment by Mathworks, which allows building a simulation model gradually from individual components – types of software simulation elements that are interconnected using a computer mouse to form simulation networks.

Simulink blocks are very similar to the elements used by Guyton for the formalized representation of

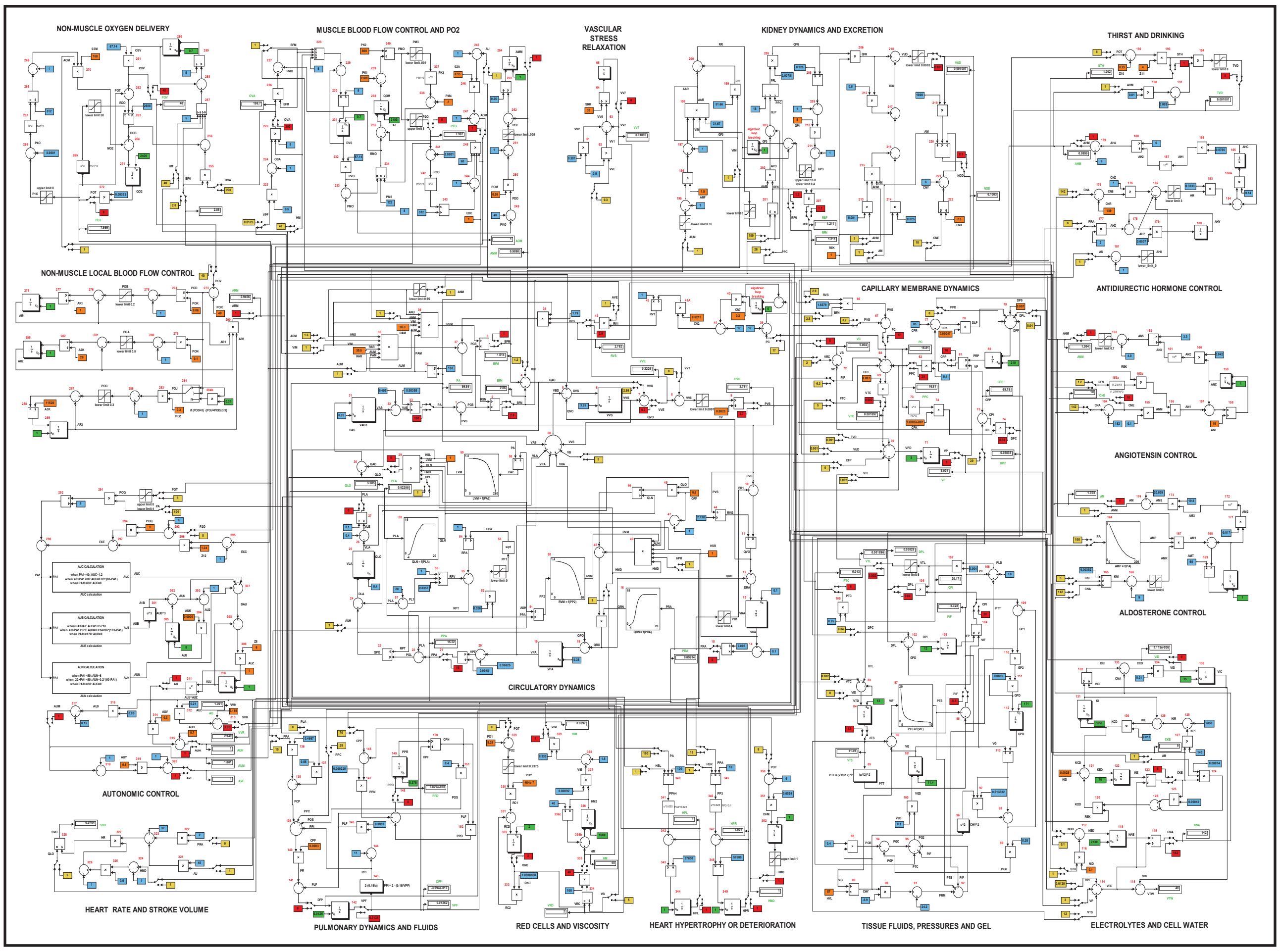

NON-MUSCLE OXYGEN DELIVERY
MUSCLE BLOOD FLOW CONTROL AND PO2
VASCULAR STRESS RELAXATION
KIDNEY DYNAMICS AND EXCRETION
THIRST AND DRINKING
NON-MUSCLE LOCAL BLOOD FLOW CONTROL
CAPILLARY MEMBRANE DYNAMICS
ANTIDIURECTIC HORMONE CONTROL
ANGIOTENSIN CONTROL
ALDOSTERONE CONTROL
CIRCULATORY DYNAMICS
AUTONOMIC CONTROL
HEART RATE AND STROKE VOLUME
PULMONARY DYNAMICS AND FLUIDS
RED CELLS AND VISCOSITY
HEART HYPERTROPHY OR DETERIORATION
TISSUE FLUIDS, PRESSURES AND GEL
ELECTROLYTES AND CELL WATER

Figure 2: The implementation of Guyton's model in Simulink preserves the original arrangement of elements in Guyton's graphic diagram.

physiological relations. The only difference is in their graphical form. This similarity inspired us to use Simulink to revive Guyton's good, classic diagram and transform it into a working simulation model. When implementing the model in Simulink, we used switches that allow us to connect and disconnect individual subsystems and control loops while the model is running. We strove to keep the appearance of the Simulink model identical to the original graphic diagram – the arrangement, wire location, variable names and block numbers are the same.

The simulation visualization of the old diagram was not without difficulties – there are errors in the original graphic diagram of the model! It does not matter in the hand-drawn illustration but if we try to bring it to life in Simulink, the model as a whole collapses immediately. A detailed description of the errors and their corrections is in [23].

Our Simulink implementation of Guyton's (corrected) model (Figs. 2 and 3) is available for download at www.physiome.cz/guyton. Also available at that address is our Simulink implementation of a much more complex, later model from Guyton et al. There is also a very detailed description of all applied mathematical relations with an explanation.

3 Block-oriented simulation networks for physiology

Block-oriented simulation languages, of which Simulink is a typical example, allow assembling computer models from individual blocks with defined inputs and outputs. The blocks are grouped in libraries; when building a model, a computer mouse is used to create individual block instances, with inputs and outputs connected through wires that "conduct" information.

A Simulink network can be arranged hierarchically. Blocks can be grouped into subsystems that communicate with their ambient environment through defined input and output "pins", making "simulation chips" of a sort. A simulation chip hides the simulation network structure from the user, much like an electronic chip hiding the interconnection of transistors and other electronic elements. Then the user can be concerned

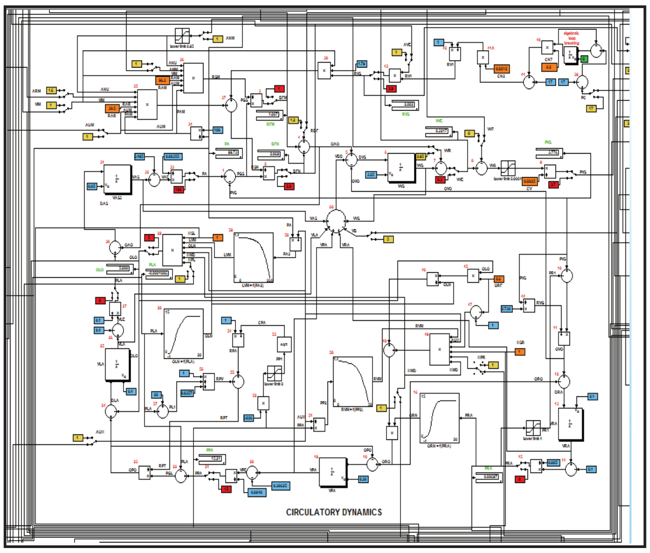


Figure 3: Circulatory dymamics - more detailed central structures of the Simulink implementation of Guyton's model, representing flows through aggregated parts of the circulatory system and the activity of the heart as a pump.

