České vysoké učení technické v Praze, Fakulta elektrotechnická

MUDr. Jiří Kofránek, CSc.

TVORBA LÉKAŘSKÝCH SIMULÁTORŮ

Habilitační práce

Pracoviště: Univerzita Karlova v Praze
1. lékařská fakulta, Ústav patologické fyziologie
Oddělení biokybernetiky a počítačové podpory výuky

Praha 30. prosince 2009

Poděkování

Tvorba biomedicínských simulačních modelů, výukových programů využívajících simulační hry a lékařských výukových simulátorů byla podporována řadou grantů a rozvojových projektů, které jsem koordinoval: rozvojové projekty MŠMT 394/2004, MŠMT 195/2005, MŠMT 207/2006, FRVŠ 981/2007, MŠMT C34/2007, MŠMT C20/2008, výzkumný grant GAUK 217/1998/C/1.LF (1998-2000), výzkumný záměr MSM 1111008 (1999-2004), projekt výzkumu a vývoje MŠMT 2C06031 (2006-2009).

Výsledkem této podpory je především vznik multidisciplinárního řešitelského týmu lékařů, techniků, matematiků a výtvarníků, adekvátně hardwarově i softwarově vybavené pracoviště - Oddělení biokybernetiky a počítačové podpory výuky, formalizovaný popis klíčových fyziologických systémů a řada příslušných publikací i navázání mezinárodní spolupráce s obdobnými pracovišti v Evropě i USA. Z praktického hlediska je výsledkem stále rostoucí sada internetem dostupných multimediálních výukových programů a multimediálních výukových simulátorů. Významným a prakticky využitelným výsledkem je ovšem vypracovaná metodologie jejich efektivní tvorby včetně originálních námi vyvinutých softwarových nástrojů.

Na vybraných výsledcích z této oblasti se podíleli doktorandi, jejichž jsem školitelem: Mgr. Marek Mateják, MUDr. Stanislav Matoušek, MUDr. Mgr. Pavol Privitzer, Ing. Martin Tribula a Ing. Ondřej Vacek.

Mé poděkování patří také bývalému pracovníkovi našeho oddělení, ing. Michalu Andrlíkovi a mé bývalé doktorandce Mgr. Anně Lábajové, PhD., s nimiž jsme metodologii vytváření výukových simulátorů začínali a kteří stáli u počátků internetového Atlasu fyziologie a patofyziologie.

Nesmím zapomenout i na naše výtvarníky, akademické malíře Miroslava Gemrota a Josefa Hlaváčka, kteří stáli u zrodu naší spolupráce s výtvarnou školou Václava Hollara a následným založením vyšší odborné školy s novým tříletým oborem “interaktivní grafika”. Její absolventi, (Kateřina Tribulová, DiS., Stanislava Karbušická, DiS., Iva Češková, DiS., Pavla Holá, DiS., Kateřina Kofránková, DiS.) jakož i další bývalí i stávající studenti vyšší odborné školy se podíleli a podílejí na tvorbě uživatelského rozhraní našich výukových simulátorů..

Mé zvláštní poděkování patří doc. MUDr. Zdeňku Wünschovi, který před mnoha lety vedl mé první kroky v oblasti biokybernetiky, a který je dodnes aktivním členem našeho vývojového týmu. Ukazuje se, že mnoho jeho konceptů, formulovaných před desítkami let, dnes nachází své uplatnění.

Musím vzpomenout i doc. MUDr. Zdeňka Pokorného, CSc., který před léty vedl mou disertační práci jako neoficiální školitel (z politických důvodů měl zakázáno vyučovat), od kterého jsem dostal mnoho cenných rad nejen v oblasti odborné práce.

Oddělení biokybernetiky a počítačové podpory výuky vzniklo v roce 1995 na Ústavu patologické fyziologie 1. LF UK především díky iniciativě a mimořádnému porozumění přednosty ústavu prof. MUDr. Emanuela Nečase, DrSc. Přál bych si proto, aby poděkováním za jeho dlouholetou podporu byly výsledky práce tohoto našeho oddělení.

Musím se v této souvislosti poděkovat i děkanovi 1. lékařské fakulty UK v Praze, prof. MUDr. Tomáši Zimovi, DrCs., který od svého nástupu do úřadu děkana velmi podporuje rozvoj e-learningu a uplatnění elektronických výukových materiálů v pedagogické praxi lékařské výuky.

V neposlední řadě patří mé poděkování i rodině, především mé ženě Radaně, která mi po celou dobu vytváří zázemí pro tvůrčí práci.

Motto:

„Snili jsme po léta o instituci nezávislých učenců, kteří spolupracují nikoli jako podřízení nějakého velkého vedoucího, ale spojení touhou, ba duchovní potřebou, pochopit tuto oblast jako celek a dodávat si navzájem sílu, aby to pochopili.”

Norbert Wiener

Obsah

ANOTACE PRÁCE

1 Úvod

Autor této práce se zabývá tvorbou modelů rozsáhlých fyziologických systémů již od počátku osmdesátých let. Rozšířením modelu acidobazické rovnováhy krve (Kofránek J., 1980) byl komplexní model regulace vnitřního prostředí (Kofránek, Pokorný, Wünsch, Brelidze, Gondžilašvili, & Verigo, 1982a-c; Kofránek, Brelidze, & Gondžilašvili, 1984; Gondžilašvili, Kofránek, Pokorný, & Brelidze, 1987). V osmdesátých letech v této oblasti autor spolupracoval na tvorbě rozsáhlých modelů fyziologických regulací v rámci tehdejšího sovětského kosmického výzkumu (Verigo, 1987). Autor rovněž zabýval i problematikou využití simulačních modelů pro vyhodnocování klinicko-fyziologických dat (Kofránek, a další, 1988), využitím modelů pro vysvětlení mechanismů fyziologických regulací např. (Maršálek & Kofránek, 2005; Kofránek, Matoušek, & Andrlík, 2007) i jejich využitím v lékařské výuce, kde simulační modely jsou jádrem s lékařských simulátorů a výukových programů využívajících simulační hry.

Perspektivní oblasti praktického uplatnění simulačních modelů v lékařské výuce a návazným technologiím jejich tvorby se věnuje tato práce. Je výsledkem více než dvacetiletých zkušeností autora v této oblasti. Během této doby se značně posouvaly technologické možnosti, jak v oblasti modelování, tak i v oblasti softwarových nástrojů pro tvorbu multimediálních výukových aplikací, které dnes mimo jiné umožňují vytvářet výukové simulátory spustitelné přímo v internetovém prohlížeči.

2 Východisko práce a současný stav řešené problematiky

Vstupní kapitola se nejprve věnuje motivačním pohnutkám, které vedly nejen k sepsání této práce, ale nakonec i k volbě celoživotního odborného zaměření jejího autora.

Prvotním impulzem byl článek, na který autor této práce v roce 1974 náhodně narazil ve Státní lékařské knihovně. Článek (Guyton, Coleman, & Grander, 1972) byl uveřejněn v ročence Annual Review of Physiology, věnované přehledovým článkům z oblasti fyziologie. Vévodilo mu rozsáhlé schéma, (viz obr. 1 na straně 3) svým vzhledem připomínalo spíše zapojení elektronických obvodů než ilustrační znázornění fyziologických regulací. Jednotlivé prvky tohoto rozsáhlého schématu představovaly matematické operace a celé schéma v podstatě reprezentovalo soustavu matematických vztahů, popisujících širší fyziologické souvislosti a příslušné regulační vazby oběhového systému. Guytonovo schéma bylo jedním z prvních matematických popisů fyziologických systémů propojených do komplexního celku. Bylo mnohokrát přetiskováno v nejrůznějších publikacích (v poslední době např. Hall, 2004; Van Vliet & Montani, 2005).

Podobnost původní Guytonovy grafické notace a vnější grafické reprezentace jednotlivých bloků v prostředí Simulink bylo pro nás inspirací k převedení klasického Guytonova diagram z roku 1972 do podoby funkčního simulačního modelu v Simulinku (viz obr. 6 na straně 8) při zachování identického vnějšího vzhledu jako v původním grafickém schématu (Kofránek & Rusz, 2007). Simulační vizualizace původního schématu ale vyžadovala odstranit určité drobné chyby (viz obr. 5 na straně 7) v původním schématu (Kofránek, Rusz, & Matoušek, 2007). Guyton a spolupracovníci svůj model původně implementovali ve Fortranu (nezávisle na grafickém schématu). Je zajímavé, že dosud nikdo z těch, kdo Guytonovo schéma přetiskoval na tyto chyby neupozornil.

Guyton, jeho spolupracovníci a žáci model dále rozvíjeli (Montani, Adair, Summers, Coleman, & Guyton, 1989), modifikovaný Guytonův model byl využíván v programu „Digital Astronauts“ NASA (White & Phee, 2006), byl podkladem i inspirací pro tvorbu rozsáhlých modelů fyziologických regulací, sloužících pro vysvětlení kauzálních řetězců reakcí organismu na nejrůznější podněty i pro pochopení mechanismů rozvoje nejrůznějších patologických stavů.

V devadesátých letech technologický pokrok umožnil využití rozsáhlých modelů fyziologických systémů spustitelných na personálním počítači. Objevily se také i vývojové nástroje určené pro tvorbu simulačních modelů – např. Matlab/Simulink od firmy Mathworks. Následný rozvoj informačních a komunikačních technologií spolu s rozšířením vysokorychlostního internetu v posledních letech přinesl další možnosti uplatnění simulačních modelů jako velice efektivní výukové pomůcky. Objevila celá plejáda výukových programů, využívajících interaktivní multimediální rozhraní pro vysvětlení složitých procesů v biomedicínských vědách ať již ve všeobecně dostupných (např. http://www.apsarchive.org) nebo komerčních (např. http://interactivephysiology.com) aplikacích. Většinou však jde o animace a nikoli o simulace.

Zároveň se ale také objevují výukové programy využívající simulační modely. Většinou se týkají modelů jednotlivých fyziologických subsystémů. Pro výuku patofyziologie a studium patogenezy nejrůznějších patologických stavů mají velký význam komplexní simulátory, zahrnující ***modely nejen jednotlivých fyziologických subsystémů, ale i jejich propojení do komplexnějšího celku***. Již v roce 1982 Guytonův žák a spolupracovník Thomas Coleman vytvořil model „Human" určený především k výukovým účelům (Coleman & Randall, 1983). V poslední době Meyers a Doherty implementací v Javě původní Colemanův model zpřístupnili na webu (Meyers & Doherty, 2008). Dalším, velmi podstatným, rozpracováním modelu Human je rozsáhlý výukový simulátor Quantitative Circulatory Physiology (QCP) (Abram, Hodnett, Summers, Coleman, & Hester, 2007) který, pro podporu jeho využívání jako výukové pomůcky v lékařské výuce, autoři volně zpřístupnili na webu (http://physiology.umc.edu/themodelingworkshop/). Jeho dalším rozšířením je výukový simulátor Quantitative Human Physiology (QHP), (Coleman, Hester, & Summers, 2009) obsahující více než 4000 proměnných, který v současné době zřejmě představuje nejrozsáhlejší model fyziologických regulací. Simulátor je veřejně přístupný na stejné webové adrese jako QCP (poslední verzi autoři nazvali již Digital Human). Struktura modelu, včetně všech matematických vztahů, je veřejně přístupná jako „open source", je ale zapsána pomocí speciálního jazyka, založeného na upravené XML notaci v celkem 2833 souborech umístěných v 772 složkách. Celková struktura modelu a jednotlivé návaznosti jsou proto velice nepřehledné. Pro přehledné zobrazení matematických vztahů z této notace jsme vytvořili speciální softwarový nástroj QHPView a nabídli ho na webových stránkách autorů QHP jako „open source" (viz obr. 19 na straně 21).

Zveřejňování matematického modelu u výukových simulátorů je ale často výjimkou. Model se nezřídka stává pečlivě chráněným firemním know how. Zejména to platí o hardwarových simulátorech určených nejen k praktickému procvičování některých zdravotnických úkonů na figurině pacienta (např. kardiopulmonální resuscitace, intubace, katetrizace apod.) a také i k procvičování lékařského rozhodování.

Tvorba modelů fyziologických systémů úzce souvisí s problematikou formalizace Formalizovanému popisu fyziologických systémů je v současné době věnován mezinárodní projekt PHYSIOME (http://www.physiome.org), který je nástupcem projektu GENOME, jehož výsledkem byl podrobný popis lidského genomu. Cílem projektu PHYSIOME je formalizovaný popis fyziologických funkcí. Metodickým nástrojem jsou zde počítačové modely (Bassingthwaighte, 2000; Hunter, Robins, & Noble, 2002).

3 Zkušenosti s tvorbou a využitím lékařského výukového simulátoru a cíle další práce

Kapitola popisuje zkušenosti autora z tvorby a využití výukového lékařského simulátoru Golem (Kofránek, Velan, & Kerekeš, 1997; Kofránek & Velan, 1999; Kofránek, Velan, & Janicadis, 2000; Kofránek, Anh Vu, Snášelová, Kerekeš, & Velan, 2001; Kofránek, Andrlík, Kripner, & Mašek, 2002a; Kofránek, Andrlík, & Kripner, 2003; Kofránek, a další, 2004; Kofránek, Andrlík, & Kripner, 2005), na jejichž základě byly formulovány cíle práce.

Teoretickým podkladem simulátoru Golem byl rozsáhlý model propojených fyziologických systémů. Požadavek komplexnosti vyplývá z toho, že při selhání jednotlivých orgánů se zapojují kompenzační mechanismy různých fyziologických systémů. Podkladem lékařských simulátorů proto nemohou být izolované modely jednotlivých fyziologických systémů, ale komplexní integrované modely. Proto také

tvorba lékařských simulátorů není jen čistě vývojová práce, ale předpokládá také teoretický výzkum formalizace propojených fyziologických regulací lidského organismu.

Vlastní simulátor byl implementován ve vývojovém prostředí Control Web, původně určeném pro tvorbu a vizualizaci řídících a měřících systémů v průmyslu. Control Web poskytl velmi výkonné prostředky pro tvorbu a vývoj uživatelského rozhraní simulátoru. Na rozdíl od průmyslových aplikací komponenty uživatelského rozhraní nekomunikovaly přes příslušný ovladač s technologickým zařízením ale se speciálně naprogramovaným ovladačem, v jehož jádru byl simulační model. Místo posílání dat do připojené technologie se zadávaly vstupy modelu, a místo vizualizace dat načítaných z připojeného zařízení se zobrazovaly výstupy modelu. Vlastní ovladač byl původně psán v jazyce Modelica, v pozdějších implementacích v jazyce C++. Nakonec byl vyvinut speciální nástroj, který umožnil generovat zdrojový text ovladače přímo ze simulinkového schématu (viz obr. 46 na straně 44). To umožnilo jednoduše modifikovat simulátor při nejrůznějších úpravách simulačního modelu v prostředí Simulink (Kofránek, Andrlík, Kripner, & Mašek, 2002a).

Simulátor byl šířen bezplatně jako public domain a využíval se ve výuce lékařů v některých našich i zahraničních lékařských fakultách***. Zkušenosti s jeho výstavbou a jeho využitím*** ve výuce lze shrnout do následujících bodů:

- Z didaktického hlediska se ukázalo velmi výhodné, že jsme do simulátoru Golem zavedli možnost rozpojovat některé regulační smyčky a umožnit studentům v simulační hře sledovat reakce zvoleného fyziologického subsystému na změny vstupních veličin (které jsou ovšem v reálném organismu samy regulovány). To umožňuje studentům lépe pochopit význam jednotlivých regulačních okruhů a studovat vliv (rozpojených a zprvu „ručně“ řízených) regulačních vazeb na chování organismu při nejrůznějších patologických poruchách a reakcích na příslušnou terapii. Podle našich zkušeností tento přístup vede k lepšímu pochopení významu jednotlivých regulačních smyček a porozumění jejich úlohy v patogeneze nejrůznějších onemocnění a pochopení patofyziologických principů příslušných léčebných zásahů (Kofránek, Anh Vu, Snášelová, Kerekeš, & Velan, 2001).
- Na druhé straně ale naše zkušenosti s uplatněním simulátoru Golem ve výuce ukázaly, že velké a složité simulátory mají z didaktického hlediska značnou nevýhodu v poměrně komplikovaném ovládání. Obdobnou zkušenost s didaktickým využitím složitých modelů ve výuce potvrzují i zahraniční autoři (Lane, 2001; de Freitas, 2006). Velké množství vstupních a výstupních proměnných totiž vyžaduje od uživatele důkladnější porozumění struktury modelu i znalost toho, jak složitý model ovládat a jaké proměnné je vhodné při simulaci nejrůznějších patologických stavů sledovat (v opačném případě se složitý model uživateli jeví jako složitá a málo srozumitelná technická hračka). Výukové modely pro jejich efektivní didaktické využití ve výuce proto musí být provázeny s interaktivním výkladem – teprve spojení výkladu se simulační hrou dává možnost využití všech výhod virtuální reality pro vysvětlení složitých regulačních procesů ve zdravém i nemocném organismu.
- Z didaktického hlediska je vhodné při výkladu postupovat od jednoduchého ke složitějšímu, a proto je vhodné při výkladu využívat nejprve jednodušší agregované modely (s několika proměnnými), s jejich pomocí vysvětlit základní principy a poté model (a popisovanou fyziologickou realitu) postupně zesložiťovat. I jednoduchý interaktivní model může být dobrým pomocníkem pro vysvětlení patogenetických řetězců rozvoje nejrůznějších patologických stavů.
- Z hlediska metodologie tvorby simulátoru se ukázalo jako výhodné používat odlišné nástroje pro modelování (v případě simulátoru Golem – vývojové prostředí Matlab/Simulink) a jiné nástroje pro tvorbu simulátoru (Control Web).
- Výhodnou se jevila i komponentová výstavba modelů ve formě hierarchicky uspořádaných komponent (jakýchsi „simulačních čipů“), které zpřehledňují strukturu modelu a usnadňují mezioborovou spolupráci.
- Ukázalo se také, že ideálním prostředím pro distribuci i vlastní provozování výukových simulátorů je prostředí internetu. Jako didakticky účinné se ukázalo využití interaktivních animací, které jsou řízené výstupy modelu. Pro tyto požadavky však dosud používaná technologie výstavby simulátoru prostřednictvím vývojového prostředí Control Web se nejevila zcela opti-

mální.