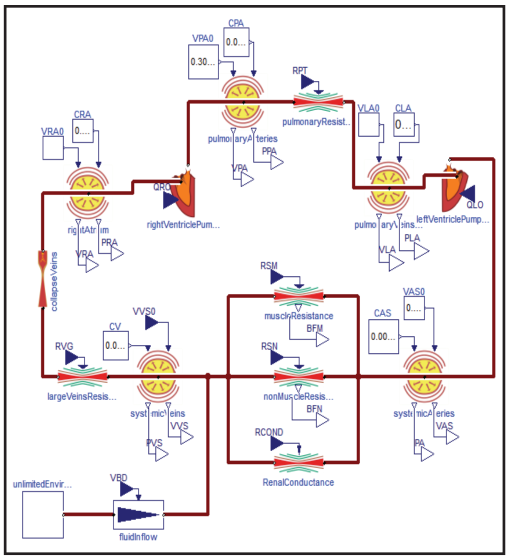


Figure 4: The same model structure as is shown in figure 3 implemented in Modelica. The structure of the model in Simulink corresponds to the structure of computational steps, while the Structure of Modelica model reflects the structure of the modeled physiological reality.

just with the behavior of the chip and does not have to bother about the internal structure and calculation algorithm.

The behavior of a simulation chip can be tested by monitoring its outputs using virtual displays or virtual oscilloscopes connected to it. This is very useful especially for testing the behaviour of a model and expressing the mutual relations of variables.

Simulation chips can be stored in libraries and users can create their instances for use in their models. For example, we created a Physiolibrary for modelling physiological regulations.

Hierarchical, block-oriented simulation tools are thus used advantageously in the description of the complex regulation systems that we have in physiology.

A formalized description of physiological systems is the subject matter of PHYSIOME, an international project that is a successor to the GENOME project. The output of the GENOME project was a detailed description of the human genome; the goal of the PHYSIOME project is a formalized description of physiological functions. It uses computer models as its methodological tool [3, 12].

Several block-oriented simulation tools developed under the PHYSIOME project have been used as a reference database for a formalized description of the structure of complex physiological models. These include JSIM (http://www.physiome.org/model/doku.php) and CellML (http://www.cellml.org).

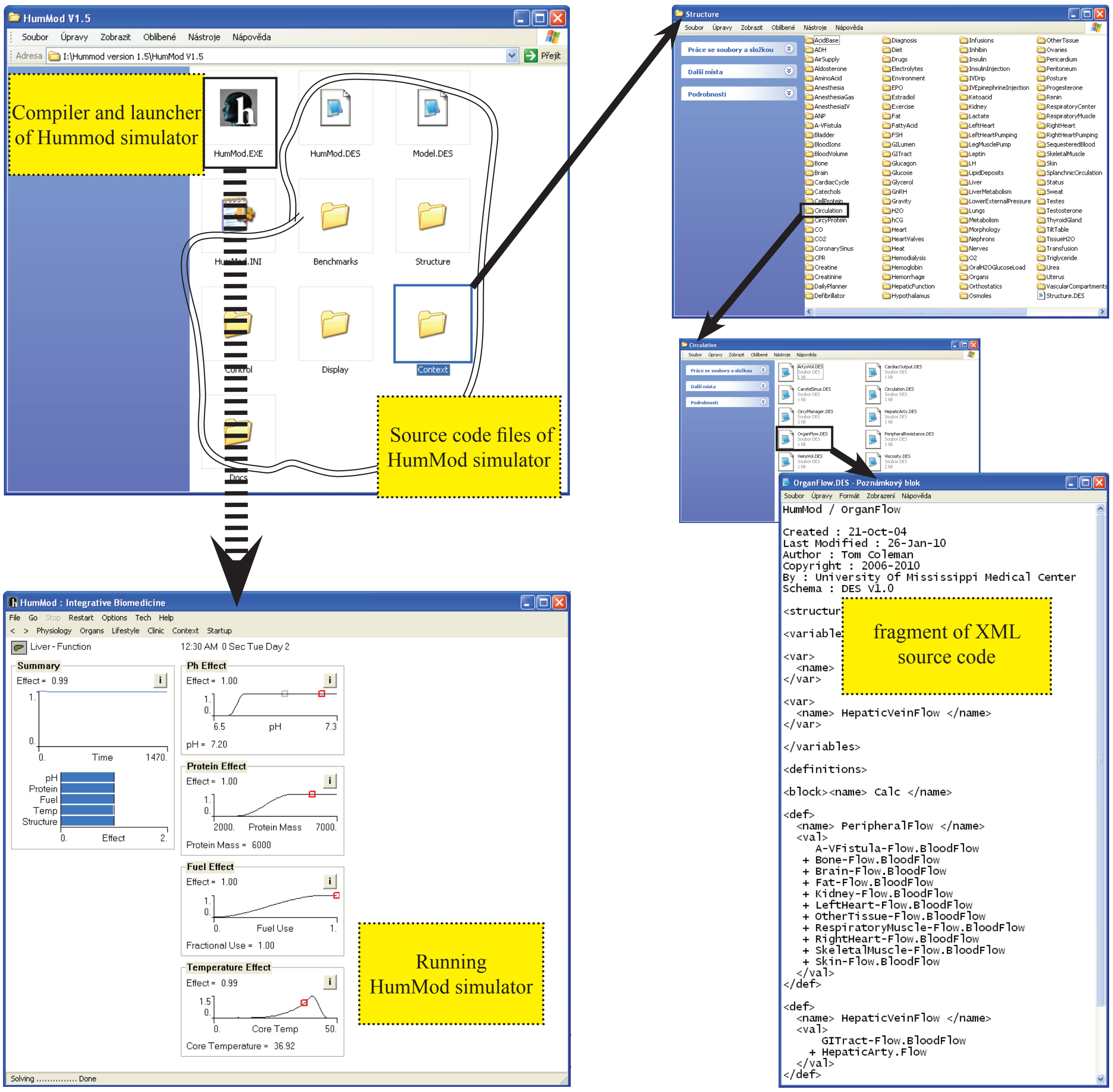
4 From Simulink to Modelica in modeling of large-scale physiological systems

We have been using Matlab and Simulink for years to create and develop models of physiological systems [16, 17, 23] and have also been developing the relevant application Simulink library – the Physiolibrary (http://www.physiome.cz/simchips).

We have also developed the relevant software tools that simplify the transfer of models implemented in Simulink over to development environments (ControlWeb and Microsoft .NET), where we create tutorial and education simulators [18, 22]. Our development team gained invaluable experience in previous years working with the Matlab/Simulink development environment made by the renown company MathWorks. On the other hand, we were also attracted by the acausal development environments using the Modelica language.