Během postupné tvorby nových verzí výukového simulátoru Golem přicházely nové technologie, které slibovaly nové možnosti pro vytváření multimediálních výukových simulačních aplikací dosažitelných přes internetový prohlížeč. Objevily se i nové technologie zefektivňující výstavbu simulačního jádra výukových simulátorů.

Na základě dosavadních zkušeností s výstavbou a využitím simulátoru Golem byly proto stanoveny ***cíle další práce***:

- v technologii tvorby simulačních modelů využít nové technologie, zefektivňující tvorbu, testování a ladění složitých hierarchicky uspořádaných modelů;
- zavézt novou technologii tvorby výukových simulátorů propojujících výklad se simulační hrou;
- vytvořit nástroje umožňující propojení interaktivních animací se simulačním modelem na pozadí;
- vytvořit nástroje usnadňující týmovou multidisciplinární spolupráci při vytváření výukových simulačních aplikací (pedagogů, tvůrců simulačních modelů, výtvarníků a programátorů);
- zajistit dostupnost výukových aplikací přes internetový prohlížeč;
- jako konkrétní výsledek vytvořit internetovou výukovou aplikaci „Atlas fyziologie a patofyziologie“ kombinující výklad a simulační hry;
- využít nové technologie tvorby simulačních modelů pro vytváření nové složitější verze komplexního simulátoru fyziologických funkcí (pracovní název „eGolem“).

4 Dosažené výsledky v technologii tvorby a využití výukových simulátorů - „od umění k průmyslu“

Zdá se, že pomalu končí doba, kdy vytváření výukových programů bylo otázkou entuziasmu a píle skupin nadšenců. Tvorba moderních výukových aplikací je náročný a komplikovaný projekt, vyžadující ***týmovou spolupráci*** řady profesí – od zkušených učitelů, jejichž scénář je základem kvalitní výukové aplikace, přes systémové analytiky, kteří jsou ve spolupráci s profesionály daného oboru odpovědni za vytvoření simulačních modelů pro výukové simulační hry, výtvarníky, kteří vytvářejí vnější vizuální podobu, až po programátory, kteří celou aplikaci „sešijí“ do výsledné podoby. Aby mezioborová spolupráce byla účinná, je zapotřebí pro každou etapu vývoje mít k dispozici řadu vývojových nástrojů a metodologií, které práci jednotlivých členů týmu usnadní a pomohou jim překonat mezioborové bariéry. Tvorba výukového softwaru tak přestává být výsledkem kreativity a pracovitosti jedinců a stále více získává rysy inženýrské práce (Kofránek, Andrlík, Kripner, & Mašek, 2002d).

Při tvorbě výukových simulátorů je třeba řešit dva typy problémů:

1. ***tvorba modelu*** (výzkumná práce, spočívající ve vytvoření formalizovaného popisu fyziologické reality, odladění a verifikaci vytvořeného simulačního modelu).
2. ***tvorba vlastního simulátoru*** (vývojová práce, vyžadující skloubit nápady a zkušenosti pedagogů, vytvářející scénář výukového programu, kreativitu výtvarníků, vytvářejících multimediální komponenty propojené se simulačním modelem na pozadí a úsilí programátorů,kteří „sešijí“ výsledné dílo do konečné podoby).

Každý z těchto problémů má svou specifiku, a proto vyžaduje použít zcela odlišné vývojové nástroje.

Pro tvorbu, ladění a verifikaci simulačních modelů se využívají speciální softwarové vývojové nástroje (Kofránek, Mateják, & Privitzer, 2009a).

- Dlouhá léta jsme pro vývoj modelů využívali převážně Matlab/Simulink od firmy Mathworks. Simulink patří k ***blokově orientovaným simulačním jazykům***, které umožňují sestavovat počítačové modely z jednotlivých bloků s definovanými vstupy a výstupy, propojovat bloky do počítacích sítí a počítací sítě seskupovat do bloků vyšší hierarchické úrovně. Bloky je možné ukládat do knihoven. Z knihoven je možné tyto bloky vyjímat a vytvářet

jejich jednotlivé instance. Při tvorbě modelů v jazyce Simulink jsme možnosti hierarchické komponentové architektury důsledně využívali. Jednotlivé bloky pak představovaly jakési „simulační čipy“ skrývající před uživatelem strukturu simulační sítě, obdobně jako elektronický čip ukrývá před uživatelem propojení jednotlivých tranzistorů a dalších elektronických prvků. Uživatel se pak může zajímat pouze o chování čipu a nemusí se starat o vnitřní strukturu a algoritmus výpočtu. Pomocí „simulačních čipů“ lze snadněji testovat chování modelu a zejména přehledněji vyjádřit vzájemné závislostí mezi proměnnými modelovaného systému. Celý složitý model pak můžeme zobrazit jako propojené simulační čipy a ze struktury jejich propojení je jasné, jaké vlivy a jakým způsobem se v modelu uvažují. To je velmi výhodné pro mezioborovou spolupráci – zejména v hraničních oblastech, jako je např. modelování biomedicínských systémů. Experimentální fyziolog nemusí dopodrobna zkoumat, jaké matematické vztahy jsou ukryty „uvnitř“ simulačního čipu, z propojení jednotlivých simulačních čipů mezi sebou však pochopí strukturu modelu a jeho chování si může ověřit v příslušném simulačním vizualizačním prostředí. Proto hierarchické blokově orientované simulační nástroje našly svoje velké uplatnění při popisu složitých regulačních systémů, s nimiž se setkáváme ve fyziologii (Kofránek, Andrlík, Kripner, & Mašek, 2002b). My jsme ve vývojovém prostředí Simulink ze simulačních čipů postupně vytvořili knihovnu ***Physiolibrary*** určenou pro modelování fyziologických regulací. Knihovna obsahuje i dokumentaci zaintegrovanou do nápovědy k prostředí Matlab/Simulink. Knihovnu i příslušný instalátor lze bez omezení stáhnout z adresy http://physiome.cz/simchips. Blokově orientované simulační nástroje mají výhodu v tom, že umožňují přehledně strukturovat model do propojených hierarchicky uspořádaných komponent. Ze struktury bokově orientovaného popisu je pak zřejmé, jakým způsobem se v modelu počítají hodnoty jednotlivých proměnných – tj. jaký je algoritmus výpočtu. Hovoříme proto o tzv. ***kauzálním modelování***. Propojování bloků do sítě vztahů ale bohužel nemůže být zcela libovolné. V propojených prvcích se nesmějí vytvářet algebraické smyčky – tj. cyklické struktury, kdy nějaká vstupní hodnota přiváděná jako vstup do výpočetního bloku ve stejném časovém kroku závisí (přes několik prostředníků) na výstupní hodnotě z tohoto bloku. Požadavek pevně zadaného směru spojení od vstupů k výstupům s vyloučením algebraických smyček vede i k náročnější stavbě modelu. Propojení bloků proto odráží spíše postup výpočtu než vlastní strukturu modelované reality.

- V poslední době došlo k vývoji nových tzv. „akauzálních“ nástrojů pro tvorbu simulačních modelů. Zásadní inovaci, kterou akauzální modelovací nástroje přinášejí, je možnost popisovat jednotlivé části modelu přímo ***jako soustavu rovnic a nikoli jako algoritmus řešení těchto rovnic***. Zápis modelů je deklarativní (popisujeme strukturu a matematické vztahy, nikoli algoritmus výpočtu) – zápis je tedy akauzální. Akauzální modelovací nástroje pracují s propojenými komponentami, které představují instance tříd, v nichž jsou přímo definovány rovnice. Komponenty se mohou propojovat pomocí speciálních akauzálních konektorů – přes akauzální propojení se vlastně propojují jednotlivé proměnné v rovnicích příslušných komponent a definují tím soustavy rovnic. Akauzální propojení rovnic je možné kombinovat i s kauzálními (řídícími) vztahy. Moderním simulačním jazykem, který je přímo postaven na akauzálním zápisu modelů je ***Modelica***. Na rozdíl od blokově orientovaného simulačního prostředí Simulink, struktura modelů v Modelice mnohem lépe zobrazuje fyzikální podstatu modelované reality (o algoritmus řešení výsledné soustavy algebrodiferenciálních rovnic se pak stará příslušný kompilátor). Modely v Modelice jsou, v porovnání s modely v Simulinku, přehlednější a samodokumentující. Modelica je prozatím využívána převážně v průmyslu. Podstatným způsobem ulehčuje modelování zejména rozsáhlých a komplexních systémů, k nimž ale také patří biomedicínské systémy. Proto jsme jako nový implementační prostředek pro tvorbu modelů pro výukové simulátory zvolili Modelicu a upustili od vývoje modelů v blokově orientovaném prostředí Simulink/Matlab. Výhody akauzálního přístupu jsou demonstrovány na příkladě implementace rozsáhlého modelu fyziologických regulací, který je podkladem připravovaného simulátoru „eGolem“. Model vychází z původního modelu, který byl podkladem simulátoru „Golem“ a z mode-

lu QHP, který modifikuje a rozšiřuje zejména v oblasti modelování acidobazické rovnováhy a přenosu krevních plynů (Kofránek, Mateják, & Privitzer, 2009b).

Vlastní ***tvorba výukových simulátorů*** je vývojová práce multioborového tvůrčího týmu, využívajícího různé softwarové vývojové nástroje (Kofránek J. , 2009).

- Kostrou výukové simulační aplikace je kvalitní scénář, vytvořený zkušeným pedagogem. Čím složitější je výukový simulátor, tím důležitější je mít předem jasnou představu o scénáři výukových simulačních her, které s ním budeme provádět.
- Pro uživatelské rozhraní simulátoru je z didaktického hlediska vhodné využívat interaktivní animace, které budou v simulátoru řízeny simulačním modelem na pozadí. Grafický vzhled v nezanedbatelné míře rozhoduje o tom, jak bude výuková aplikace přijímána svými potenciálními uživateli. Pro profesionální vzhled aplikace je ale nezbytné, aby vlastní animace vytvářel výtvarník. Proto jsme museli věnovat určité úsilí výuce výtvarníků a začali spolupracovat s výtvarnou školou Václava Hollara. Na této škole jsme otevřeli „Laboratoř interaktivní grafiky“ jako detašované pracoviště Univerzity Karlovy. Iniciovali jsme také založení Vyšší odborné školy, která vyučuje v tříletém studiu předmět „interaktivní grafika“ Grafické animace jsou vytvářeny v prostředí Adobe Flash a nyní i v prostředí Microsoft Blend.
- Vývoj vlastního výukového simulátoru je náročná programátorská práce, která vyžaduje skloubit simulační jádro s uživatelským rozhraním včetně vytvořených animací. Jako vývojovou platformu jsme dříve používali vývojové prostředí Control Web. Nyní používáme vývojové prostředí Microsoft Visual Studio pro platformu Microsoft .NET a pro jednoduché simulátory též programovací jazyk Action Script pro prostředí Adobe Flash. Simulační model, který byl nejprve vytvořen v některém z vývojových nástrojů pro tvorbu simulačních modelů (v Simulinku a nyní v Modelice), je nutné přetvořit do podoby počítačového programu ve zvoleném programovacím jazyce (např. C# aj.). Simulační jádro je možné naprogramovat i „ručně“, ale u složitějších modelů se vyplatí, pokud máme k dispozici některý nástroj, který tuto činnost zautomatizuje. Proto byl vytvořen speciální nástroj, usnadňující automatický převod vytvořeného simulačního modelu z prostředí Matlab/Simulink do prostředí Control Web a do prostředí Microsoft .NET (viz obr. 93 na straně 102). Pro převod simulačního modelu, vytvořeném v jazyce Modelica je v testovací verzi náš nový nástroj „Modelica .NET“ automaticky generující zdrojový text simulačního jádra simulátoru v jazyce C# (viz obr. 94 na straně 103).
- Pro návrh vnitřní logiky aplikace používáme hierarchické stavové automaty (jejichž pomocí je možno zapamatovat příslušný kontext modelu a kontext uživatelského rozhraní). Vyvinuli jsme také vizuální prostředí (Statecharts editor) umožňující graficky automaty navrhnout, vygenerovat jejich kód a také je ladit.
- Výukové simulátory, simulační hry a výkladové kapitoly jsou řešeny jako webové aplikace. Pro propojení simulátorů s výkladovými kapitolami jsme využívali prostředí Adobe Flash a vývojové prostředí Adobe Connect. Ozvučené interaktivní přednášky jsou provázeny animovanými obrázky synchronizovanými s výkladem, který lze v libovolném okamžiku přerušit, nebo posunout dopředu či dozadu. Každá animace je přesně synchronizovaná s textem. Pro vytvoření této synchronizace jsme si vytvořili speciální softwarový nástroj. Jednoduché simulátory jsou řešeny jako flashové aplikace (se simulačním jádrem naprogramovaným v jazyce ActionScript). Složitější simulátory vytvořené v platformě .NET jsou instalovány z webového rozhraní výukového programu na kliknutí.
- Z hlediska pedagogického efektu je výhodné, když máme možnost spouštět a ovládat i složité simulátory přímo z internetového prohlížeče. Proto jsme v poslední době začali využívat platformu Microsoft Silverlight, která umožňuje distribuovat simulátory, které mohou být spuštěny přímo v chráněném prostředí internetového prohlížeče (a to i na počítačích s různými operačními systémy, pokud mají v prohlížeči nainstalován příslušný plugin). Animace pro platformu Silverlight jsou vytvářeny v prostředí Microsoft Blend, simulační jádro je vytvářeno ve vývojovém prostředí pro platformu .NET. Pro větší efektivitu tvorby grafické vrstvy a zjednodušení spolupráce výtvarníka s programátorem, jsme vyvinuly pomocný

softwarový nástroj Animtester, s jehož pomocí mohou designéři – výtvarníci vytvářet a ladit animace jako „loutky“ ovládané změnou určitých vlastností, které se nastavovali pomocí táhlíček. Takto vytvořené „loutky“ je pak možné přímo napojit na výstupy modelů a není potřeba přidávat další programovou mezivrstvu pro propagaci dat jako tomu bylo při využívání flashových animací.

Konkrétním výsledkem je internetem dostupný „Atlas fyziologie a patofyziologie“ (http://www.physiome.cz/atlas) koncipovaný jako multimediální výuková pomůcka, která názornou cestou prostřednictvím Internetu s využitím simulačních modelů pomáhá vysvětlit funkci jednotlivých fyziologických systémů, příčiny a projevy jejich poruch. Atlas je webová aplikace, kombinující výkladové kapitoly se simulačními hrami, která je postupně rozšiřována (a na základě výsledků s jejím využitím v praktické výuce lékařů i modifikována). Je využívána ve výukové praxi na českých a slovenských lékařských fakultách jako součást výukových materiálů šířených v síti MEFANET, která zpřístupňuje elektronické výukové materiály 13 fakult v ČR a SR (viz http://www.mefanet.cz/).

Dalším výsledkem je komplexní model fyziologických regulací, implementovaný v akauzálním prostředí jazyka Modelica, který je základem vytvářeného výukového simulátoru „eGolem“.

5 Trendy a směry dalšího vývoje

Pokrok v informačních technologiích je velmi rychlý. Objevují se nové technologie, které posouvají možnosti jak v nástrojích pro modelování rozsáhlých systémů, tak i prostředků pro implementaci výukových simulátorů. V nedávné době jsme změnili technologii pro tvorbu simulačních modelů - od blokově orientovaného nástroje Matlab/Simulink jsme přešli k na novou platformu postavenou na jazyce Modelica. Jedním z důvodů našeho rozhodnutí je i to, že Modelica není firemní standard jako Simulink ale programovací jazyk pro modelování. V současné době jsou na trhu dvě komerční implemetace tohoto jazyka a zároveň je úsilím konsorcia 12 firem a 9 univerzit společně vyvíjeno prostředí Open Modelica, šiřitelné jako open source (Open Modelica Source Consortium - viz http://www.ida.liu.se/labs/pelab/modelica/OpenSourceModelicaConsortium.html). Tohoto vývoje se účastní i náš vývojový tým (v rámci firmy Creative Connections s.r.o., která je členem tohoto konsorcia (viz http://www.creativeconnections.cz/). V rámci tohoto konsorcia vyvíjíme nástroj, umožňující z modelu vyvinutého a odladěného v Modelice vygenerovat zdrojový text modelu v jazyce C#. To nám umožní model spouštět v prostředí Siverlight a vytvářet výukové simulátory spustitelné přímo v internetovém prohlížeči. Tuto možnost využijeme především pro provozování složitých simulátorů, zejména simulátoru „eGolem“ (vyvíjeného v rámci projektu MŠMT č. 2C06031, veškerá průběžná dokumentace o řešení tohoto projektu je dostupná na adrese http://patf-biokyb.lf1.cuni.cz/wiki/projekty/e-golem).

Naším budoucím cílem v rámci projektu Open Modelica je také vytvoření editoru jazyka Modelica v prostředí Siverlight a kompilátoru jazyka Modelica (do prostředí .NET) na serveru (viz obr. 110 na straně 119). To umožní vytvářet modely v akauzáním prostředí Modeliky v prohlížeči, překládat jej na serveru a spouštět výsledný model opět v prohlížeči (v prostředí Silverlight). Chtěli bychom tak vybudovat internetem dostupné úložiště biomedicínských modelů implementovaných v akauzálním jazyce Modelica pro mezinárodní projekt Physiome - prozatím v rámci tohoto projektu existují úložiště modelů pouze v kauzálních blokově orientovaných jazycích.