In the Modelica language environment the essence of physiological regulation is much clearer than in Simulink causal network (see Figures 3 and 4). We were facing a decision whether to continue with the development process of physiological system models in Simulink (using new acasual libraries), or to make a radical decision and switch to the new Modelica language platform.

Our decision was affected by our efforts to imple-

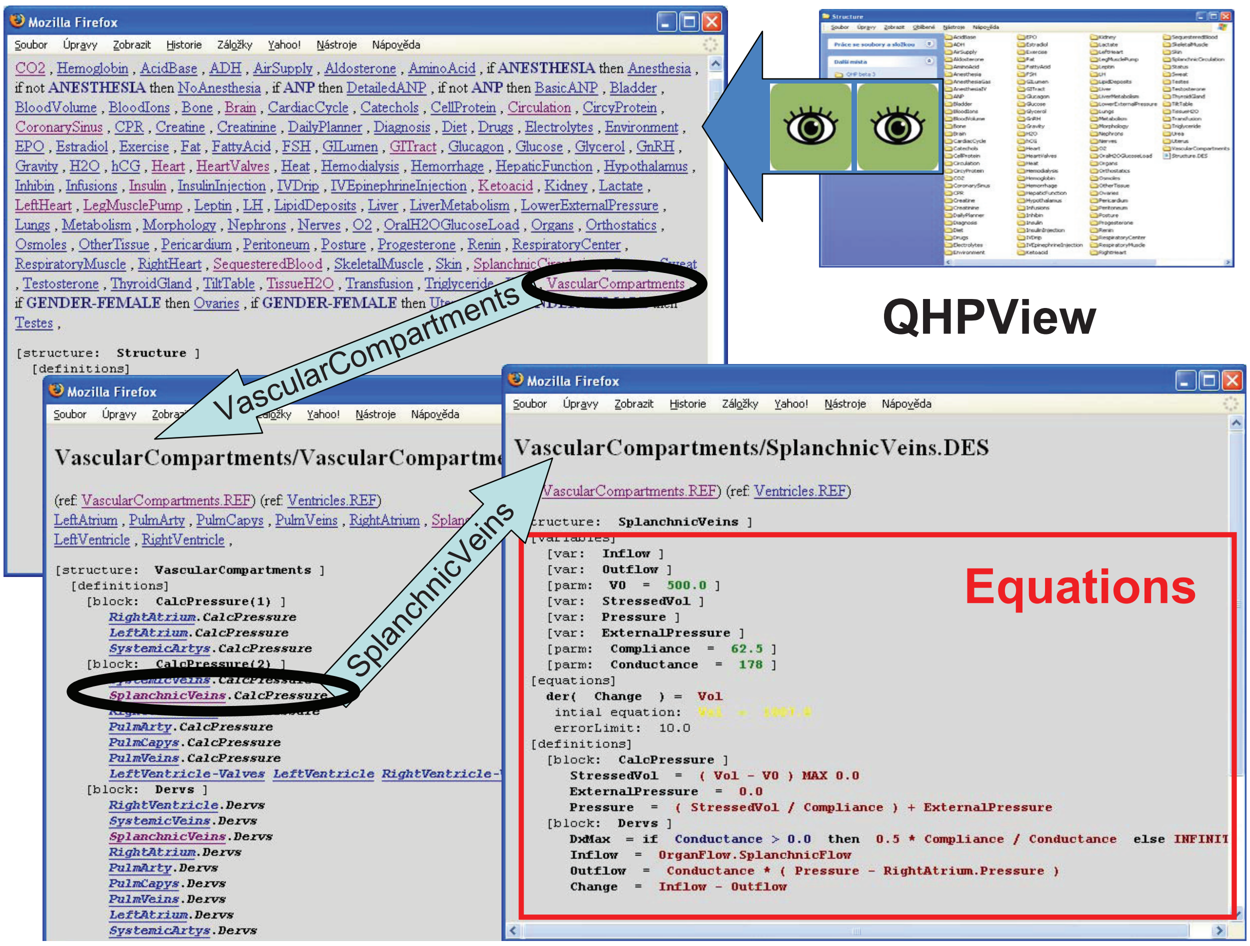


```
HumMod / OrganFlow

Created : 21-Oct-04
Last Modified : 26-Jan-10
Author : Tom Coleman
Copyright : 2006-2010
By : University Of Mississippi Medical Center
Schema : DES V1.0

<structur

<variable

<var>
  <name>
</var>

<var>
  <name> HepaticVeinFlow </name>
</var>

</variables>

<definitions>

<block><name> Calc </name>

<def>
  <name> PeripheralFlow </name>
  <val>
     A-VFistula-Flow.BloodFlow
   + Bone-Flow.BloodFlow
   + Brain-Flow.BloodFlow
   + Fat-Flow.BloodFlow
   + Kidney-Flow.BloodFlow
   + LeftHeart-Flow.BloodFlow
   + OtherTissue-Flow.BloodFlow
   + RespiratoryMuscle-Flow.BloodFlow
   + RightHeart-Flow.BloodFlow
   + SkeletalMuscle-Flow.BloodFlow
   + Skin-Flow.BloodFlow
  </val>
</def>

<def>
  <name> HepaticVeinFlow </name>
  <val>
      GITract-Flow.BloodFlow
    + HepaticArty.Flow
  </val>
</def>
```

Figure 5: All necessary files of the Quantitative Human Physiology tutorial simulator (called the HumMod by the authors in the last version). This simulator has been designed for the Windows operating system and does not require special installation. Only zip files must be unzipped into a selected folder. After you click the Hummod.exe icon, the translator translates the source text embedded within hundreds of directories and more than two thousand files and initiate its own simulator. Even though the source text of the simulator and the entire mathematical model on the background is offered as an open source (and in theory, the user may modify the model), the navigation through thousands of mathematical relations and viewing thousands of XML and interconnected files is rather difficult.

ment a large model made by Guyton's disciples and followers. Their ***Quantitative Human Physiology*** model is an extension of a tutorial simulator called the ***Quantitative Circulatory Physiology (QCP)*** [1]. ***Quantitative Human Physiology (QHP)*** simulator [10], which is now distributed as ***"HumMod"*** [11], represents today's most comprehensive and largest model of physiological functions.

The ***HumMod*** model contains more than 4000 variables and at the present time, it probably represents the largest and most extensive model of physiological regulations. It enables the user to simulate a wide range of pathological stages and statuses, including the effects of the relevant applied therapy. The authors developed a special block-oriented simulation system to represent the complex model structure. Compared