Biomedicínské simulátory nemusí být provozovány jenom na obrazovce počítače. Stále větší pedagogický význam budou mít simulátory pro lékařské rozhodování, které využívají robotizovanou figurínu pacienta propojenou se simulačním modelem. To je i strategický směr našeho dalšího výzkumu (ve spolupráci dvou vývojových firem, Creative Connections s.r.o. a Inomech s.r.o.), kde uplatníme vytvořený rozsáhlý simulační model pro simulátor „eGolem“.

Dalším směrem naší práce bude také další rozšíření a rozvoj výukového Atlasu fyziologie a patofyziologie v rámci mezinárodní spolupráce.

6 Závěr

Autor se léta zabývá problematikou formalizace popisu fyziologických regulací prostřednictvím simulačních modelů. Aby výsledky této teoretické práce nebyly jen obsahem vědeckých publikací, ale dostaly se tam, kde mohou být velmi užitečné, tj. ke studentům medicíny a klinickým lékařům, rozhodl se věnovat vytváření výukových pomůcek, které by pomocí simulačních her usnadnily pochopení dynamických souvislostí ve zdravém a nemocném organismu. Založil Oddělení biokybernetiky a počítačové podpory výuky na Ústavu patologické fyziologie 1. LF UK http://www.physiome.cz/wiki) a spolupracující vývojovou firmu Creative Connections s.r.o. http://www.creativeconnections.cz/).

Vytvořil multidiscilinární tým zaměřený na teoretický výzkum fyziologických systémů pomocí matematických modelů, využití modelování pro interpretaci výsledků experimentálního výzkumu, vyhodnocování klinickofyziologických dat a na praktické uplatnění modelů v lékařských simulátorech. Ve výuce medicíny se využívání simulátorů stále více prosazuje a o poslední výsledky autora v této oblasti projevila zájem Americká fyziologická společnost, která se rozhodla některé výsledky zahrnout do „teaching resources“.

Konkrétním přínosem práce je:

- odstranění chyb a následná implementace klasického Guytonova schématu z roku 1972 v prostředí Simulink;
- vytvoření nástroje QHPView pro vizualizaci matematických vztahů v modelu Quantitative Human Physiology;
- vytvoření technologie tvorby výukových simulátorů podporujících mezioborovou spolupráci včetně vytvoření příslušných softwarových propojovacích nástrojů;
- vznik multioborového týmu pedagogů, tvůrců simulačních modelů, výtvarníků a programátorů;
- vytvoření simulátoru Golem, využívaného ve výuce na lékařských fakultách u nás i v zahraničí;
- vytvoření internetového Atlasu fyziologie a patofyziologie, využívaného na lékařských fakultách v ČR a SR, nyní se připravuje začlenění anglicky lokalizovaných součástí atlasu do „teaching resources“ Americké fyziologické společnosti;
- vytvoření knihovny fyziologických modelů Physiolibrary pro prostředí Simulink;
- vytvoření komplexního modelu fyziologických regulací v akauzálním prostředí jazyka Modelica.

1 Úvod

Tato práce se věnuje technologii tvorby lékařských výukových simulátorů a multimediálních výukových programů se simulačními hrami. Je výsledkem mých mnohaletých zkušeností a společné práce s mými doktorandy v této oblasti.

Dlouhodobě se zabývám tvorbou matematických modelů fyziologických systémů. Tématem mé disertační práce byl model acidobazické rovnováhy krve (Kofránek J. , 1980). Rozšířením modelu acidobazické rovnováhy krve byl rozsáhlý model regulace vnitřního prostředí Kofránek, Pokorný, Wünsch, Brelidze, Gondžilašvili, & Verigo, 1982a-c; Kofránek, Brelidze, & Gondžilašvili, 1984; Gondžilašvili, Kofránek, Pokorný, & Brelidze, 1987), zahrnující také regulační vliv cirkulace, respirace, ledvin včetně neurohormonálního řízení. V osmdesátých letech jsem spolupracoval na tvorbě rozsáhlých modelů fyziologických regulací v rámci tehdejšího sovětského kosmického výzkumu (Verigo, 1987), kde modely fyziologických funkcí sloužily pro analýzu a predikci chování lidského organismu v prostředí změn gravitace.

Rovněž jsem se zabýval i problematikou využití simulačních modelů pro vyhodnocování klinicko-fyziologických dat (Kofránek, a další, 1988), využitím modelů pro vysvětlení mechanismů fyziologických regulací (např. Maršálek & Kofránek, 2005; Kofránek, Matoušek, & Andrlík, 2007).

V posledních patnácti letech, kdy pokrok výpočetní techniky umožnil využívat osobní počítač jako výukový prostředek, jsem se věnoval perspektivní oblasti praktického uplatnění modelů fyziologických systémů v lékařské výuce i návazných technologiím tvorby výukových simulátorů a multimediálních výukových aplikací využívajících simulační hry.

Během této doby se značně posouvaly možnosti, jak v oblasti metod modelování fyziologických systémů, tak i prostředků umožňujících vytvářet interaktivní multimedimediální animace, řízené modelem na pozadí, vytvářet výukové programy distribuovatelné přes internet a v poslední době dokonce i výukové simulátory spustitelné přímo v internetovém prohlížeči.

Efektivní vytváření moderních výukových lékařských programů se simulačními hrami vyžaduje mezioborovou spolupráci řady profesí: od systémových fyziologů, vytvářejících simulační modely fyziologických systémů, pedagogů, zodpovědných za scénář výukové aplikace, výtvarníků tvořících interaktivní animované obrázky a navrhujících vnější grafický vzhled celé aplikace, až po informatiky a programátory, kteří celé dílo integrují do výsledné podoby spustitelné prostřednictví internetu na počítači.

Klíčovou je dobrá úroveň mezioborová komunikace a využívání takových nástrojů, které tuto mezioborovou komunikaci podporují.

Vybudoval jsem proto mezioborový tvůrčí tým na Oddělení biokybernetiky a počítačové podpory výuky Ústavu patologické fyziologie 1. LF UK. Jsem školitelem doktorandů (absolventů lékařské fakulty, matematicko fyzikální fakulty UK a elektrotechnické fakulty ČVUT) v oborech Biomedicínská informatika a Fyziologie a patofyziologie člověka. Přednáším na 1. lékařské fakultě UK a na Fakultě elektrotechnické ČVUT. Zároveň se věnuji vzdělávání výtvarníků na Střední umělecké a vyšší odborné škole Václava Hollara, kde učím předmět „Ovládání interaktivity“.

Problematika tvorby lékařských simulátorů zdaleka není jen technologický problém. Jádrem lékařských simulátorů je simulační model, jehož teoretickým pozadím je odpovídající popis fyziologických systémů ve formalizovaném tvaru. Tvorba výukových simulátorů je proto také výzkumným problémem souvisejícím s širší problematikou integrativní fyziologie, zkoumající fyziologické regulace pomocí formálních nástrojů matematiky a kybernetiky.

2 Východisko práce a současný stav řešené problematiky

2.1 Pavučina fyziologických regulací

V roce 1972 v renomovaném odborném lékařském časopise Annual Review of Physiology byl publikován článek (Guyton, Coleman & Grander, 1972), který se svou podobou na již první pohled naprosto vymykal navyklé podobě fyziologických článků té doby. Jeho podstatnou část tvořilo rozsáhlé schéma na vlepené příloze. Schéma plné čar a propojených prvků na první pohled vzdáleně připomínalo nákres nějakého elektronického zařízení (obr. 1). Avšak místo odporů, kondenzátorů, cívek, tranzistorů či jiných elektrotechnických součástek zde byly zobrazeny propojené výpočetní bloky (násobičky, děličky, sumátory, integrátory, funkční bloky), které symbolizovaly matematické operace prováděné s fyziologickými veličinami (obr. 2).

Svazky propojovacích vodičů mezi bloky na první pohled vyjadřovaly složité zpětnovazebné propojení fyziologických veličin. Bloky byly seskupeny do osmnácti skupin, které představovaly jednotlivé propojené fyziologické subsystémy. Centrálním byl subsystém reprezentující cirkulační dynamiku – s ním byly do jednoho celku zpětnovazebně provázány ostatní bloky: od ledvin, přes tkáňové tekutiny, elektrolyty, až po autonomní nervovou regulaci a hormonální řízení zahrnující ADH, angiotenzin a aldosteron (obr. 3).

Autoři tímto tehdy naprosto novým způsobem pomocí graficky vyjádřených matematických symbolů popisovali fyziologické regulace cirkulačního systému a jeho širší fyziologické souvislosti a návaznost na ostatní subsystémy organismu – ledviny, regulaci objemové a elektrolytové rovnováhy aj. Místo vypisování soustavy matematických rovnic se v článku využívalo grafické znázornění matematických vztahů. Tato syntaxe umožnila graficky zobrazit souvislosti mezi jednotlivými fyziologickými veličinami ve formě propojených bloků reprezentujících matematické operace. Celé schéma tak představovalo formalizovaný popis fyziologických vztahů v oběhovém systému pomocí graficky vyjádřeného matematického modelu.

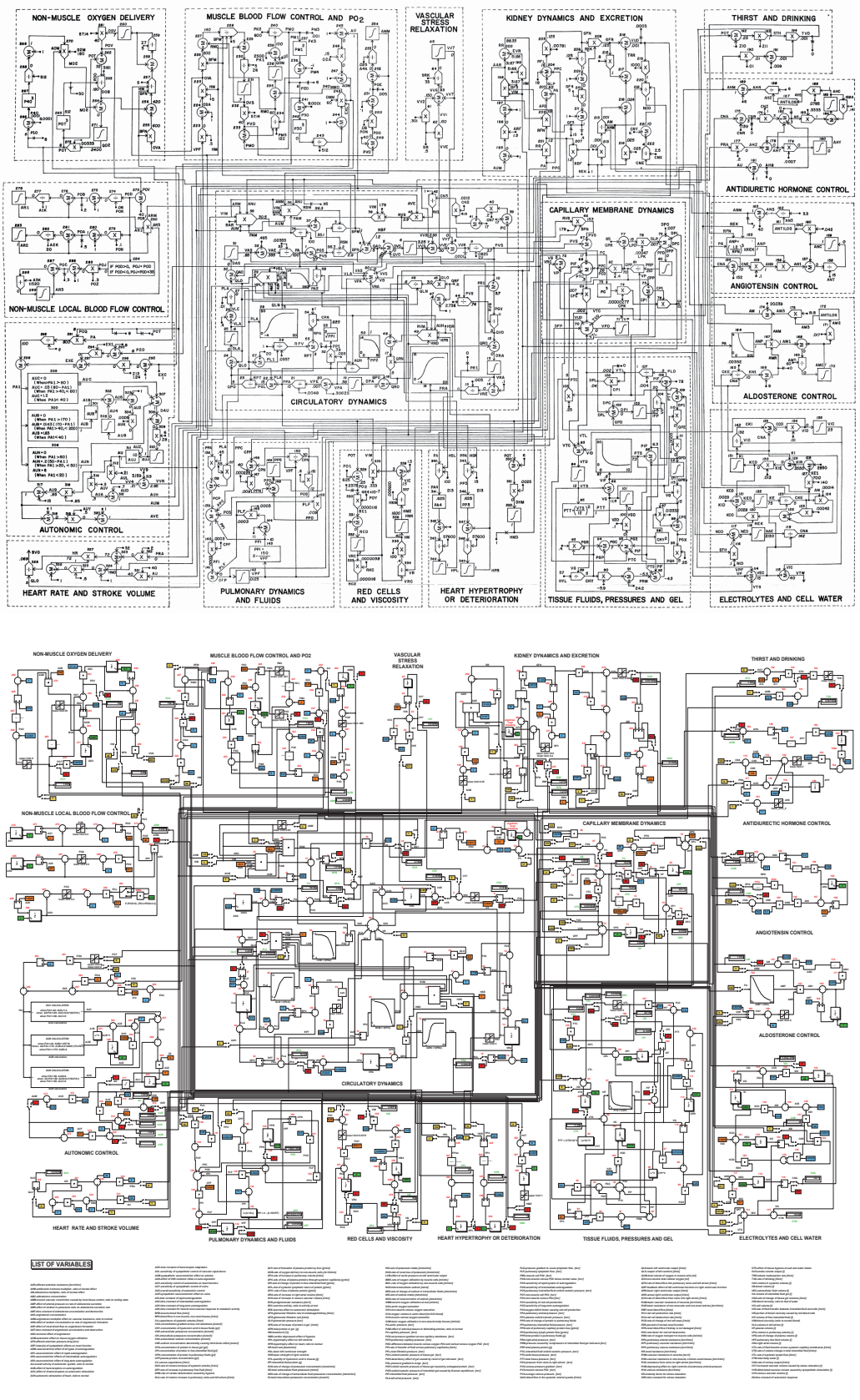
Vlastní popis modelu byl v článku reprezentován především formou základního (ale přesto plně ilustrativního) obrázku. Komentáře a zdůvodnění formulací matematických vztahů byly velmi stručné: např. „bloky 266 až 270 počítají vliv buněčného PO2, autonomní stimulace a bazální rychlosti spotřeby kyslíku tkáněmi na skutečnou rychlost spotřeby kyslíku v tkáních“. Od čtenáře to vyžadovalo nadmíru velké soustředění (a samozřejmě i solidní fyziologické a matematické znalosti) pro pochopení smyslu formalizovaných vztahů mezi fyziologickými veličinami.

O rok později, v roce 1973, vyšla monografie (Guyton, Jones & Coleman, 1973) kde byla řada použitých přístupů vysvětlena poněkud podrobněji. V roce 1975 pak Guyton se svými spolupracovníky vydal další návaznou monografii (Guyton, Taylor & Grander, 1975), kde byla podrobněji vysvětlena matematická formalizace popisu regulace oběhu a dynamiky tělních tekutin.

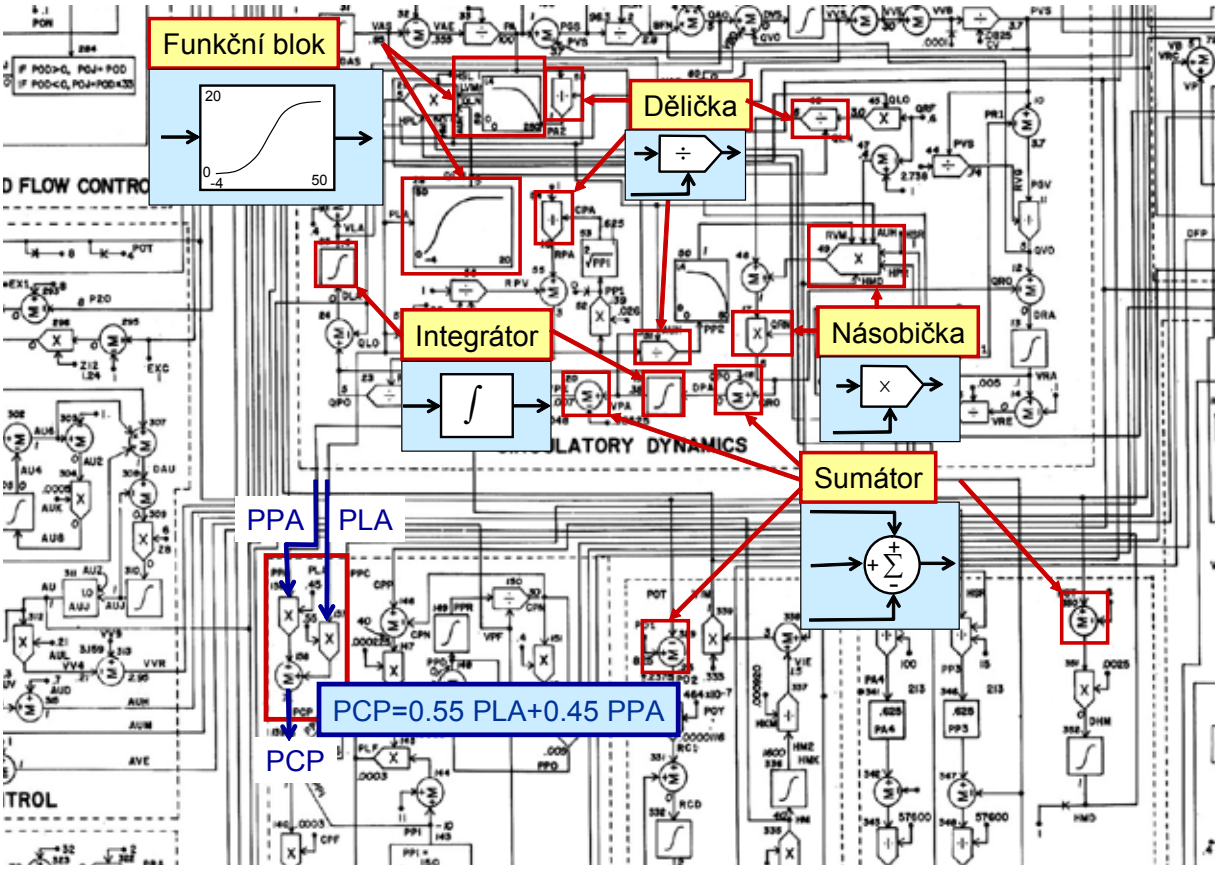
Guytonův článek, na který jsem v roce 1973 náhodou narazil ve Státní lékařské knihovně, a následné Guytonovy monografie byly také jedním z prvotních impulzů, které motivovaly mé další odborné zaměření na problematiku matematicky formalizovaného popisu fyziologické reality.

Na rozdíl od technických věd se v biologii a medicíně se s matematickým vyjádřením reality nesetkáváme často. Je třeba poznamenat, že proces formalizace, tj. převedení čistě verbálního popisu příslušné sítě vztahů na popis ve formalizovaném jazyce matematiky, je v biologických a lékařských vědách, oproti technickým vědám, fyzice či chemii opožděn. Jestliže proces formalizace ve fyzice začal již někdy v sedmnáctém století, v lékařských a biologických vědách, z důvodů složitosti a komplexnosti biologických systémů, přichází až kybernetikou a výpočetní technikou. Metodickým nástrojem jsou zde počítačové modely vytvořené na základě matematického popisu biologické reality.

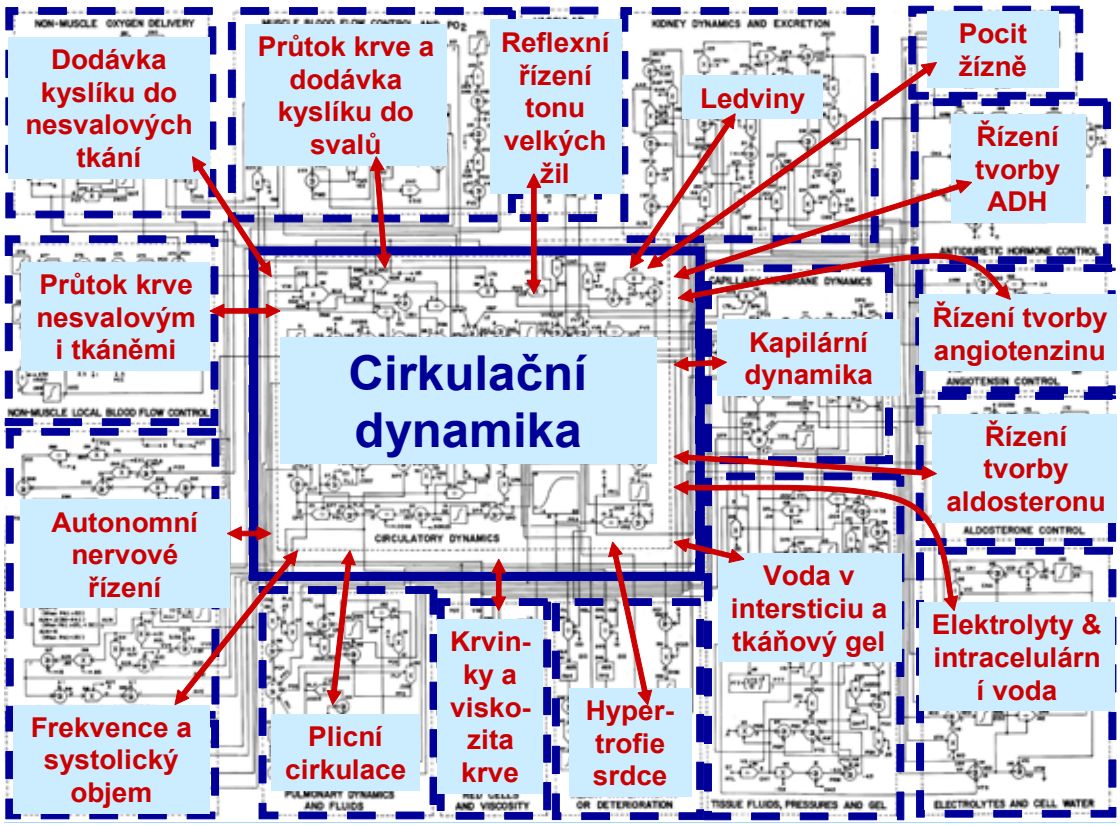
Ve fyziologii se s formalizovanými popisy setkáváme již od čtyřicátých let, kdy např. McCulloch a Pitts (McCulloch & Pitts, 1943) navrhli zjednodušený model neuronu a Sheppard (Sheppard, 1948) zavedl kompartmentový přístup, který nalezl rychlé uplatnění ve farmakokinetice. V padesátých letech Hudkgin a Huxley (Hodgkin & Huxley, 1952) publikovali svůj přelomový model vzrušivé membrány neuronu. Rozvoj počítačů v šedesátých letech vedl k další vlně publikací využívajících formalizovaný popis



Obr. 1 – Guytonův grafický diagram regulace krevního oběhu z roku 1972 (nahoře) a naše Implementace diagramu v Simulinku (dole), která zachovává rozložení prvků v původním grafickém schématu



Obr. 2 – Jednotlivé prvky na blokovém diagramu Guytonova modelu reprezentují matematické operace, propojení prvků reprezentuje rovnice v graficky vyjádřeném matematickém modelu.



Obr. 3 – Jednotlivé propojené subsystémy v Guytonově modelu.

fyziologické reality, vzpomeňme např. na Milhornovu monografii o využití teorie automatického řízení ve fyziologických systémech (Milhorn, 1966) nebo na průkopnické práce Grodinse modelující respiraci (Grodins, Buell & Bart, 1967).

Formalizovanému popisu fyziologických systémů je v současné době věnován mezinárodní projekt PHYSIOME (http://www.physiome.org), který je nástupcem projektu GENOME, jehož výsledkem byl podrobný popis lidského genomu. Cílem projektu PHYSIOME je formalizovaný popis fyziologických funkcí. Metodickým nástrojem jsou zde počítačové modely (Bassingthwaighte, 2000; Hunter, Robins, & Noble, 2002).

2.2 Vzkříšení Guytonova diagramu

Guytonův model byl jedním z prvních rozsáhlých matematických popisů fyziologických funkcí propojených subsystémů organismu a odstartoval oblast fyziologického výzkumu, která je dnes někdy popisována jako „integrativní fyziologie“ (Coleman & Summers, 1997). Obdobně jako se teoretická fyzika formálními prostředky snaží popsat fyzikální realitu a vysvětlit výsledky experimentálního výzkumu, tak se i „integrativní fyziologie“ na základě experimentálních výsledků snaží vytvořit formalizovaný popis vzájemného propojení fyziologických regulací a vysvětlit jejich funkci v rozvoji nejrůznějších onemocnění.

Z tohoto hlediska byl Guytonův model určitým mezníkem, který se snažil systémovým pohledem na fyziologické regulace zachytit dynamiku vztahů mezi regulací oběhu, ledvin, dýchání, objemu a iontového složení tělních tekutin pomocí graficky znázorněné sítě.

Guytonova grafická notace formalizovaného popisu fyziologických vztahů, inspirovaná tehdy hojně používanými analogovými počítači, představuje velmi přehledné vyjádření matematických souvztažností – bloky v uzlech sítě představují grafické symboly pro jednotlivé matematické operace a vodiče reprezentují jednotlivé proměnné. Guytonovu grafickou notaci záhy převzali i jiní autoři – např. Ikeda a spol. (Ikeda, Marumo & Shirsataka, 1979) v Japonsku nebo výzkumná skupina Amosova v Kijevě (Amosov, a další, 1977).

Grafický zápis matematického modelu prostřednictvím sítě propojených bloků byl ale v době svého vzniku pouhým obrazovým znázorněním – Guytonův model i jeho další modifikace (stejně jako i modely dalších autorů, kteří Guytonovu vyjadřovací notaci přejali) byly původně implementovány ve Fortranu a později v jazyce C++.

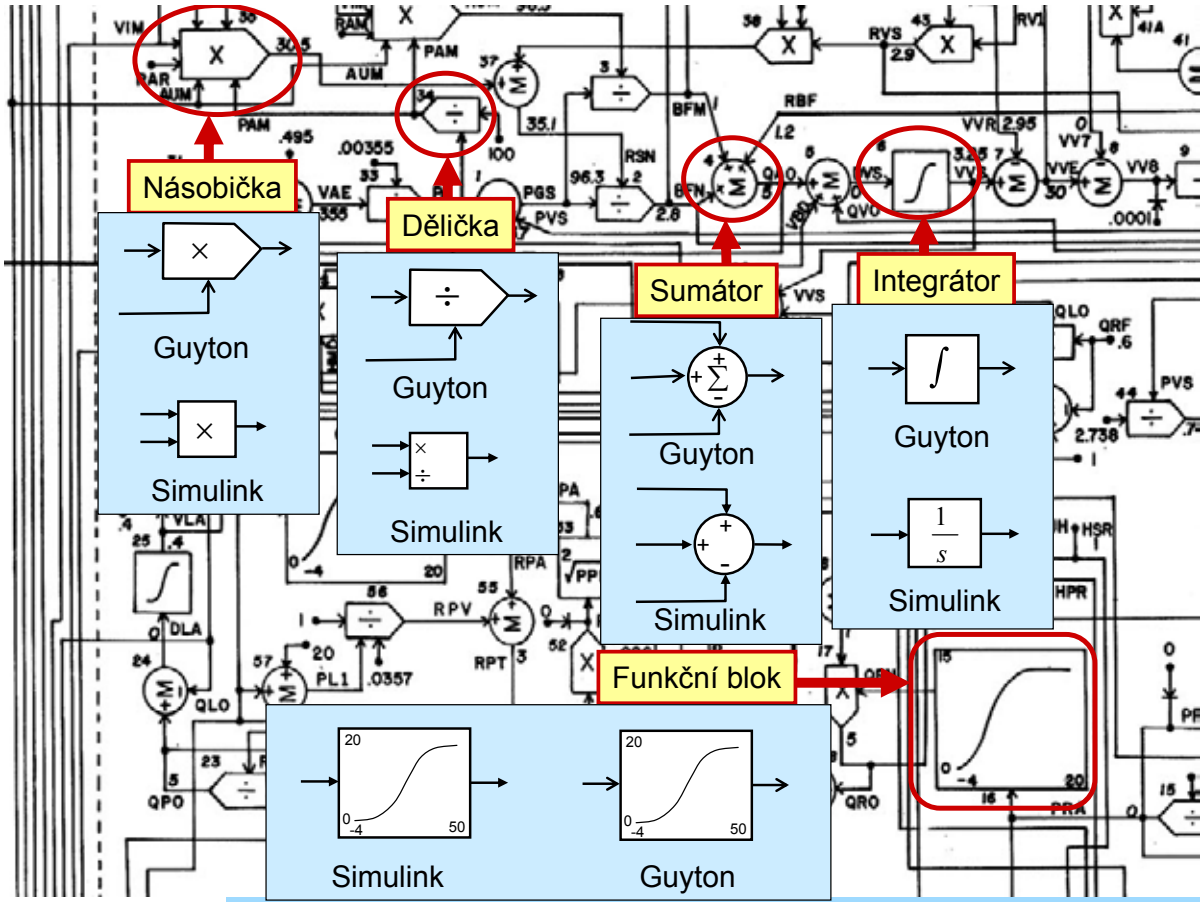
Dnes je situace jiná.

V současné době jsou pro vývoj, ladění a verifikaci simulačních modelů k dispozici specializovaná softwarová simulační prostředí, v nichž je možné vytvářet model v grafické podobě a poté i testovat jeho chování. Jedním z nich je např. široce používané vývojové prostředí Matlab/Simulink od firmy Mathworks, které umožňuje postupně sestavovat simulační model z jednotlivých komponent – jakýchsi softwarových simulačních součástek, které se pomocí počítačové myši mezi sebou propojují do simulačních sítí. Simulinkové bloky jsou velmi podobné prvkům, které pro formalizované vyjádření fyziologických vztahů použil Guyton. Rozdíl je jen v jejich grafickém tvaru (obr. 4).

Tato podobnost nás inspirovala k tomu, abychom prostřednictvím Simulinku vzkřísili starý klasický Guytonův diagram a převedli ho do podoby funkčního simulačního modelu (Kofránek & Rusz, 2007). V simulinkové implementaci modelu jsme využili i přepínače, kterými můžeme odpojovat nebo zapojovat jednotlivé subsystémy a regulační smyčky i za běhu modelu. Vnější vzhled simulinkového modelu jsme se snažili zachovat zcela stejný jako v původním grafickém schématu – rozložení, rozmístění vodičů, názvy veličin i čísla bloků jsou stejné.

Simulační vizualizace starého schématu nebyla úplně snadná – v originálním obrázkovém schématu modelu jsou totiž chyby!

V nakresleném obrázku to nevadí, pokusíme-li se ho ale oživit v Simulinku, pak model ihned zkolabuje jako celek. Chyb nebylo mnoho – přehozená znaménka, dělička místo násobičky, prohozené propojení mezi bloky, chybějící desetinná tečka u konstanty atd. Stačily však na to, aby model nefungoval. Některé chyby bylo možné vidět na první pohled (i bez znalosti fyziologie) – ze schématu je patrné, že při běhu modelu by hodnota veličin v některých integrátorech (díky špatně zakreslené zpětné vazbě)



Obr. 4 – Vzhled bloků v původní Guytonově grafické notaci a v Simulinku.

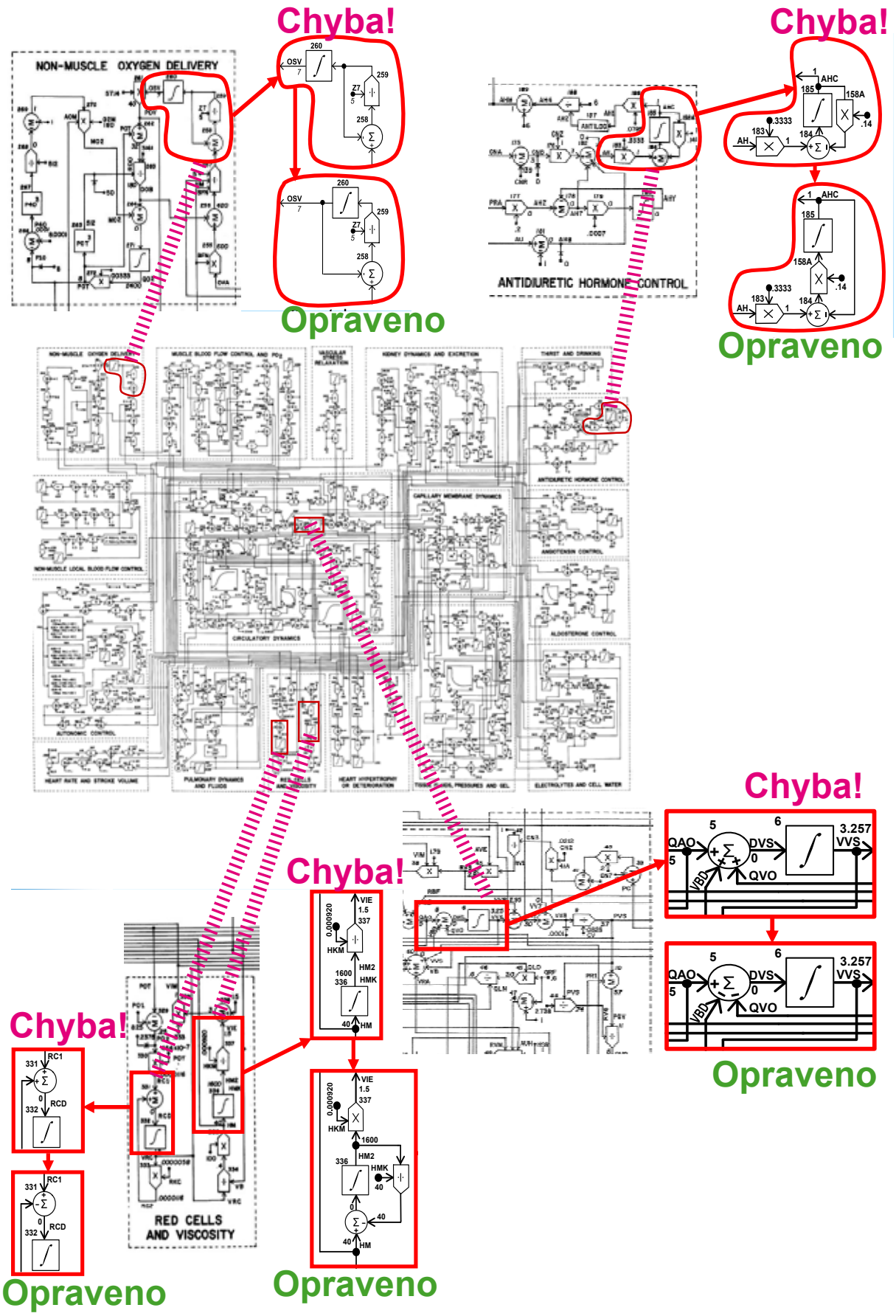
rychle vystoupala k nekonečnu a model by zkolaboval. Při znalosti fyziologie a systémové analýzy se ovšem na všechny chyby, při troše námahy, dalo přijít (obr. 5). Podrobný popis chyb a jejich oprav je v (Kofránek, Rusz & Matoušek, 2007).

Je zajímavé, že Guytonův diagram byl jako složitý obrázek mnohokrát přetiskován do nejrůznějších publikací – v poslední době se objevil např. v (Hall, 2004) a v (Van Vliet & Montani, 2005). Nikdo ale na chyby neupozornil a nedal si práci tyto chyby odstranit. To bylo pochopitelné v době, kdy obrázkové schéma vznikalo. Ještě neexistovaly kreslící programy – obrázek vznikal jako složitý výkres – a ruční překreslování složitého výkresu nebylo snadné. Možné je i to, že sami autoři modelu opravovat chyby ani příliš nechtěli – kdo si dal práci s analýzou modelu, obrazové „překlepy“ odhalil, kdo by chtěl jen tupě opisovat, měl smůlu. Konec konců, ve své době autoři rozesílali i zdrojové texty programů svého modelu v programovacím jazyce Fortran – takže pokud někdo chtěl pouze testovat chování modelu, nemusel nic programovat (maximálně pouze rutinně převedl program z Fortranu do jiného programovacího jazyka).