CO2 , Hemoglobin , AcidBase , ADH , AirSupply , Aldosterone , AminoAcid , if **ANESTHESIA** then Anesthesia , if not **ANESTHESIA** then NoAnesthesia , if **ANP** then DetailedANP , if not **ANP** then BasicANP , Bladder , BloodVolume , BloodIons , Bone , Brain , CardiacCycle , Catechols , CellProtein , Circulation , CircyProtein , CoronarySinus , CPR , Creatine , Creatinine , DailyPlanner , Diagnosis , Diet , Drugs , Electrolytes , Environment , EPO , Estradiol , Exercise , Fat , FattyAcid , FSH , GILumen , GITract , Glucagon , Glucose , Glycerol , GnRH , Gravity , H2O , hCG , Heart , HeartValves , Heat , Hemodialysis , Hemorrhage , HepaticFunction , Hypothalamus , Inhibin , Infusions , Insulin , InsulinInjection , IVDrip , IVEpinephrineInjection , Ketoacid , Kidney , Lactate , LeftHeart , LegMusclePump , Leptin , LH , LipidDeposits , Liver , LiverMetabolism , LowerExternalPressure , Lungs , Metabolism , Morphology , Nephrons , Nerves , O2 , OralH2OGlucoseLoad , Organs , Orthostatics , Osmoles , OtherTissue , Pericardium , Peritoneum , Posture , Progesterone , Renin , RespiratoryCenter , RespiratoryMuscle , RightHeart , SequesteredBlood , SkeletalMuscle , Skin , SplanchnicCi... Sweat , Testosterone , ThyroidGland , TiltTable , TissueH2O , Transfusion , Triglyceride , VascularCompartments , if **GENDER-FEMALE** then Ovaries , if **GENDER-FEMALE** then Ute... Testes ,

```
[structure:  Structure ]
  [definitions]
```

VascularCompartments/VascularCompartme

(ref: VascularCompartments.REF) (ref: Ventricles.REF)
LeftAtrium , PulmArty , PulmCapys , PulmVeins , RightAtrium , Splan...
LeftVentricle , RightVentricle ,

```
[structure:  VascularCompartments ]
  [definitions]
    [block:  CalcPressure(1) ]
       RightAtrium.CalcPressure
       LeftAtrium.CalcPressure
       SystemicArtys.CalcPressure
    [block:  CalcPressure(2) ]
       SystemicVeins.CalcPressure
       SplanchnicVeins.CalcPressure
       RightAtrium.CalcPressure
       PulmArty.CalcPressure
       PulmCapys.CalcPressure
       PulmVeins.CalcPressure
       LeftVentricle-Valves LeftVentricle RightVentricle-
    [block:  Dervs ]
       RightVentricle.Dervs
       SystemicVeins.Dervs
       SplanchnicVeins.Dervs
       RightAtrium.Dervs
       PulmArty.Dervs
       PulmCapys.Dervs
       PulmVeins.Dervs
       LeftAtrium.Dervs
       SystemicArtys.Dervs
```

VascularCompartments/SplanchnicVeins.DES

(ref: VascularCompartments.REF) (ref: Ventricles.REF)

```
[structure:  SplanchnicVeins ]
[variables]
  [var:  Inflow ]
  [var:  Outflow ]
  [parm:  V0  =  500.0 ]
  [var:  StressedVol ]
  [var:  Pressure ]
  [var:  ExternalPressure ]
  [parm:  Compliance  =  62.5 ]
  [parm:  Conductance  =  178 ]
[equations]
  der(  Change  ) =  Vol
   intial equation:  Vol  =  1007.0
   errorLimit:  10.0
[definitions]
  [block:  CalcPressure ]
     StressedVol  =  ( Vol - V0 ) MAX 0.0
     ExternalPressure  =  0.0
     Pressure  =  ( StressedVol / Compliance ) + ExternalPressure
  [block:  Dervs ]
     DxMax  = if  Conductance > 0.0  then  0.5 * Compliance / Conductance  else INFINIT
     Inflow  =  OrganFlow.SplanchnicFlow
     Outflow  =  Conductance * ( Pressure - RightAtrium.Pressure )
     Change  =  Inflow - Outflow
```

Figure 6: Visualization tool QHPView, created by us, simplifies viewing of the QHP/HumMod simulator structure, containing more than two thousand XML files, scattered in thousands of directories, where quotations and links between them may not be apparent.

with the previous ***QCP*** simulator, whose mathematical background is hidden from the user in its source code written in C++, the ***HumMod*** simulator uses a different approach. The HumMod authors decided to separate the simulator implementation and description of the model quotations, in order to make the structure of the model more clear and apparent for the larger scientific community.

In 1985 the architect of this model, Thomas Coleman, had already created a special language used to write the model structure, as well as the element definitions into the simulator user interface. The language is based on modified XML notation. The model is then written by using XML files. A special converter/decoder (DESolver) converts XML files into executable simulator code.

A detailed description of this language and DESolver converter, as well as the relevant educational tutorial, is freely accessible on the web page of the University of Mississippi (http://physiology.umc.edu/themodelingworkshop). The new HumMod model is written in the XML language as well. Its structure with all details may be found at (http://HumMod.org), published as an open source.

Therefore, the user can modify this model as he wishes. However, the model description has been divided into more than three thousand XML files in more than thousand directories, from which the special solver creates and executes the simulator (Figure 5).

The entire structure of the model and following links and references are not easily identifiable. That is why the international research and development team in its SAPHIR project (System Approach for Physiological Integration of Renal, cardiac and respiratory control) decided to use the old Guyton models from 1972 [9] and the Ikeda model from 1979 [13] for the creation of its new and extensive model of physiological functions instead of the freely available ***QHP*** model. The source codes of the ***QHP*** model appeared unclear or hard-to-understand to those involved in this project [31].

We have been able to create a special software tool called ***QHPView*** (Figure 6), which is able to create a clear and legible overview of mathematical relations and connections from thousands of source codes. We are offering this tool as an open source on the web page at (http://physiome.cz/HumMod). First, we tried to implement the ***QHP/HumMod*** model in the Simulink environment.

The model contains a wide range of relations that offer solutions for implicit equations. That is why the implementation of this block-oriented model (outputs from one block are used as inputs for the next blocks) is very difficult and as the implementation got more and more complex, the transparency of this model went down quickly. The use of new acasual Simulink libraries in this complex model proved to be problematic and the transparency of the model improved only a little bit.

Therefore, we decided to stop using the Simulink implementation and began to implement in Modelica language (using the Dymola environment). Very quickly we discovered that the ***implementation of a large and extensive model in Modelica is much more effective than using acasual libraries in Simulink***. When we compared the Simulink and Modelica implementations we also discovered a significant difference. Mainly due to the fact that the new acasual libraries are only acasual superstructure of Simulink and not an objectively oriented modeling language based on equatations, as the Modelica language is.

Therefore, if we compare the development environments based on the simulation language Modelica with the Matlab/Simulink development environments made by Mathworks, we may say the following:

- contrary to Simulink, the model implemented in Modelica much better reflects the essentials and base of the modeled reality and the ***simulation models are more clear, readable and less error prone***;
- the ***object architecture*** in Modelica enables the user to build and tweak models with an hierarchical structure gradually, while using ***reusable element libraries***;
- contrary to Simulink (which is the industrial standard from Mathworks), Modelica is a ***non proprietary programming language*** and therefore, it may contain various commercial and non-commercial developing environments competing between each other. This language is used for specific problem solutions originating in various application fields (for commercial and non-commercial specialized libraries);
- in Modelica it is possible to ***combine causal (mostly signals) and acasual links***; and unlike in Simulink, it is also possible, (within interconnected blocks) to create algebraic loops - Modelica compiler uses symbolic manipulations to resolve the loops automatically (when possible) and therefore ***the disconnection of algebraic loops is the task for the development environment and not for the programmer***.

The above specified reasons led us to use, as the main implementation tool for the model creation, the Modelica language and we also gradually stopped using the Matlab/Simulink environment [20].

5 HumMod in Modelica

The implementation of the ***HumMod*** model clearly shows the benefits of the model creation process when done in the Modelica language. If we compare the complex structure of the ***HumMod*** model by using the visualization option in ***QHPView*** (Figure 5) with examples of implementations done in the simulation language Modelica, shown in Figures 7-13, we can see that the acasual implementation done in Modelica creates a transparent and legible model structure and

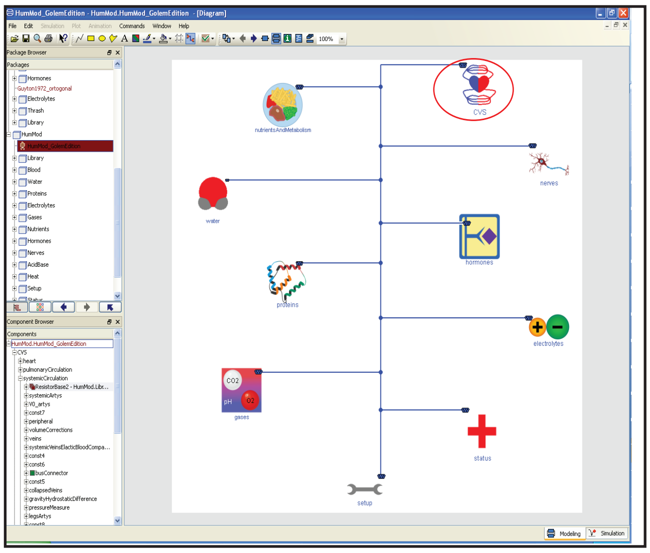


Figure 7: Structure of Hummod model. Model consist of cardiovascular component (CVS), nutritient and metabolism component, water and osmolarity component, proteins component, O_2, CO_2 and acid-base regulation component, electrolyte component, nervous regulation component, hormone regulation component, status of virtual pateint component and setup component. All components ar connected wtih bus connectors.

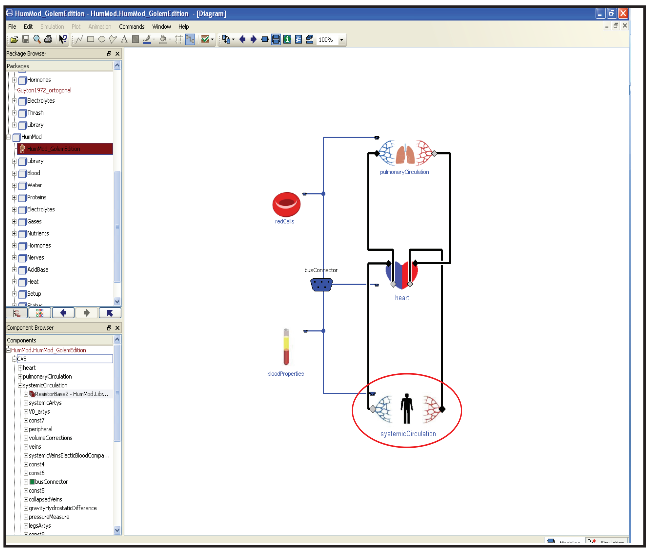


Figure 8: Structure of cardiovascular component (CVS class).

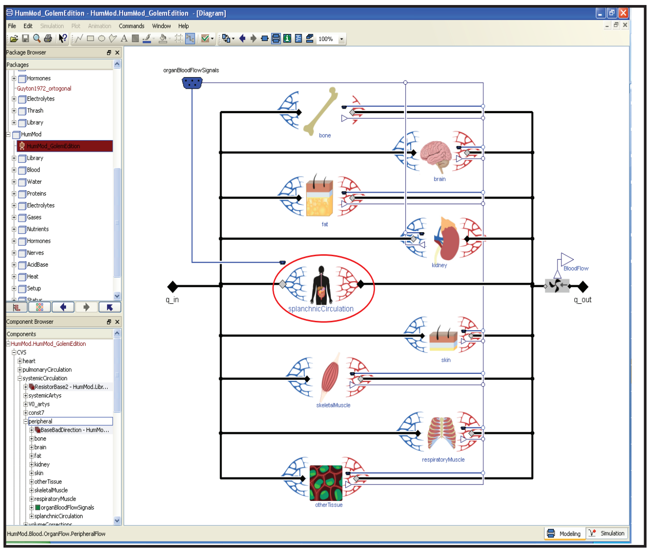


Figure 10: Structure of systemic peripheral circulation component (Peripheral class).

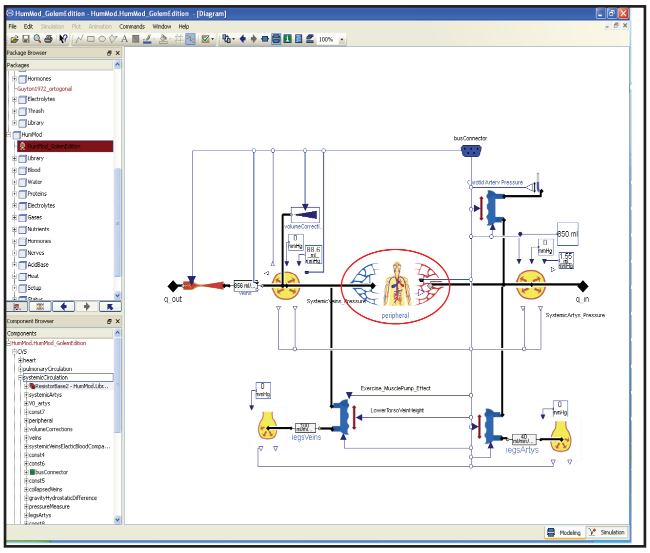


Figure 9: Structure of systemic circulation component (SystemicCirculation class).

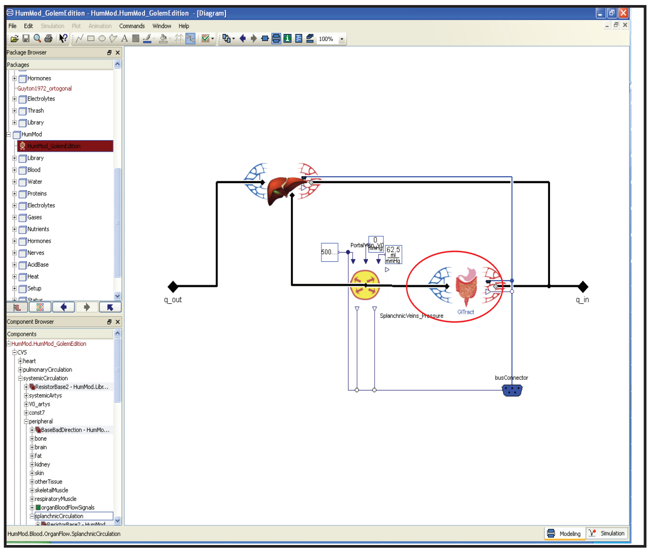


Figure 11: Structure of splanchnic circulation component (SplanchnicCirculation class).

therefore offers easier model modifications.

The ***HumMod*** model implemented in Modelica is being currently ***modified and extended***.