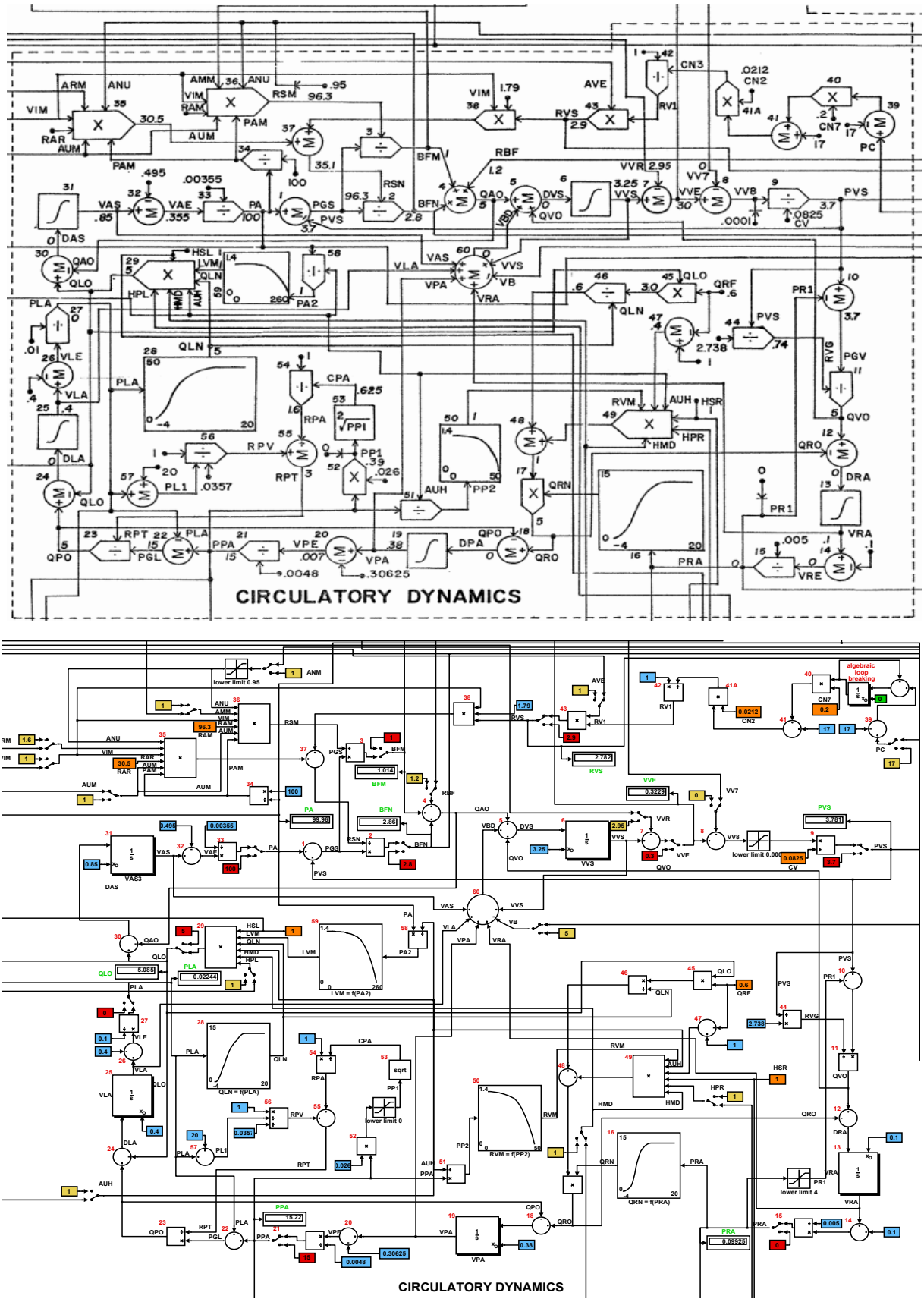
Námi vytvořená Simulinková realizace (opraveného) Guytonova modelu (viz obr. 1 a obr. 6) je zájemcům k dispozici ke stažení na adrese http://www.physiome.cz/guyton

Na této adrese je i naše Simulinková realizace mnohem složitější verze modelu Guytona a spol. z pozdějších let. Zároveň je zde i velmi podrobný popis všech použitých matematických vztahů se zdůvodněním.

Guyton a jeho žáci model nepřetržitě dále rozvíjeli (Montani, Adair, Summers, Coleman & Guyton, 1989). Guytonův model byl inspirací i podkladem pro vytvoření složitých komplexních modelů fyziologických regulací sloužících pro vysvětlení kauzálních řetězců reakcí organismu na nejrůznější podněty a i pro pochopení rozvoje různých patologických stavů. Modifikovaný Guytonův model se mimo jiné stal jedním ze základů pro rozsáhlý model fyziologických funkcí v programu „Digital Astronauts“ NASA



Obr. 5 – Oprava chyb v původním Guytonově diagramu.



Obr. 6 – Detailnější zobrazení centrální struktury Guytonova modelu v původní grafické notaci (horní část obrázku) a v simulinkové implementaci (dolní část obrázku) vyjadřující průtoky agregovanými částmi krevního řečiště a činnost srdce jako pumpy.

(White & Phee, 2006). Guytonovy modely jsou podkladem i pro rozvoj současných komplexních modelů fyziologických regulací (Thomas, a další, 2008) v rámci celoevropského projektu Virtual Physiological Human (http://www.vph-noe.eu/) .

2.3 Modely jako výuková pomůcka

Staré čínské přísloví říká *„Co slyším, zapomenu, co spatřím, to si pamatuji, co dělám, tomu rozumím“*. Tuto starou čínskou moudrost potvrzují i moderní metody učení nazývané někdy jako „learning-by-doing“, kde mají simulační hry velké uplatnění. Kromě toho, simulační hry vnášejí do výuky prvek prožitku a zároveň i jistý stupeň hravé zábavy.

Obr. 7 – Profesor Guyton se svými studenty (forografie poskytnuta laskavostí University od Mississippi, Medical Center).

Velké možnosti využívání modelů jako svébytné učební výukové pomůcky si již na počátku sedmdesátých let uvědomoval i Guyton a v rámci tehdejších možností výpočetní techniky se snažil uplatnit počítačové modely ve výuce. Guyton při výuce využíval své grafické schéma k vysvětlení základních vztahů mezi jednotlivými fyziologickými subsystémy. Pro sledování jejich chování při adaptacích na nejrůznější fyziologické i patologické podněty byl souběžně využíván model implementovaný v jazyce Fortran na číslicovém počítači (obr. 7). Guytonovo grafické schéma, implementované v prostředí Simulink, využíváme jako efektivní výukovou pomůcku i my pro výuku fyziologických regulačních systémů pro bioinženýrské specializace.

Na rozdíl od statického obrázkového schematu je ale simulinkové schéma „živé“. Studenti mohou model spouštět i zastavovat, interaktivně zadávat i měnit hodnoty jednotlivých vstupů a sledovat změny hodnot všech proměnných přímo na virtuálních displejích a osciloskopech připojených k jednotlivým prvkům grafického schématu.

Pro výuku mediků se ale takto implementovaný model příliš nehodí – medici vyžadují simulátory, jejichž uživatelské rozhraní připomíná spíše interaktivní obrázky z fyziologického atlasu, než schéma regulačních obvodů (obr. 8). To lze již dnešními prostředky realizovat (o metodách umožňujících vytvářet

Obr. 8 – Využití Guytonova modelu ve výuce bioinženýrů. Na rozdíl od původního Guytonova diagramu je jeho simulinková implementace „živé“ a interaktivní schéma. Pro výuku lékařů se ale příliš nehodí. Pro lékařskou výuku je zapotřebí vytvořit výukový simulátor, jehož uživatelské rozhraní, místo interaktivního regulačního schématu, připomíná spíše obrázky z lékařských učebnic a monografií

multimediální výukové simulátory pojednávají další kapitoly).

V době publikování Guytonova modelu, v první polovině sedmdesátých let, však bylo uplatnění počítačových modelů ve výuce medicíny výsadou jen několika univerzit a záviselo především na technickém vybavení a entuziazmu pracovníků příslušných fakult.

Na Karlově univerzitě bylo takovým pionýrským pracovištěm Biokybernetické oddělení Fyziologického ústavu Fakulty všeobecného lékařství UK v Praze, kde se v rámci fyziologie studenti seznamovali se základními pojmy teorie systémů, obecnými principy regulačních obvodů a jejich aplikacemi na fyziologické regulace (Wünsch, 1969, Wünsch, 1974). Studenti pracovali s modely implementovanými na analogových počítačích MEDA a později, když to technické vybavení pracoviště dovolilo, i na terminálech číslicových počítačů (Wünsch & Trojan, 1986).

Velkým impulzem pro rozšíření uplatnění modelů ve výuce medicíny byl nástup osobních počítačů v polovině osmdesátých let. S růstem jejich numerických schopností i možností grafického uživatelského rozhraní se objevila možnost používat počítač jako skutečnou výukovou pomůcku kombinující interaktivní multimediální rozhraní se simulačním modelem na pozadí.

V devadesátých letech technologický pokrok umožnil na personálním počítači spouštět i poměrně rozsáhlé modely fyziologických systémů. Objevily se také specializované vývojové nástroje, určené pro tvorbu simulačních modelů – např. Matlab/Simulink od firmy Mathworks. Následný rozvoj informačních a komunikačních technologií spolu s rozšířením vysokorychlostního internetu v posledních letech přinesl další možnosti uplatnění počítačů jako velice efektivní výukové pomůcky. Tyto možnosti, nabízející praktické uplatnění teoretických výzkumů formalizace fyziologických systémů ve výuce lékařů i bioinženýrů, byly také velkou motivací pro zaměření naší vlastní práce v této oblasti.

2.4 Schola ludus v moderním hávu

Obdobně, jako počítače dnes již prakticky vytlačily psací stroje z kanceláří a staly se běžným kancelářským vybavením, lze očekávat, že se počítače propojené na vysokorychlostní internet v blízké budoucnosti stanou běžnou výukovou pomůckou. Zřejmě to ale nepůjde nijak závratně rychle. Jestliže textový procesor, tabulkový kalkulátor, kreslící a prezentační program pokrývají značnou část potřeby automatizace administrativních prací, pak pro použití počítačů ve výuce je klíčovým limitujícím faktorem dostatek vhodných výukových programů.

Jejich tvorba není jednoduchá. Zdaleka nestačí jen převést skripta (případně doplněné multimediálními komponenty) do počítačem prezentovatelné podoby. Kombinace hypertextu, obrázků, zvuku, videa a interaktivních animací na jedné straně dává velké pedagogické možnosti pro názorné vysvětlení složitých problémů, na druhé straně však klade na autory výukových programů i použité technologie pro jejich tvorbu velké nároky.

V posledních letech se objevila celá plejáda výukových programů, využívajících interaktivní multimediální rozhraní pro vysvětlení složitých procesů v biomedicínských vědách ať již ve všeobecně dostupných (např. http://www.apsarchive.org) nebo komerčních (např. http://interactivephysiology.com) aplikacích. Většinou však jde o jen animace a nikoli o simulace.

Rozšíření výukových multimediálních programů o simulační hry klade na tvůrce výukových aplikací a technologii tvorby další nároky, na druhé straně však výsledné možnosti interaktivních simulačních her jsou z didaktického hlediska nepoměrně větší než využití pouhých animovaných interaktivních ilustrací.

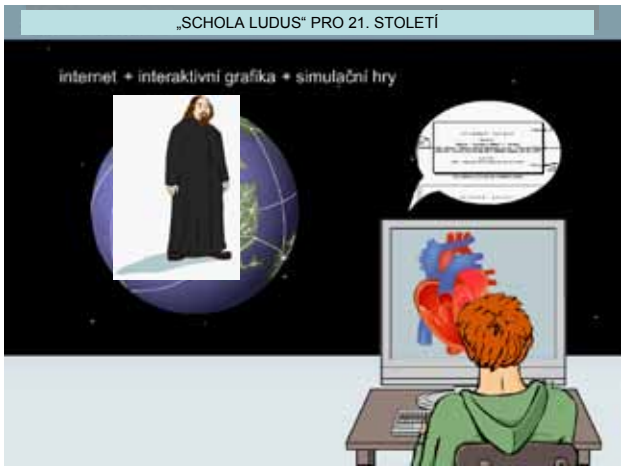
Výukové ***multimediální programy*** se ***simulačními komponentami*** nejsou jen moderní náhradou klasických učebnic. Jsou zcela ***novou výukovou pomůckou*** umožňující prostřednictvím výukových simulačních her názorně prozkoumat vykládaný problém ve virtuální realitě.

Internet, jako distribuční médium dokáže tyto nové výukové pomůcky učinit snadno dostupnými kdekoli po světě. Masové rozšíření internetu přineslo možnost snadné dostupnosti těchto moderních výukových pomůcek na pouhé kliknutí myší. Jejich vytváření však není jednoduché. Technologii jejich tvorby jsou věnovány následující kapitoly této práce.

Spojení internetu, multimediálního prostředí, sloužícího jako zvukové a vizuální uživatelské rozhra-

ní, se simulačními modely umožňuje studentům po připojení do kouzelné internetové pavučiny si prostřednictvím výukové simulační hry názorně ozřejmit dynamické vztahy mezi vykládanými pojmy.

Zapojení multimediálních výukových her do výkladu přináší zcela nové pedagogické možnosti zejména při vysvětlování složitě provázaných vztahů a pro aktivní procvičování praktických dovedností a ověřování teoretických znalostí.



Obr. 9 – Simulační hry, propojené s interaktivními multimédii ve výukových programech umožňují studentům názorně si „osahat" vykládaný problém ve virtuální realitě. Internet umožní jejich snadnou dostupnost. Staré Komenského krédo – „škola hrou" tak dnes nachází své moderní uplatnění.

Simulační hrou je možné bez rizika otestovat chování simulovaného objektu – např. zkusit přistávat virtuálním letadlem nebo, v případě lékařských simulátorů, léčit virtuálního pacienta, či testovat chování jednotlivých fyziologických subsystémů.

Právě zde nachází své moderní uplatnění staré krédo Jana Amose Komenského „Schola Ludus" – tj. „škola hrou" (Comenius, 1656), které tento evropský pedagog razil již v 17. století (obr. 9).

Výklad pomocí internetem dostupných simulačních her je častý ve fyzice či chemii, méně časté je využití simulačních her a simulátorů v oblasti medicíny, což je zřejmě dané složitostí příslušných simulačních modelů.

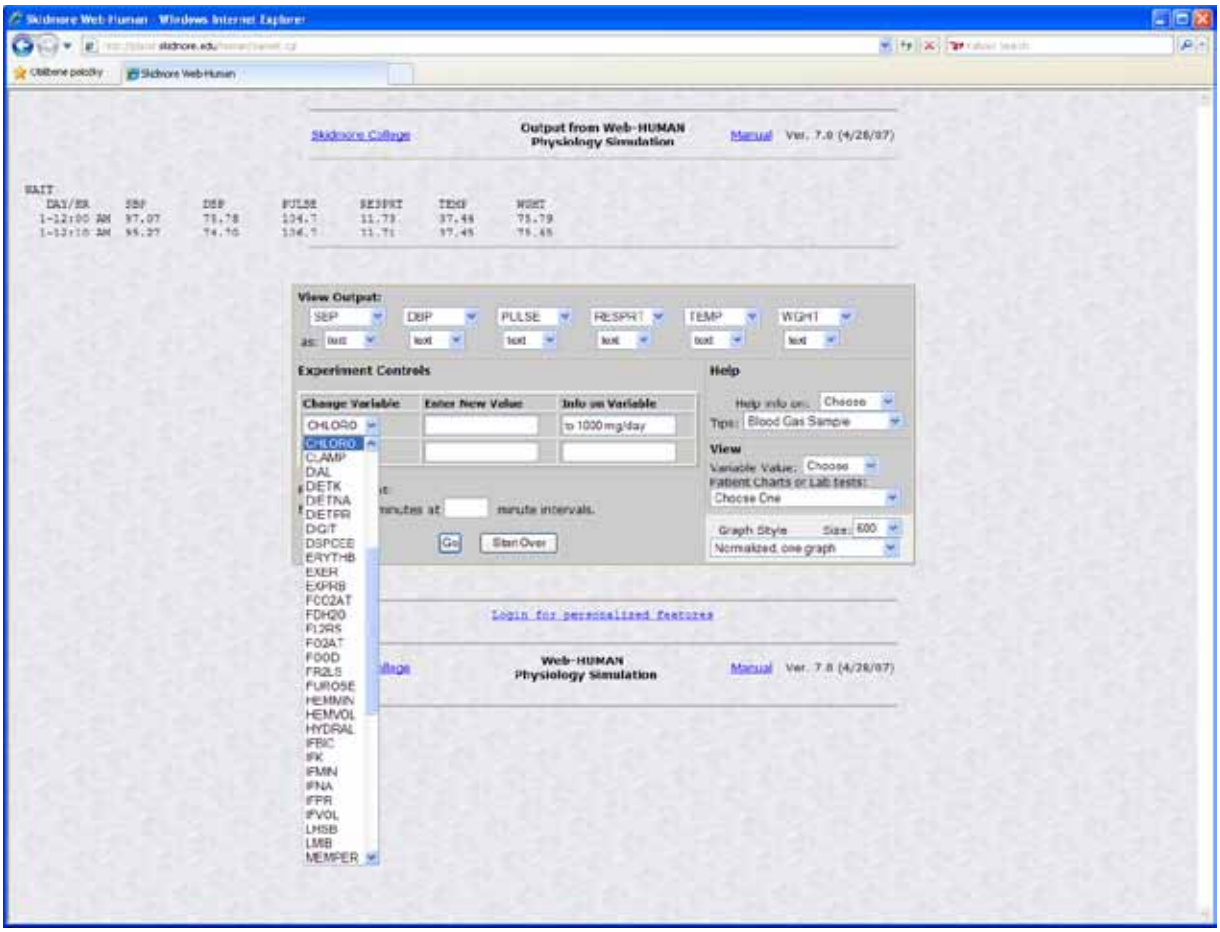
Nicméně i v oblasti medicíny se na internetu dá najít řada výukových aplikací se simulačními hrami. Na pavučině internetu je možné najít k volnému pedagogickému použití mnohé výukové ***simulátory jednotlivých fyziologických subsystémů***.