Modifications and extensions of ***HumMod*** were partially taken from our original model Golem [16, 17] and further modified according to newest findings and experiences. Our modifications are mainly extensions, which improve the usability of the model during the modeling of difficult disorders in acid-base, ionic, volume and osmotic homeostasis of inner environments, which is very important for urgent medicinal statuses. Our modification of the HumMod model is based mainly on the process of ***re-modeling the subsystem of acid-base balance***, which is based in the original ***QHP*** on the so-called Stewart acid-base balance theory. Simply put, the so-called "modern approach" of Stewart [30] and his followers (e.g. Fencl et al. [8], Sirker et al. [29]) explaining disorders in the acid-base balance, uses mathematical relations calculating the concentration of hydrogen ions $[H^+]$ from partial pressure CO_2 in plasma (pCO_2), total concentration ($[Buf_{tot}]$), weak (partially dissociated) acids ($[HBuf]$) and their base ($[Buf]$), where:

$$[Buf_{tot}]=[Buf]+[HBuf]$$

and from the difference between the concentration of fully dissociated cations and fully dissociated anions in *SID* (strong ion difference):

$$[H+]=Function\ (pCO_2,\ SID,\ Buf_{tot})$$

The problem of this approach is that the precision of acid-base calculations in the model depends on the

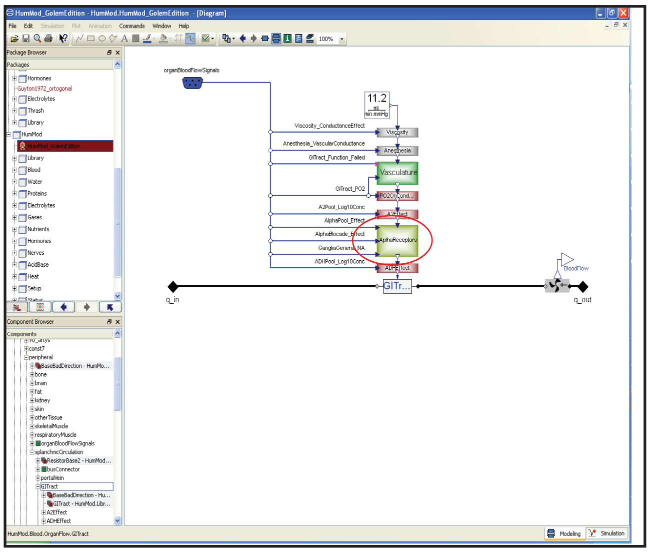


Figure 12: Structure of gastrointestinal vascular resistance component (GITract class).

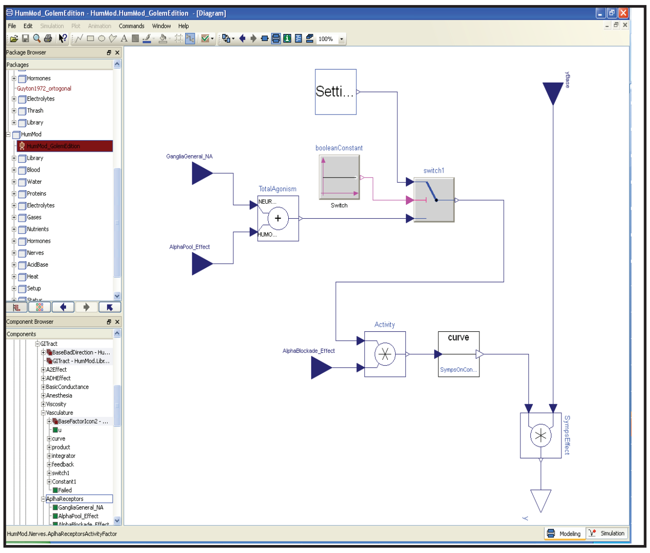


Figure 13: Structure of component calculating influence of alpha receptors stimulation on gatrointestinal vascular resistance (AlphaReceptors class).

precision of the SID calculation, that is the difference between the concentration of fully dissociated cations (that is mainly sodium and potassium) and fully dissociated anions (mostly chlorides). Imprecision that is created during the modeling of sodium, potassium and chlorides intake and excretion are transferred and reflected by the imprecision in the modeling process of the acid-base status.

Even though Hester et al. [11], significantly improved the modeling of reception and excretion of sodium, potassium and chlorides in kidneys in his ***HumMod*** model, if we model a long-term status (when nothing is happening with the virtual patient), the virtual patient (in the current model version) has a tendency to fall into slight and steady metabolic acidosis after one month of the simulated time.

Our evaluative approach towards the modeling and evaluation of disorders in acid-basic balance [14, 15, 21] is based on the modeling and evaluation of two flows – the creation and excretion of CO_2 and the creation and excretion of strong acids, connected through the purification systems of each part of the bodily fluids. This approach, according to our opinion, better explains the physiological causality of acid-base regulations, rather than direct modeling of acid-base disorders through the balancing of accompanying electrolytes. Besides, the fidelity and truthfulness of the modeling process is getting better; mainly in mixed (acid-base and electrolyte) disorders in inner environments.

Another important modification of the ***HumMod***, is the fact that the ***model was extended by adding the dependency of the potassium flow on the intake of glucose as a result of insulin***, which enables us to model (besides other things), the influence of potassium solution infusions with insulin and glucoses, which are distributed in acute medicine for treating potassium depletions.

We have been using this "balancing and evaluation" approach [18] towards the modeling of acid-base balance in our old "Golem" simulator [17]. The extended ***HumMod*** model serves as the base for the educational simulator ***„eGolem"***, used in medical tutoring in clinical physiology of urgent statuses which is being currently developed.

On the webpage http://physiome.cz/HumMod you may find the updated and current structure of our implementation of the ***HumMod*** model (***„HumMod--Golem edition"***). In collaboration with M. Tiller we are preparing a detailed description of this model with extensive descriptions of the various physiological regulatory circuits.

6 From a model to the simulator

A simulation model, implemented in a sophisticated development environment, cannot be used for education as is alone. It is the implementation of the formalized description of the modeled reality that enables testing of the behavior of the mathematic model during various input values and the search for model quotations and parameters, which within the established precision range, can ensure the sufficient compatibility of the behavior of the model with the modeled system (model identification).

Even after this goal is reached, there is still a long

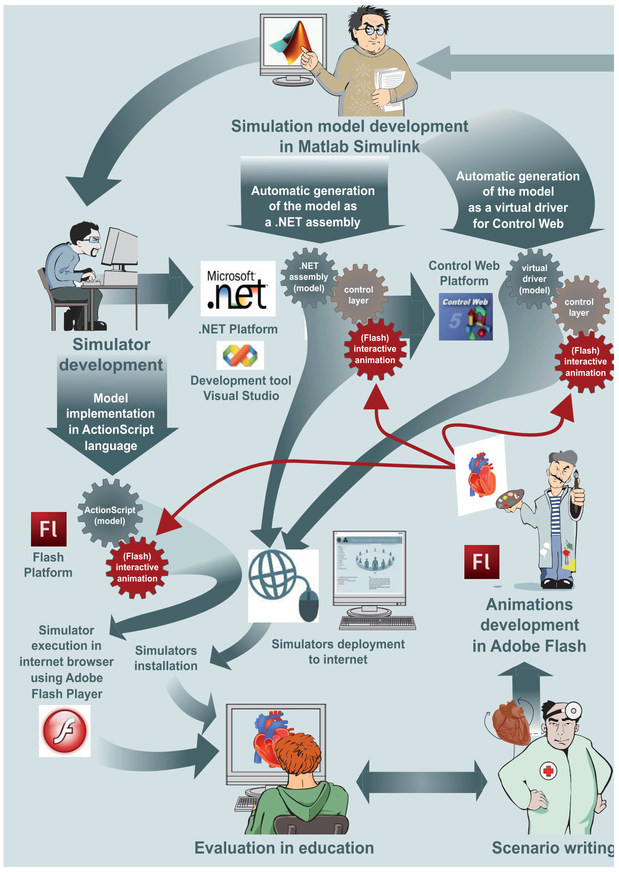

Simulation model development in Matlab Simulink
Automatic generation of the model as a .NET assembly
Automatic generation of the model as a virtual driver for Control Web
Simulator development
Model implementation in ActionScript language
Microsoft .net
.NET Platform
Development tool Visual Studio
.NET assembly (model)
control layer
(Flash) interactive animation
Control Web Platform
virtual driver (model)
control layer
(Flash) interactive animation
ActionScript (model)
Fl
Flash Platform
(Flash) interactive animation
Simulator execution in internet browser using Adobe Flash Player
Simulators installation
Simulators deployment to internet
Fl
Animations development in Adobe Flash
Evaluation in education
Scenario writing