Tak například simulátor ECGsim (http://www.ecgsim.org/) umožňuje studovat tvorbu a šíření elektrického potenciálu v komorách srdce a studovat mechanismus vzniku komorového komplexu QRS za různých patologií – od poruch vedení vzruchu až po ischemie a infarkty (van Oosterom & Oostendorp, 2004). Tlakově oběhové křivky v komorách srdce při různých patologiích srdce (chlopenních vadách, levostranném či pravostranném selhání) umožňuje sledovat simulátor srdce z Columbia Univerzity (http://www.columbia.edu/itc/hs/medical/heartsim), (Burkhoff & Dickstein, 2003). Simulátory anesteziologických přístrojů z University of Florida umožňují dávat anestézii virtuálnímu pacientovi (http://vam.anest.ufl.edu/) a sledovat příslušné fyziologické odezvy (složitější simulátory ale vyžadují placený přístup). Přenos krevních plynů a acidobazické parametry jsou tématem simulátoru OSA (Oxygen Status Algorithm), určeného pro výuku i klinickou praxi (Siggaard-Andersen & Siggaard-Andersen, 1995). Poslední verze tohoto simulátoru z roku 2005 je dostupná na adrese http://www.siggaard-andersen.dk/OSA.exe. Činnost neuronu a neuronových sítí umožňuje studovat simulační program NEURON z Yale University http://www.neuron.yale.edu/ (Hines & Carnevale, 2001; Carnevale & Hines, 2006). Výukový simulátor AIDA (http://www.2aida.net/) modeluje virtuálního diabetického pacienta a umožňuje sledovat vliv dávkování různých druhů inzulinu při zadaném příjmu potravy na glukózový metabolismus (Reed & Lehmann, 2005; Lehmann, Tarin, Bondia, Teufel, & Deutsch, 2007). Softwarové simulační hry pro výuku medicíny jsou i tématem nabídky řady komerčních firem, např. firmy Anesoft (http://www.anesoft.com/), Medical Simulation Corporation (http://www.medsimulation.com) a další nabízejí řadu softwarových simulátorů pro nejrůznější oblasti medicíny.

Výukové simulátory dnes šířeny prakticky téměř výhradně prostřednictvím internetu, což, na rozdíl od dřívější distribuce prostřednictvím CD ROM, usnadňuje šíření aktualizovaných verzí. Moderním trendem je spouštění simulačních výukových programů přímo v internetovém prohlížeči.

2.5 Komplexní modely pro lékařské simulátory

Pro výuku patofyziologie a studium patogenezy nejrůznějších patologických stavů mají velký význam



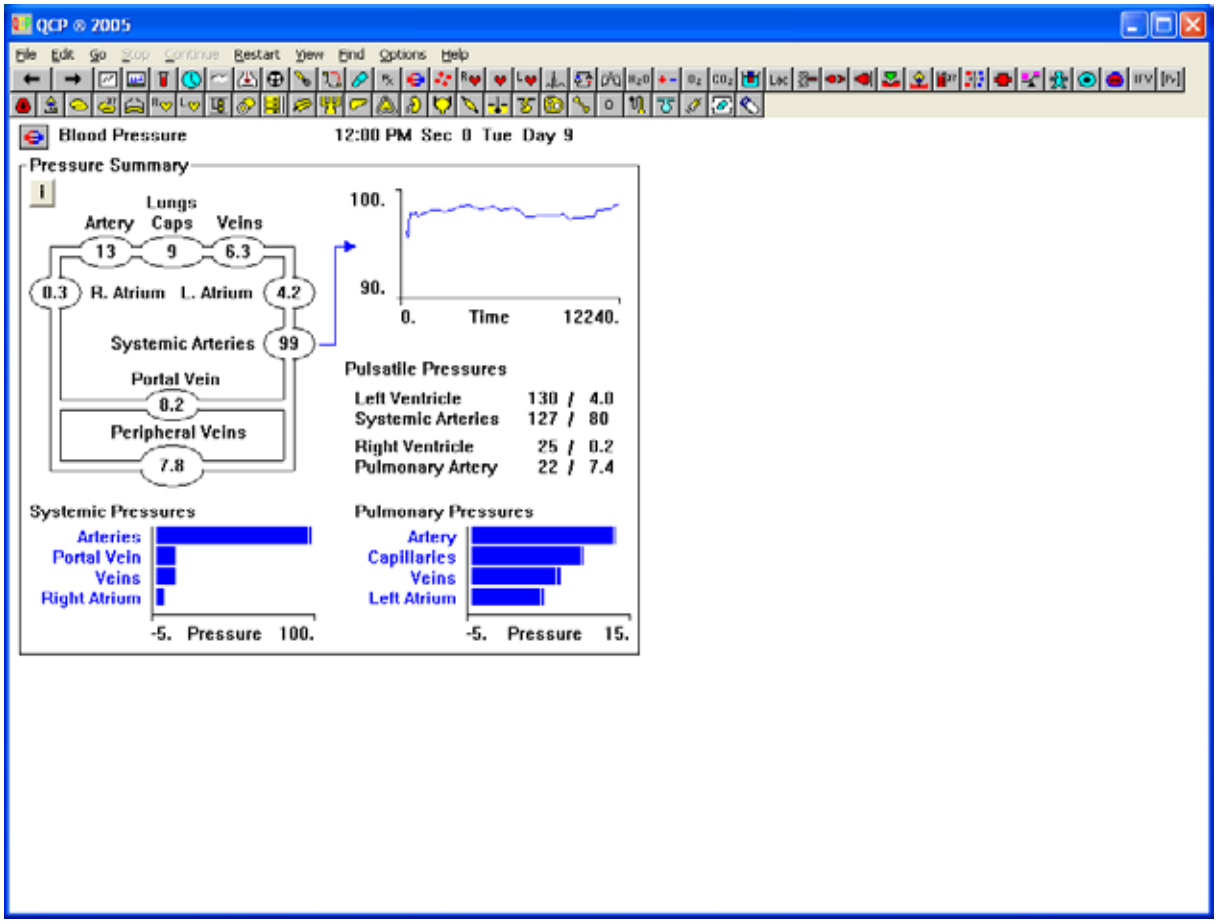
Obr. 10 – Prostředí webové implementace původního Colemanova výukového modelu Human. Jeho ovládání je poněkud těžkopádné, vyžaduje znalost významu jednotlivých proměnných modelu, jejichž názvy odpovídají názvům proměnných v implementaci modelu ve Fortranu z roku 1983.

komplexní simulátory, zahrnující ***modely nejen jednotlivých fyziologických subsystémů, ale i jejich propojení do komplexnějšího celku***.

V roce 1982 Guytonův žák a spolupracovník Thomas Coleman vytvořil model ***„Human“*** určený především k výukovým účelům (Coleman & Randall, 1983). Model umožnil simulovat řadu patologických stavů (kardiální a renální selhání, hemorhagický šok aj.) i vliv některých terapeutických zásahů (infúzní terapii, vliv některých léků, transfúzi krve, umělou plicní ventilaci, dialýzu atd.). Autoři za mírný poplatek na požádání rozesílali zdrojový text programu v jazyce Fortran. Model byl v osmdesátých letech jedním z nejrozsáhlejších simulačních výukových biomedicínských programů.

V poslední době Meyers a Doherty implementací v Javě původní Colemanův model zpřístupnili na webu (Meyers & Doherty, 2008). Model je na webu (http://placid.skidmore.edu/human/index.php) snadno ovladatelný, nicméně z původní implementace ve Fortranu si odnáší názvy proměnných (omezených na šest alfanumerických znaků), jejichž význam musí uživatel nejprve rozklíčovat (obr. 10).

Dalším, velmi podstatným, rozpracováním modelu Human je rozsáhlý výukový simulátor ***Quantitative Circulatory Physiology (QCP)*** (Abram, Hodnett, Summers, Coleman & Hester, 2007). Pro podporu jeho využívání jako výukové pomůcky v lékařské výuce ho autoři v kompilované formě volně zpřístupnili na webu University od Mississippi (http://physiology.umc.edu/themodelingworkshop/). Simulátor je možné stáhnout a instalovat na počítači v prostředí Windows (obr. 11). Na rozdíl od modelu Human, význam jednotlivých proměnných již není kryptický, nicméně proměnných je velmi mnoho (několik tisíc) a efektivní práce s modelem vyžaduje znalost složité nabídkové struktury simulátoru i představu o tom, jaké proměnné je třeba při simulacích jednotlivých patologických stavů vhodné sledovat a hodnotu jakých parametrů je nutno změnit pro simulaci nejrůznějších patologických poruch (simulátor umožňuje měnit hodnoty cca 750 parametrů, modifikujících fyziologické funkce). Hodnoty těchto parametrů je



Obr. 11 – Prostředí výukového simulátoru Quantitative Circulatory Physiology (QCP). Simulátor nabízí sledování stovek proměnných. Jeho ovládání je však složité a vyžaduje prostudovat rozsáhlou strukturu simulátoru, i dobrou znalost toho, jaké procesy je zapotřebí při simulacích určitých patologických stavů sledovat.

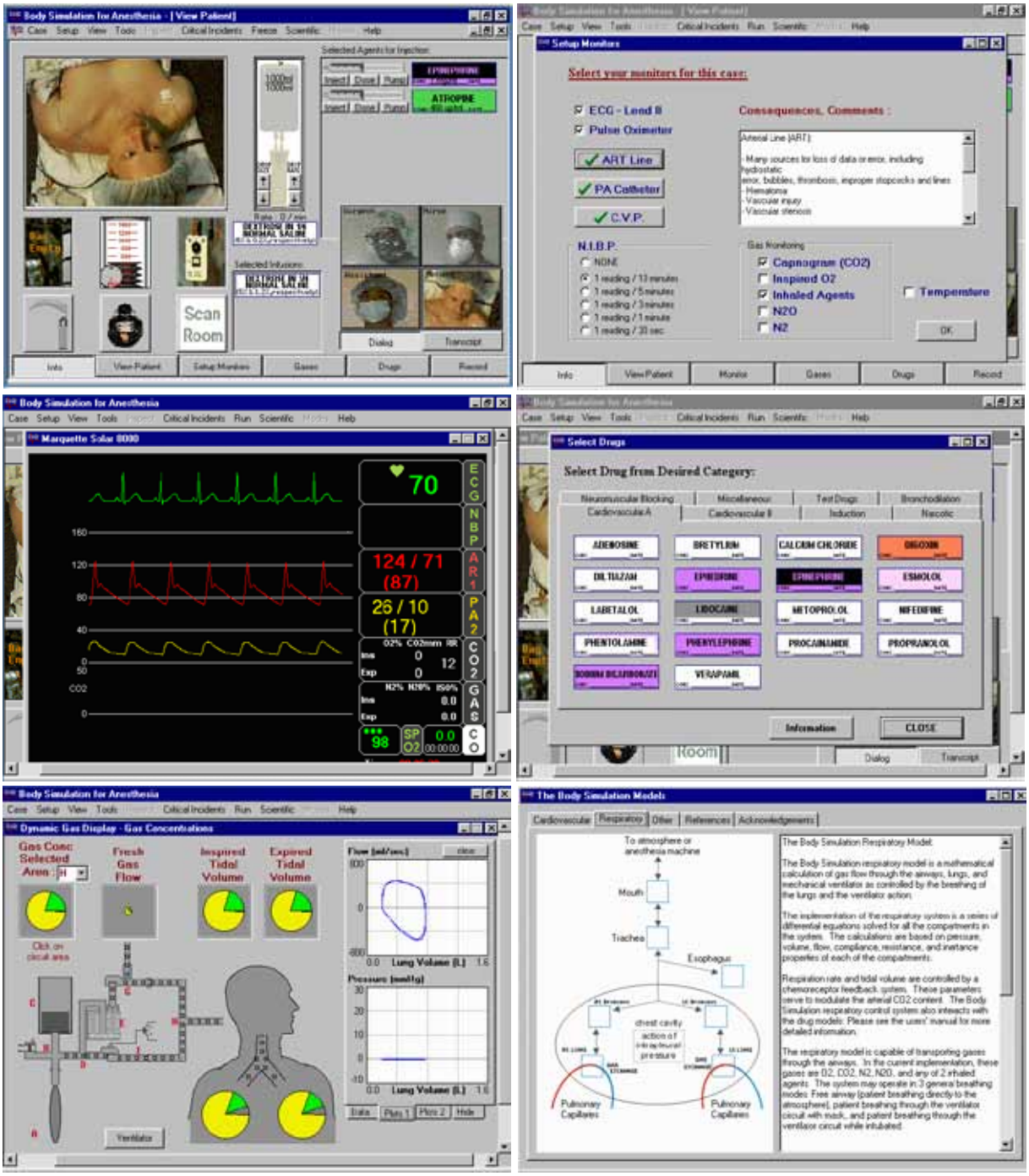
možno ukládat do externího souboru nebo z externího souboru načítat, což umožňuje připravit řadu scénářů pro různé scénáře modelovaných patologických stavů. Autoři pro výukové potřeby řadu těchto scénářů (ve formě vstupních souborů) připravili a spolu s příslušným komentářem umožnili jejich stahování z příslušné webové stránky.

Nicméně ve výukové praxi (tam, kde ho využívali) se simulátor osvědčil. Tak např. španělští autoři Rodriges-Barbero a Lopes-Novoa referují, že více než 70% studentů v anonymizovaném dotazníku uvedlo, že výuka pomocí simulátoru QCP jim umožnila lepší chápání komplexity fyziologických procesů a vzájemného koordinovaného působení několika fyziologických subsystémů v homeostatické odpovědi na nejrůznější podněty. Studenti si pochvalovali, že výukou na simulátoru získali lepší pochopení o funkcích lidského organismu bez ohledu na to, že simulátor byl pouze v angličtině a nebyl lokalizován do španělštiny (Rodriguez-Barbero & Lopez-Novoa, 2009).

2.6 Modely pro simulátory jako komerční know how

Lékařské simulátory se v poslední době staly i žádaným komerčním artiklem. Objevily se i v nabídce řady specializovaných komerčních firem.

Tak například americká společnost Advanced Simulation Corporation, vyrábějící letecké simulátory, se od roku 1993 věnuje též vývoji lékařských simulátorů (http://www.advsim.com). Pod názvem ***Body Simulation*** prodává interaktivní multimediální výukový simulátor, umožňující na personálním počítači simulovat fyziologické funkce lidského organismu v normě i za různých patologických stavů, včetně vlivu nejrůznějších farmak. ***Body CV*** je rozšířením simulátoru o podrobnější simulaci kardiovaskulárních funkcí a ***Body Simulation for Anaesthesia*** dále rozšiřuje simulátor o simulaci anesteziologických



Obr. 12 – Ukázky obrazovek z komerčního lékařského simulátoru Body Simulation od firmy Advanced Simulation Corporation.

přístrojů, ventilátorů a dalších technologických zařízení operačního sálu a lůžka intenzivní péče (viz obr. 12). Základní verze simulátoru se prodává za 699 USD, celková cena kompletního simulátoru se však vyšplhá na bezmála pět tisíc amerických dolarů (4 995 USD). Podkladem simulátoru je rozsáhlý model fyziologických funkcí, jehož tvorbu koordinoval Dr. N.T. Smith. Model vychází z řady prací, např. z (Fukui & Smith, 1981a,1981b; Smith, 1987; Smith, 1987; Masuzawa, Fukui, & Smith, 1988; Masuzawa, Fukui, & Smith, 1991; Smith & Starko, 1995; Smith, Starko, & Davidson, 1996) – jeho detailní struktura je však součástí (vzhledem k ceně simulátoru) firemního know how.

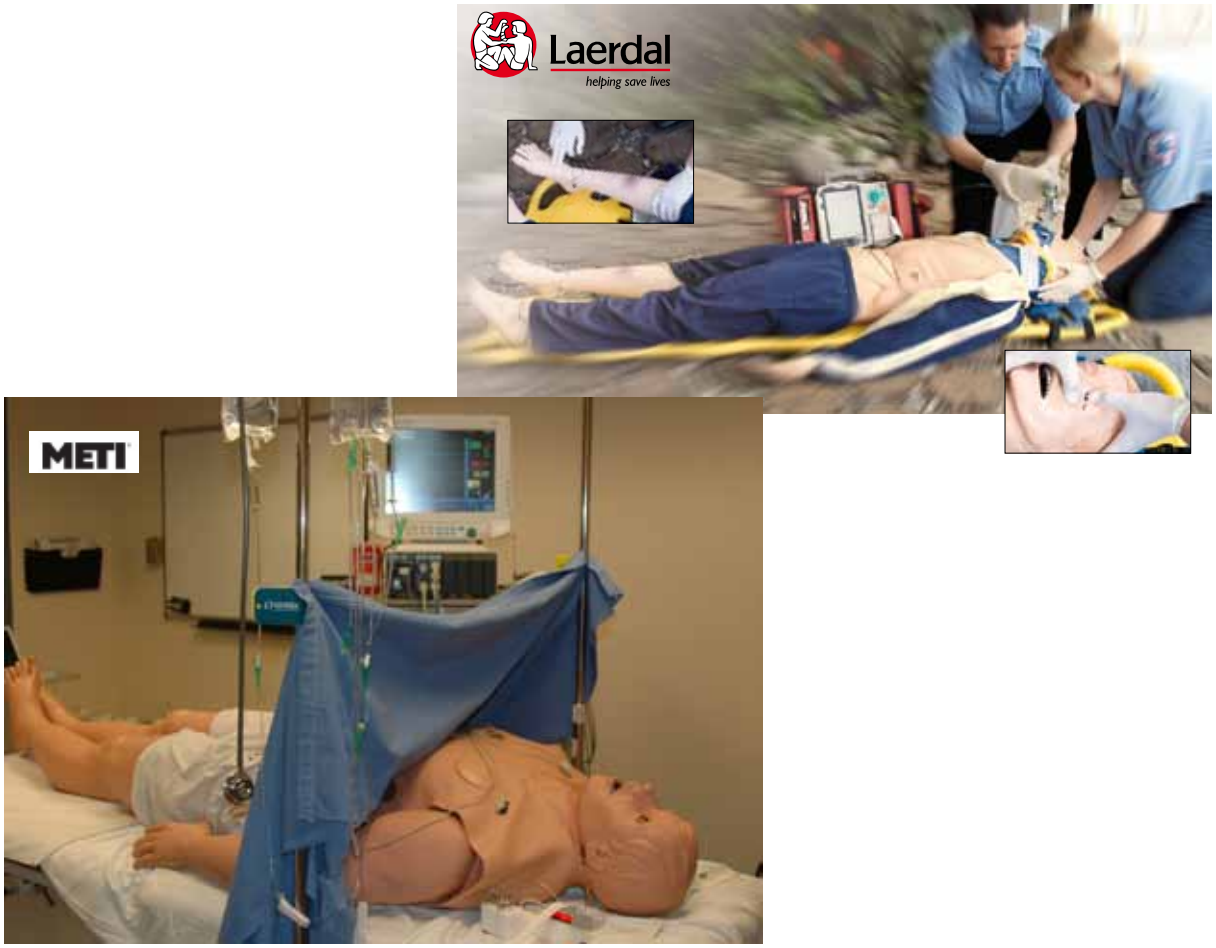
Dalším příkladem komerční firmy specializující se na multimediální lékařské simulátory je španělská firma Alfa Multimedia, která nabízí multimediální výukový simulátor MEDITECA (http://www.medicalsimulator.net). Firma při jeho vývoji spolupracuje s řadou španělských odborných lékařských spo-

lečností a univerzit. Simulátor je především určen pro nácvik rychlého lékařského rozhodování v medicíně akutních stavů. Umožňuje mimo jiné simulovat akutní stavy v imunologii (jako je imunodeficience, anafylaxe, angioedém), akutní stavy v otorhinolaryngologii, a gastroenterologii, akutní otravy řadou látek, poškození fyzikálními vlivy (barotrauma, úrazy elektrickým proudem, působení nízké teploty aj.).

Rozhraním výukových simulátorů nemusí být jen obrazovka počítače. Pokrok v technologii haptického snímání a v zobrazování virtuální reality přinesl novou třídu simulátorů určenou pro nácvik chirurgických technik (Liu, Tendick, Cleary & Kaufmann, 2003; Dunkin, Adrales, Apelgren & Mellinger, 2007; Satava, 2008; Kössi & Luostarinen, 2009).

Rozvíjí se trh lékařských simulátorů, určených k procvičování praktického provádění některých zdravotnických úkonů (kardiopulmonální resuscitace, katetrizace, endoskopie, intubace pacienta, echografické vyšetřování apod.) na figuríně pacienta. Stále více se však objevuje v nabídce i řada hardwarových trenažérů určených zároveň i k procvičování lékařského rozhodování (Hammond, Berman, Chen & Kushins, 2002; Lighthall, 2007). Jejich jádrem je komplexní model fyziologických regulací lidského organismu, propojený s hardwarovým simulátorem.

Studenti na základě výsledků simulovaných vyšetření, a v sofistikovanějších simulátorech i na základě dat z reálných lékařských monitorů, připojených k figuríně pacienta, rozhodují o terapeutickém postupu, provádějí léčbu (od připojování pacienta na ventilátor umělé plicní ventilace či anasteziologický přístroj, přes kardiopulmonální resuscitaci, až po podávání simulovaných infúzí, a simulaci podávání příslušných léků). Velký význam mají tyto trenažéry zejména pro nácvik správného lékařského rozhodování v medicíně akutních stavů. Vlastní trenažér je, obdobně jako u leteckých pilotních simulátorů, řízen ze stanoviště operátora, odkud učitel může ovládat simulovaného pacienta a volit mezi nejrůznějšími scénáři simulovaných onemocnění. Ukazuje se, že z pedagogického hlediska je velice efektivní,



Obr. 13 – Simulátory pacienta od původně norské a nyní nadnárodní firmy Laerdal (http://laerdal.com) a od americké firmy METI (http://www.meti.com). Na pozadí těchto simulátorů je matematický model. Nejefektivnější využití těchto simulátorů je zejména v oblasti urgentní medicíny.

když veškeré akce studentů jsou monitorovány a simulátor poskytuje podklady pro pozdější rozbor (debriefing) diagnostického a terapeutického postupu studentů u simulovaného onemocnění. (Clay, Que, Petrusa, Sebastian & Govert, 2007).

Sofistikované robotizované simulátory pacienta dnes vyrábějí především dvě velké společnosti (obr. 13). První je původně norská hračkářská firma Laerdal (http://www.laerdal.com/), která je dnes společností s pobočkami ve dvaadvaceti státech světa. Vyrábí sadu robotizovaných simulátorů, včetně simulátor SimBaby úpěšně využívaný jako lékařský trenažér pro péči o novorozence a kojence (Sawyer, Hara, Thompson, Chan, & Berg, 2009). Trenažéry Laerdal se osvědčily nejen ve výuce lékařů, ale i ve výuce sester (Cason, Kardong-Edgren, Cazzell, Behan, & Mancini, 2009).

Druhým úspěšným výrobcem je americká firma METI, jejíž robotizované trenažéry jsou velmi efektivní (i když nákladnou) výukovou pomůcka pro výcvik anesteziologů a zdravotnických týmů zejména v oblasti medicíny akutních stavů (Sethi, Peine, Mohammadi, & Sundaram, 2009; Ellaway, Kneebone, Lachapelle, & Topps, 2009).

Za komerčním úspěchem komerčních lékařských simulátorů stojí dobře identifikovaný simulační model na pozadí. Jeho podrobná struktura (soustava použitých rovnic a příslušné hodnoty parametrů) obvykle není zveřejňována a stává se pečlivě chráněným technologickým know-how.

2.7 Jen simulátor nestačí

Lékařské trenažéry umožňují, obdobně jako letecké trenažéry zcela nový způsob výuky, kdy si student, bez nebezpečí pro pacienta může ve virtuální realitě procvičovat diagnostické a terapeutické úkony. A na rozdíl od reálného světa, ve virtuální realitě jsou chyby vratné. Ale nejenom to, student může podrobně sledovat průběhy hodnot nejrůznějších veličin, které u reálného pacienta nejsou běžně dostupné klinickému vyšetřování. Student také může opakovaně rozpojovat i zapojovat jednotlivé regulační smyčky, sledovat odezvy jednotlivých fyziologických subsystémů na nejrůznější vstupy odděleně od jejich okolí (což v reálném světě často není možné ani v těch nesložitějších experimentech). Simulátory tak mohou sloužit jako interaktivní výuková pomůcka k vysvětlení fyziologických mechanismů fungování lidského organismu.

Pořídit si drahý simulátor k efektivní výuce samo o sobě ale nestačí.

Jak, zvláště v poslední době, upozorňuje řada autorů, výuka se simulátorem klade citelně vyšší nároky na vyučujícího, než klasická výuka.

Při správném využití simulátoru, je však pedagogický efekt velmi výrazný, zvláště v takových oblastech, kde je rychlé a správné rozhodování velmi důležité, například v medicíně akutních stavů a v anesteziologii (Binstadt, Walls, White, Nadel, Takavesu & Barker; Lammers, 2006; Day, 2006; Wayne, Didwania, Feniglass, Fudala, Barsuk & McGaghie, 2008; Rosen, 2008; Kobayashi, Patterson, Overly, Shapiro, Williams & Jay, 2008; Jones & Lorraine, 2008; McGaghie, Siddall, Mazmanian & Myers, 2009).

Před několika lety nebyl řídký jev, kdy si instituce z různých grantových i jiných prostředků si nejprve pořídí za drahé peníze nejrůznější simulační zařízení s předstihem před jasnou vizí jeho efektivního využití (Seropian, Brown, Gavilanes & Deriggers, 2004), pak byl ovšem pedagogický efekt takového nákupu podstatně podvázán. Velmi brzo se ukázalo, že bez specializovaného týmu pedagogů, kteří mají jasnou představu o využití simulátorů v pedagogické praxi, jsou investice do drahého zařízení neefektivní.

Vzhledem k technologickým i personálním nárokům proto vznikla na řadě univerzit i mimo ně specializovaná simulační centra pro lékařskou výuku na simulátorech, např. na Harwardu existuje „Center for Medical Simulation“ - http://www.harvardmedsim.org/, v Oxfordu se problematikou lékařské výuky na simulátorech zabývá „Oxford Simulation Centre“ - http://www.oxsim.ox.ac.uk/, a v Izraeli vzniklo štědře dotované „Israel Center for Medical Simulation“ - http://www.msr.org.il/.

2.8 Simulátory na internetové pavučině

Internet je dnes stále častěji využíván jako distribuční médium pro výukové simulátory (šíření progra-

mů na discích CD ROM či DVD ROM je na ústupu). Internet dává možnost na jednom distribučním místě soustřeďovat nabídku, řídit přístupová práva a snadno distribuovat jeho aktualizace.

Internet není jen distribučním prostředím pro stažení simulátoru (s nutností jeho následné instalace na počítači), stává se někdy i prostředím pro vlastní běh výukových programů sledovaných přímo v internetovém prohlížeči. Často však jde jen o interaktivní animace a nikoli simulace s matematickým modelem na pozadí (viz např. http://www.interactivephysiology.com). Objevují se však i výukové simulátory běžící přímo v internetovém prohlížeči – jako např. výše zmíněný simulátor tlakových a ob-

Obr. 14 – Ukázky obrazovek z virtuální fakultní nemocnice University of Aukland a péče o virtuálního pacienta v 3D prostředí Second Life (http://slurl.com/secondlife/Long%20White%20Cloud/98/66/27).

jemových poměrů v srdci Columbia Univerzity (Burkhoff & Dickstein, 2003) – viz http://www.columbia.edu/itc/hs/medical/heartsim, nebo simulátor AIDA, modelující virtuálního diabetického pacienta (Reed & Lehmann, 2005; Lehmann, Tarin, Bondia, Teufel, & Deutsch, 2007) – viz - http://www.2aida.net/).

Nové možnosti, zatím ještě příliš často využívané, přináší využití virtuálního internetového 3-D světa pro lékařskou výuku. Virtuální 3D světy představují kolaborativní prostředí, zobrazitelné pomocí internetového prohlížeče. V tomto světě je každý účastník reprezentován figurkou (tzv. avatarem), kterého ovládá. Prostřednictvím svého avatara se může toulat po virtuálním světě (procházet se či dokonce teleportovat do jiných oblastí virtuálního světa), v reálném čase může komunikovat s okolními „avatary“ a provádět nejrůznější aktivity mimo jiné třeba i ve virtuálním 3D prostředí lékařského zařízení může pečovat o virtuálního pacienta (obr. 14). Virtuálním pacientem může být avatar ovládaný učitelem, ale také i naprogramovaný avatar propojený se simulačním modelem (Danforth, Procter, Heller, Chen, & Jonson, 200). Jedním z nejrozšířenějších 3D virtuálních prostředí je 3D prostředí Second Life (http://secondlife.com/).Právě v tomto prostředí se v poslední době také nezřídka využívá i pro lékařskou výuku. (Boulos, Hetherington, & Wheeler, 2007; Toro-Troconis, Partridge, & Barret, 2008; Diener, Windsor, & Bodily, 2009; Toro-Troconis, & Boulos, 2009, Procetr, Heller, Chen, & Jonson, 2009).

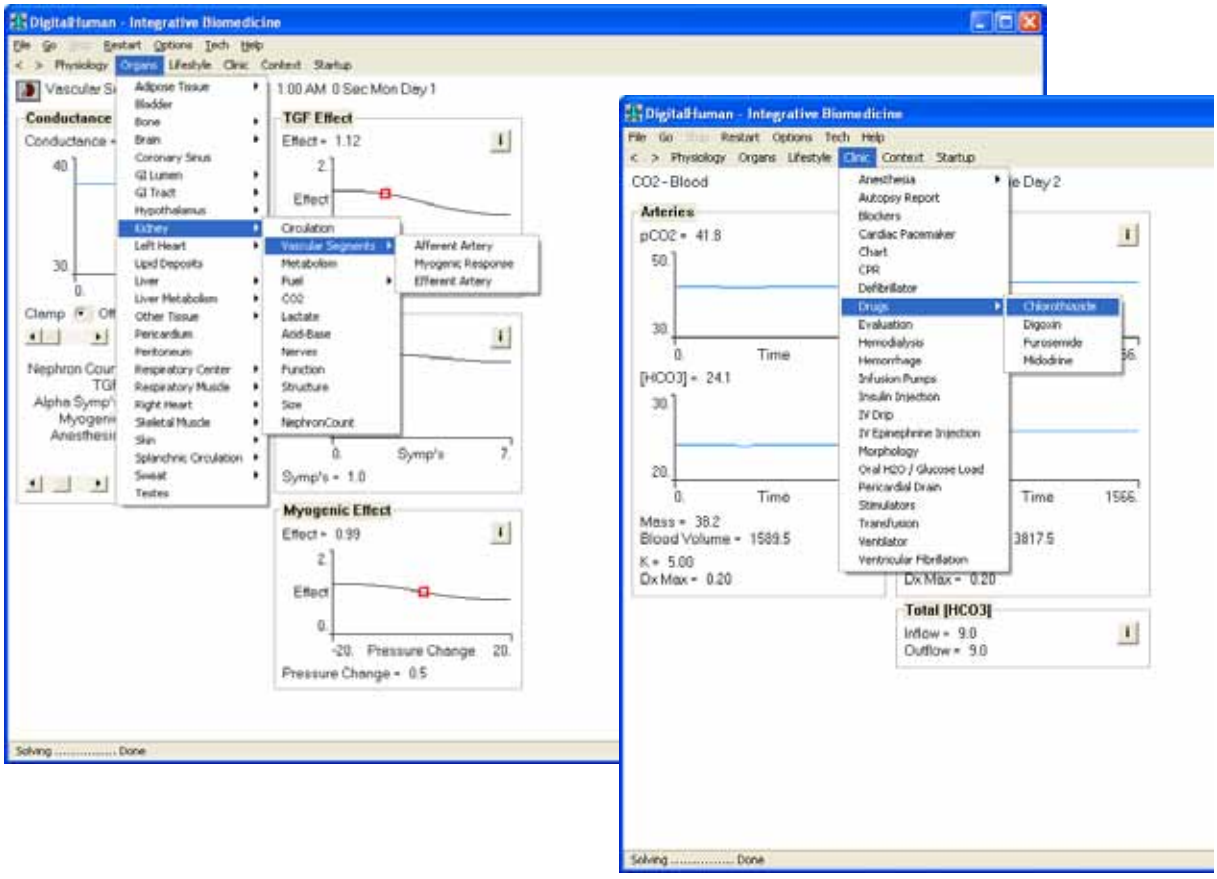
2.9 Modely pro simulátory jako open source

Modely, které jsou v pozadí výukových simulátorů, představují formalizované vyjádření popisu modelované fyziologické reality. Tento cíl mají i mezinárodní projekty PHYSIOME a EUROPHYSIOME, které se snaží o co možno nejúplnějším zveřejňování fyziologických modelů jako vědeckého výsledku; proto také struktura modelů využívaných ve výukových simulátorech je nezřídka šířena jako „open source“. V rámci tohoto projektu vzniklo i několik databází obsahující přesnou strukturu fyziologických modelů.

Databáze modelů vyjádřených v prostředí standardizovaného jazyka CellML (Lloyd, Halstead & Nielsen, 2004) je dostupná na adrese http://models.cellml.org/. Jiná databáze fyziologických modelů vytvořená v otevřeném prostředí jazyka JSIM (Raymond, Butterworth & Bassingthwaighte, 2003) je dostupná na adrese http://www.physiome.org/model/doku.php. Databáze anotovaných biologických modelů popsaných ve značkovacím jazyce SBLM (System Biology Markup Language), používaného jako jeden ze standardů pro popisování modelů biologických systémů (http://sbml.org/) je dostupná na adrese http://www.ebi.ac.uk/biomodels-main/.

Modely, které jsou podkladem rozsáhlých výukových simulátorů, však v těchto databázích obvykle nenajdeme, protože jejich podrobnou strukturu autoři, až na některé výjimky, většinou nezveřejňují.

Naším skromným příspěvkem do volně šířených databází modelů fyziologických systémů je volně přístupná knihovna PHYSIOLIBRARY vytvořená pro prostředí jazyka Simulink (http://physiome.cz/simchips). Tato knihovna mimo jiné obsahuje též zdrojový kód modelu, který byl podkladem pro náš výukový simulátor Golem (Kofránek, Andrlík, Kripner, & Mašek, 2002b).