Figure 14: The original solution of creative interconnection of tools and applications, used for the creation of simulators and tutorial programmes using simulation games. The base of an e-learning program is a high-quality script, created by an experienced pedagogue. The creation of animated pictures is done by artists who create interactive animations in Adobe Flash. The core of simulators is the simulation model, created with special development tools, designed for the creation of simulation models. For a long time, we have been using Matlab/Simulink made by Mathworks for the development of models. The simulator development process is a demanding programming work. To make this task easier, we have developed special programmes that simplify the automatic transfer process of simulation models created in Matlab/Simulink over to ControlWeb or over to the Microsoft .NET environment.

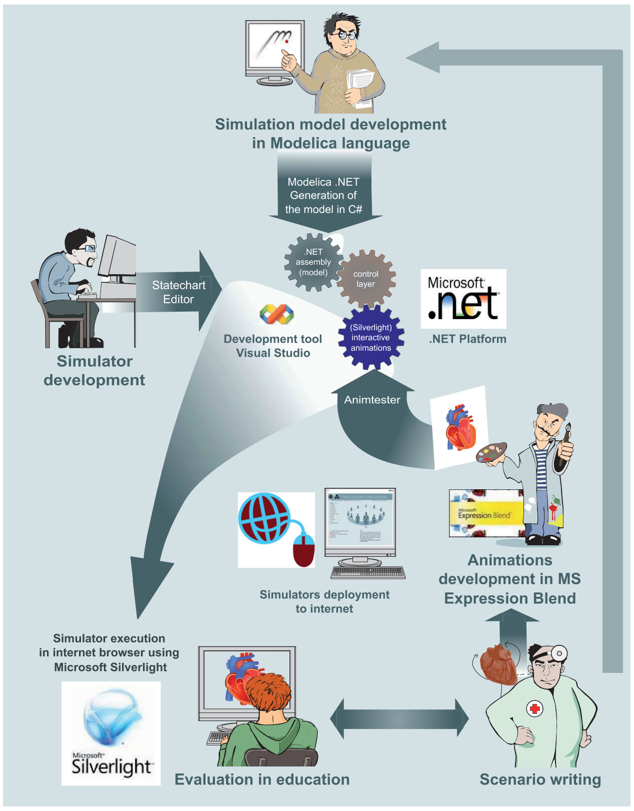

Simulation model development in Modelica language
Modelica .NET Generation of the model in C#
Statechart Editor
.NET assembly (model)
control layer
Microsoft .net
.NET Platform
Simulator development
Development tool Visual Studio
(Silverlight) interactive animations
Animtester
Simulators deployment to internet
Animations development in MS Expression Blend
Simulator execution in internet browser using Microsoft Silverlight
Microsoft Silverlight
Evaluation in education
Scenario writing

Figure 15: Our new solution of creative interconnection of tools and applications, used for the creation of simulators and tutorial programmes using simulation games. The base of an e-learning program is still a high-quality script, created by an experienced pedagogue. The creation of animated figures is done by artists who create interactive animations in Expression Blend. To create and test animations that will be controlled by the simulation model, art designers use the Animtester software tool, developed by us. The core of simulators is the simulation model, created in the Modelica simulation language environment. Within the project Open Modelica Source Consortium, we are in the process of creating a tool which is able to generate the source code from Modelica to C# language. This enables us to generate a component from .NET used in the final application on the Silverlight platform, which enables to distribute the simulator as a web application, running in the internet browser (even on computers with various operating systems).

road ahead from the identified model to the educational or tutorial simulator. It is a very demanding development work, which requires the combination of ideas and experiences of teachers who create the script of the tutorial application, the creativity of art designers who create the multimedia components interconnected with the simulation model in the background, as well as the efforts of programmers who finally "sew up" the final masterpiece into its final shape. We have used a special web simulator creation technology for creation of educational simulators [20].

To automate the model debugging transfer from the simulation development environment (previously using Simulink and nowadays using Modelica) into the development environment where the development application is programmed, specialized software tools (developed by us) are used. We have been creating tutorial simulators in Microsoft .NET and Adobe Flash environments (Figure 14). Recently, we began using the Microsoft Silverlight platform (Figure 15), which enables distribution of simulators over the internet and may be executed directly into the internet browser environment (even on computers running various operating systems).

7 Conclusions

Nowadays, the old Comenius motto – "schola ludus," or "playful school" [6], has found a modern use in interactive educational programs that use simulation games. Connection of the multimedia environment, serving as an audio-visual user interface, with simulation models, gives the studied problem a much more tangible feeling. A simulation game offers the possibility to test, without any risk, the simulated object's behavior. The behavior of individual physiological subsystems can be appreciated in a simulation game, both under normal conditions and in the presence of a disorder.

Complex integrative simulators of human physiology can be of large importance when teaching pathophysiology or studying pathogenesis of varied medical conditions and syndromes using virtual patients. Such simulators include models of not only individual physiological subsystems but also of their mutual connection into more complex units. Modelica is a very convenient developing tool for design of those complex hierarchical models.

References

[1] Abram, S.R., Hodnett, B.L., Summers, R.L., Coleman, T.G., Hester R.L., Quantitative Circulatory Physiology: An Integrative Mathematical Model of Human Physiology for medical education. *Advannced Physiology Education*, 31 (2), pp.202 - 210, 2007.

[2] Amosov, N.M, Palec, B.L., Agapov, G.T., Ermakova, I.I., Ljabach, E.G., Packina, S.A., Soloviev, V.P.. *Theoretical research of physiological systems (in Russian).* Naukova Dumaka, Kiev, 1977.

[3] Bassingthwaighte J. B., Strategies for the Physiome Project", *Annals of Biomedical Engeneering* 28, pp. 1043-1058, 2000.

[4] Brugård, J., Hedberg, D., Cascante, M., Gedersund, G., Gómez-Garrido, A., Maier, D., Nyman, E., Selivanov, V., Stralfors, P., Creating a Bridge between Modelica and the Systems Biology Community, *Proceedings 7th Modelica Conference, Como, Italy, Sep. 20-22, 2009.*, pp. 473-479, The Modelica Association., Como, 2009. Available http://www.ep.liu.se/ecp/043/052/ecp09430016.pdf.

[5] Cellier, F. E., Nebot, A., Object-oreiented modeling in the service of medicine, *Proceedings of the 6tha Asia Conference, Bejing, China 2006.* 1, pp 33-40. International Academic Publishers, Bejing, 2006.

[6] Comenius, J. A., *Schola ludus seu Encyclopaedea Viva.,* Sarospartak, 1656.

[7] Fencl, J., Jabor, A., Kazda, A., Figge, J., Diagnosis of metabolic acid-base disturbances in critically ill patients, *Am. J. Respir. Crit. Care.*, 162, pp. 2246-2251, 2000.

[8] Guyton A. C., Coleman T. A., Granger H. J., Circulation: Overall Regulation, *Ann. Rev. Physiol.*, 41, 13-41, 1972.

[9] Haas, O. C., Burnham, K. J., Systems Modeling and Control Applied to Medicine, in O. C. Haas, K. J. Burnham, *Intelligent and Adaptive Systems in Medicine*, pp. 17-52, CRC Press, Boca Raton Fl , 2008.

[10] Hester R. L., Coleman T., Summers, R., A multilevel open source model of human physiology."*The FASEB Journal*, 22, p. 756, 2008

[11] Hester, R.L, Ilescu, R., Summers R. L , Coleman T. G., Systems biology and integrative physiological modeling, *Journal of Physiology*, Published online before print December 6, 2010, doi: 10.1113/jphysiol.2010.201558, Availabe: http://jp.physoc.org/content/early/2010/12/01/jphysiol.2010.201558.full.pdf+html.

[12] Hunter P.J., Robins, P., Noble D., The IUPS Physiome Project, *Pflugers Archive-European Journal of Physiology*, 445, pp. 1–9, 2002.

[13] Ikeda N., Marumo F., Shirsataka M. A., Model of overall regulation of body fluids, *Ann. Biomed. Eng.* 7, pp. 135-166, 1979.

[14] Kofránek, J., Modelling of blood acid base balance, *Ph,D, Thesis,* Charles University, Faculty od General Medicine, Prague, 1980.

[15] Kofránek, J., Complex model of acid-base balance (in Czech)., in *M. Zeithamlová (Editor), MEDSOFT 2009*, Praha: Agentura Action M., pp. 23-60. English translation of the paper is

available at http://www.physiome.cz/references/medsoft2009acidbase.pdf, model is available at http://www.physiome.cz/acidbase.

[16] Kofránek, J., Andrlík, M., Kripner, T., Mašek, J., Simulation chips for GOLEM – multimedia simulator of physiological functions, in J. G. Anderson, M. Kapzer (Editor), *Simulation in the Health and Medical Sciences 2002*. pp. 159-163, Society for Computer Simulation International, Simulation Councils, San Diego, 2002. Available: http://www.physiome.cz/references/simchips2002.pdf.

[17] Kofránek J. Anh Vu L. D., Snášelová H., Kerekeš R. and Velan T., GOLEM – Multimedia simulator for medical education, in *MEDINFO 2001, Proceedings of the 10th World Congress on Medical Informatics. (London, UK, 2001)*, Patel, L., Rogers, R., Haux R. Eds., pp. 1042-1046, IOS Press, London, Available: http://www.physiome.cz/references/medinfo2001.pdf.

[18] Kofránek, J., Mateják, M., Matoušek, S., Privitzer, P., Tribula, M., & Vacek, O., School as a (multimedia simulation) play: use of multimedia applications in teaching of pathological physiology. *MEFANET 2008. CD ROM Proceedings,* (ISBN 978-80-7392-065-4), kofranek.pdf: pp. 1-26, Masarykova Univerzita, Brno, 2008. Available: http://www.physiome.cz/references/mefanet2008.pdf.

[19] Kofránek, J., Mateják, M. Privitzer, P., Tribula, M., Causal or acausal modeling: labour for humans or labour for machines, in V C. Moler, A. Procházka, R. Bartko, M. Folin, J. Houška, P. Byron (Editor), *Technical Computing Prague 2008, 16th Annual Conference Proceedings, CD ROM,* 058_kofranek.pdf: pp. 1-16. Humusoft s.r.o., Praha, 2008, Available: http://www.physiome.cz/references/tcp2008.pdf.

[20] Kofránek, J. Mateják, M., Privitzer, P.:Web simulator creation technology. *MEFANET report*, 3, pp. 32-97. Available: http://www.physiome.cz/references/mefanetreport3.pdf.

[21] Kofránek, J., Matoušek, S., Andrlík, M., Border flux ballance approach towards modelling acid-base chemistry and blood gases transport, in V B. Zupanic, S. Karba, S. Blažič (Editor), Proceedings of the 6th *EUROSIM Congress on Modeling and Simulation, Full Papers (CD)* (TU-1-P7-4, pp. 1-9), University of Ljubljana, Ljubljana 2007. Available: http://www.physiome.cz/references/ljubljana2007.pdf.

[22] Kofránek, J., Matoušek, S., Rusz, J., Privitzer, P., Mateják, M, Tribula, M., The Atlas of physiology and pathophysiology: web-based multimedia enabled interactive simulations, *Computer Methods and Programs in Biomedicine*, Article in Press, doi:10.1016/j.cmpb.2010.12.007, 11 pp., 2011, Available: http://www.physiome.cz/references/CMPB2011.pdf.

[23] Kofránek, J, Rusz, J., Restoration of Guyton diagram for regulation of the circulation as a basis for quantitative physiological model development *Physiological Research, 59,* pp 897-908, 2010, Available: http://www.biomed.cas.cz/physiolres/pdf/59/59_897.pdf.

[24] Logan, J. D., Wolesensky, J. D., *Mathematical methods in biology.* John Wiley & Sons,,Inc., Hoboken, NJ, 2009.

[25] McLeod, J., PHYSBE: A ophysiological simulation benchmark experiment, *Simulation*, 15: pp. 324-329, 1966.

[26] McLeod, J., PHYSBE...a year later, *Simulation*, 10, pp. 37-45, 1967.

[27] McLeod, J., Toward uniform documentation-PHYSBE and CSMP, *Simulation*, 14, pp. 215-220, 1970.

[28] Oomnes, C., Breklemans, M., Baaijens, F., *Biomechanics: concepts and computation.* Cambridge University Press, Cambridge, 2009.

[29] Sirker, A. A., Rhodes, A., Grounds, R. M.,Acid-base physiology: the ‚traditional‘ and ‚modern‘ approaches, *Anesthesia*, 57, pp. 348-356, 2001.

[30] Stewart, P. A., Modern quantitative acid-base chemistry. *Can. J. Physiol. Pharmacol.*, 61, 1444-1461, 1983.

[31] Thomas, R. S., Baconnier, P., Fontecave, J., Francoise, J., Guillaud, F., Hannaert, P., Hermandéz, A., Le Rolle, V., Maziére, P, Tahi, F., White R. J., SAPHIR: a physiome core model of body fluid homeostasis and blood pressure regulation, *Philosophical Transactions of the Royal Society*, 366, pp. 3175-3197, 2008.

[32] Wallish, P., Lusignan, M., Benayoun, M., Baker, T. I., Dickey, A. S., Hatsopoulos, N. G., *MATLAB for Neuroscientists: An Introduction to Scientific Computing in MATLAB.* Academic Press, Burlington, MA, 2008.

Acknowledgement

Work on the development of medical simulators has been supported by the grants MPO FR-TI3/869, MSM 0021620806 and by Creative Connections s.r.o.