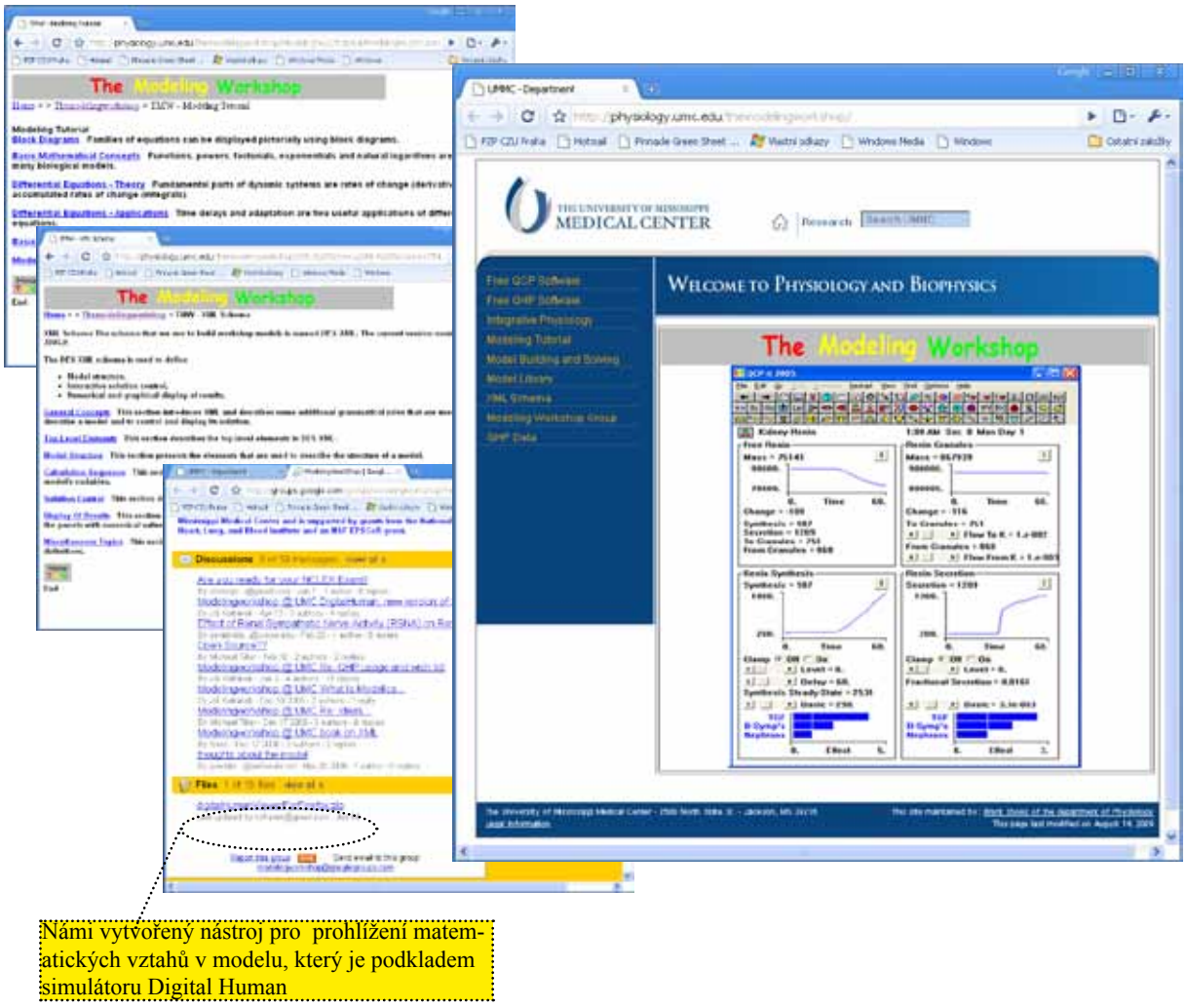
Obr. 15 – Prostředí výukového simulátoru Quantitative Human Physiology (QHP) (nyní též šířeného pod názvem Digital Human). Simulátor sice zatím nemá tak atraktivní ikonky v panelové liště jako jeho předchůdce, simulátor QCP (viz obr. 11), nabízí ale rozvětvenější menu, větší počet proměnných, jejichž hodnoty lze sledovat, ale především neskrývá strukturu matematického modelu fyziologických regulací, který je jádrem výukového simulátoru. Simulátor je šířen ve zdrojové formě obsahující popis všech matematických vztahů prostřednictvím speciálního jazyka odvozeného z formátu XML. Zároveň se zdrojovým textem je šířen i jeho překladač, který po překladu zdrojového textu také spustí vlastní simulátor.

Jeden z nejsložitějších modelů fyziologických systémů je jádrem výše zmíněného výukového simulátoru ***Quantitative Circulatory Physiology (QCP)***. Jeho dalším rozšířením je výukový simulátor ***Quantitative Human Physiology (QHP)***, (Coleman, Hester & Summers, 2009) obsahující více než 4000 proměnných, který v současné době zřejmě představuje nejrozsáhlejší model fyziologických regulací. Simulátor je veřejně přístupný na webu University od Mississippi http://physiology.umc.edu/themodelingworkshop/ (poslední verzi autoři nazvali již ***Digital Human***).

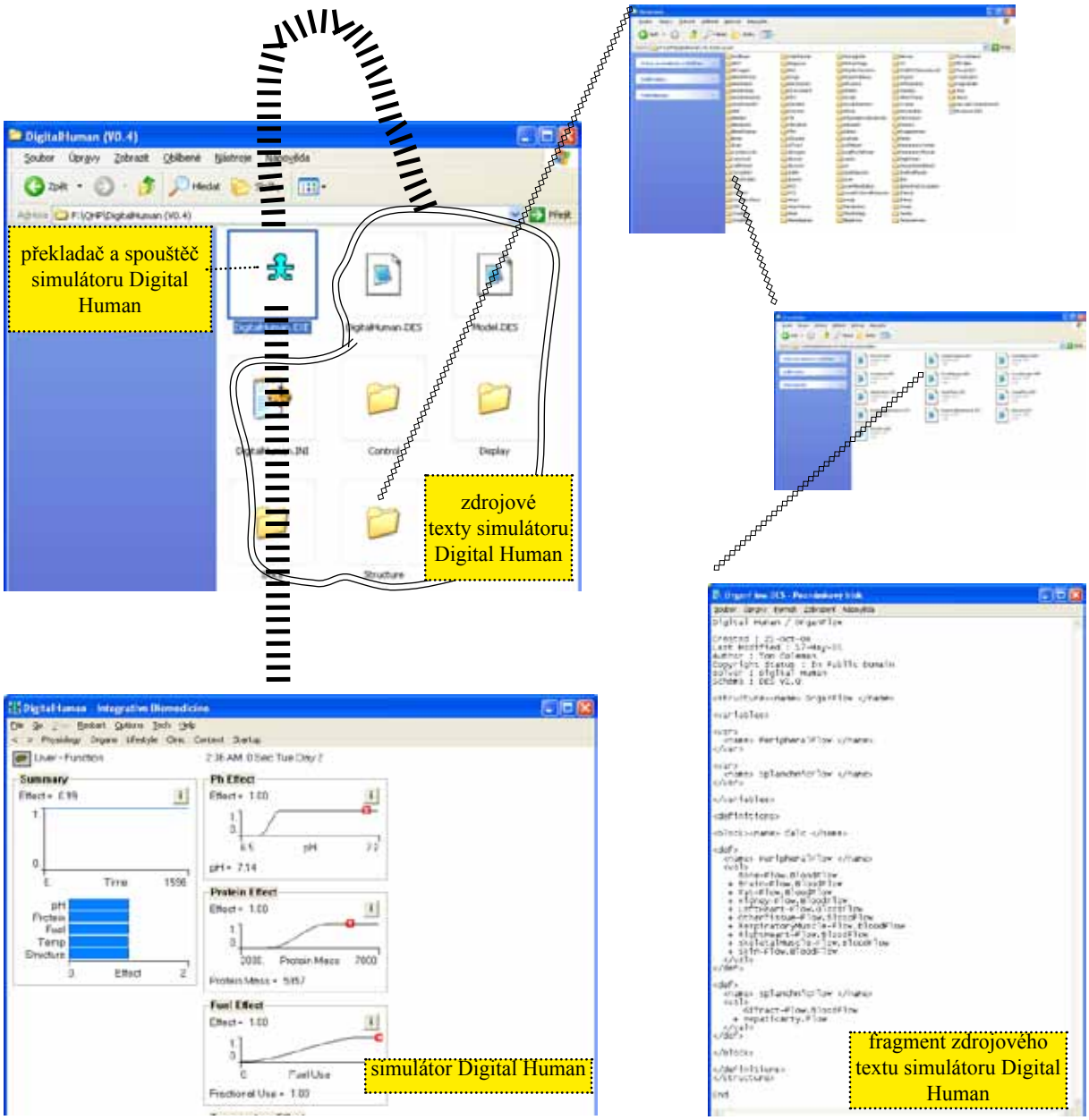
Model má velmi rozvětvené menu a umožňuje simulovat řadu patologických stavů, včetně vlivu příslušné terapie (obr. 15).

Na rozdíl od předchozího simulátoru QCP, jehož matematické pozadí je uživateli skryto ve zdrojovém kódu simulátoru napsaném v C++, jde simulátor QHP jinou cestou. Jeho autoři se rozhodli oddělit implementaci simulátoru a popis rovnic modelu tak, aby struktura modelu mohla být zřejmá pro širší vědeckou komunitu.

Hlavní architekt tohoto modelu Thomas Coleman proto již v roce 1985 vypracoval speciální jazyk pro zápis struktury modelu i definic prvků uživatelského rozhraní simulátoru. Jazyk je založen na upravené XML notaci. Model je pak zapsán pomocí XML souborů. Speciální překladač (DESolver) pak přeloží tyto XML soubory do spustitelného kódu simulátoru.



Obr. 16 – Jazyk, požívaný na popis simulátoru QHP (resp. Digital Human) je podrobně popsán na webových stránkách University of Mississippi Medical Center. K dispozici jsou nabízeny nejrůznější menší modely popsané v tomto jazyce i vlastní zdrojový text a příslušný překladač simulátoru QHP. Zároveň je zde organizována diskusní skupina pro výměnu zkušeností i informací. Z ní je možné stáhnout i námi vytvořený nástroj pro prohlížení matematických vztahů v modelu, které jsou ve své zdrojové podobě roztroušeny ve více než dvou tisícovkách souborů.



Obr. 17 – Veškeré potřebné soubory výukového simulátoru QHP/Digital Human. Simulátor je určen pro systém Windows, nevyžaduje zvláštní instalaci, stačí pouze rozbalit „zazipované“ soubory do nějakého adresáře. Po kliknutí na ikonku překladače DigitalHuman.exe překladač přeloží zdrojový text obsažený ve stovkách adresářů a více než dvou tisících souborů a spustí vlastní simulátor. I když je zdrojový text simulátoru i celý matematický model na jeho pozadí tímto způsobem nabízen jako „open source“ (a uživatel si teoreticky může i model modifikovat), je orientace v matematických vztazích prohlížením tisícovek vzájemně provázaných XML souborů poměrně obtížná.

Podrobný popis tohoto jazyka, stejně jako jeho překladač (DESolver) a příslušný výukový tutoriál jsou volně přístupné na výše uvedeném webu University od Mississippi (obr. 16).

Právě pomocí tohoto XML jazyka je zapsán i nový model QHP (resp. Digital Human). Vzhledem k rozsahu tohoto modelu byl ale pro něj vyvinut speciální překladač (který je možné stejně jako všechny zdrojové soubory, z univerzitního webu volně stáhnout).

Uživatel proto může model upravovat i modifikovat. Potíž tkví ale v tom, že zdrojové XML texty celého modelu jsou napsány v celkem 2833 souborech umístěných v 772 složkách (obr. 17).

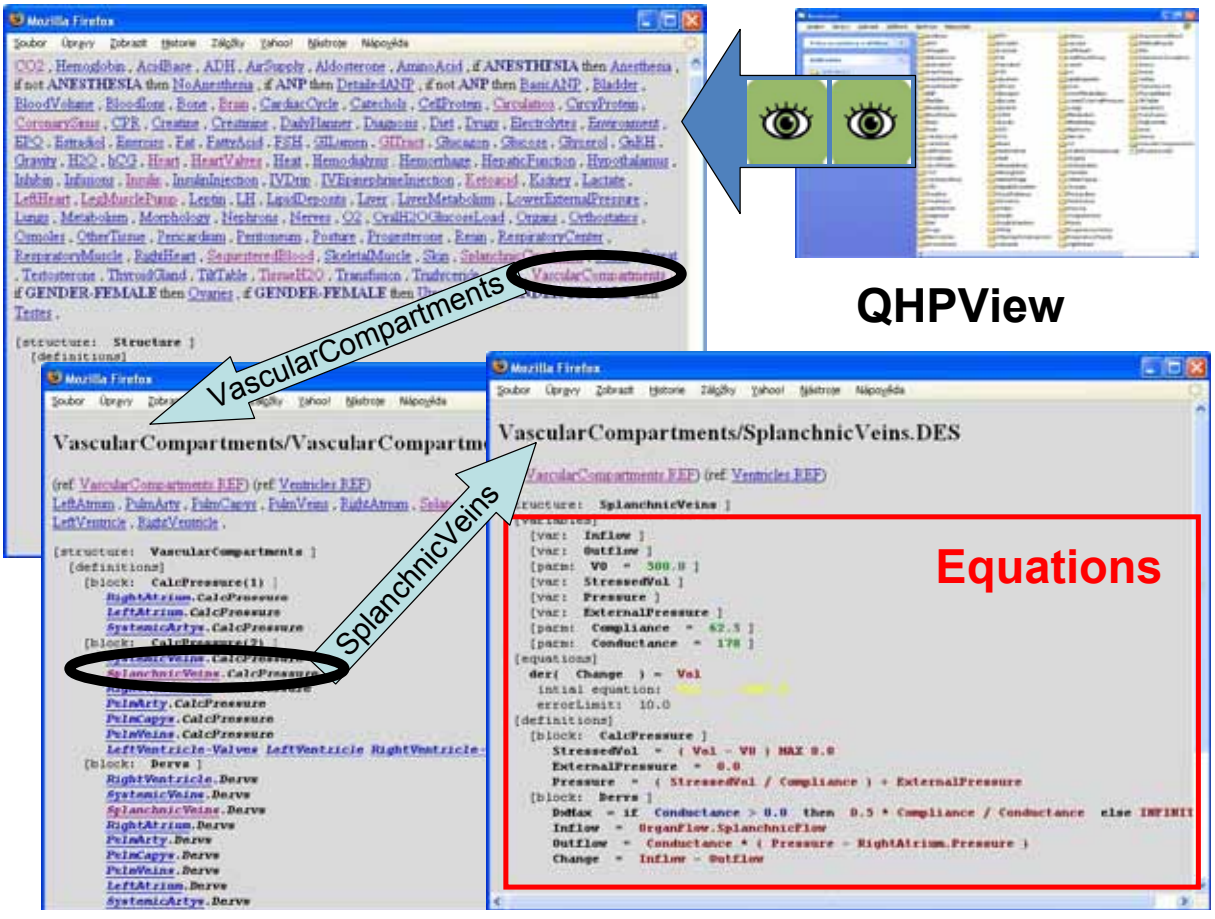
Celková struktura modelu a jednotlivé návaznosti jsou tudíž velmi nepřehledné. Proto například

Obr. 18 – Thomas Coleman, spolupracovník A.C. Guytona, který v sedmdesátých letech programoval všechny jeho modely, tvůrce simulátorů Human a QCP, nyní hlavní systémový architekt simulátoru Quantitative Humman Physiology/Digital Human u hrobu rabína Jehudy Levy ben Becalela, známého jako rabi Löw, údajného tvůrce mýtického golema – umělé bytosti z hlíny na návštěvě v Praze v roce 2008.

mezinárodní výzkumný tým v projektu SAPHIR (System Approach for Physiological Integration of Renal, cardiac and repiratory control) jako východisko pro tvorbu nového rozsáhlého modelu fyziologickch funkcí raději zvolil staré modely Guytona z roku roku 1972 (Guyton, Coleman & Grander, 1972) a model Ikedy z roku 1979 (Ikeda, Marumo & Shirsataka, 1979) a nesáhl po volně přístupném modelu QHP. Zdrojové texty modelu QHP se účastníkům nového projektu zdály velmi špatně čitelné a obtížně srozumitelné (Thomas, a další, 2008).

Nový model QHP (resp. model Digital Human) je stále ještě ve fázi testování, rozšiřování a modifikace. S hlavním architektem tohoto simulátoru Thomasem Colemanem (obr. 18) a dalšími americkými autory z University od Mississippi jsme dojednali ***dlouhodobou přímou spolupráci na dalším rozvoji tohoto modelu***.

V rámci naší spolupráce jsme nejprve vytvořili ***speciální softwarový nástroj QHPView*** (obr. 19), který z tisícovek souborů zdrojových



Obr. 19 – Námi vytvořený vizualizační nástroj QHPView umožní zpřehlednit strukturu modelu QHP/Digital Human, původně zapsaného ve více než dvou tisícovkách XML souborů rozházených do stovek adresářů, v nichž rovnice a jednotlivé návaznosti nebyly na první pohled zřetelné.

textů vytvoří přehledné zobrazení všech matematických vztahů a souvislostí.

Nástroj jsme nabídli na webových stránkách QHP (http://physiology.umc.edu/themodelingworkshop/) jako „open source“. Naše další účast na tvorbě a rozšiřování modelu QHP spočívá v jeho implementaci (a modifikaci) v prostředí nového modelovacího jazyka Modelica (Fritzon, 2003), v němž bude struktura modelu mnohem přehlednější. Zdrojový text modelu v Modelice zveřejňujeme jako open source.

2.10 Současné trendy tvorby a využití lékařských výukových simulátorů

Výukové programy využívající simulační hry a specializované lékařské trenažéry se pomalu stávají běžnou součástí moderní výuky lékařů. Jejich tvorba ale není jednoduchá. Vznikají na univerzitách i ve firmách. Současné trendy lze stručně shrnout:

- pro distribuci tak i pro vlastní provozování výukových simulátorů se stále častěji využívá internet;
- výukové simulátory jsou často doplňovány výkladovou částí realizovanou jako internetem dostupný multimediální výukový program;
- výuka s využitím simulátorů klade vyšší nároky na učitele, než klasická výuka; na druhé straně simulátory dávají vyučujícímu do rukou didakticky velice efektivní prostředek;
- na univerzitách postupně vznikají specializovaná pracoviště realizující výuku na simulátorech;
- pro simulátory se začíná též využívat virtuální 3D prostředí (např. Second Life);
- pro simulační hry se využívá nejen obrazovky počítače, jako trenažéry pro lékařskou výuku se používá i robotizovaná figurína pacienta;
- tvorba výukových programů a simulátorů se také stává druhem podnikání;
- struktura simulačních modelů (příslušné rovnice a hodnoty parametrů) využívaných ve výukových simulátorech se často stává pečlivě střeženým firemním know-how;
- mezinárodní projekty Physiome a Europhysiome se snaží o co možno nejúplnějším zveřejňování fyziologických modelů jako vědeckého výsledku; proto také struktura modelů využívaných ve výukových simulátorech je nezřídka šířena jako „open source“
- mezi nejsložitější fyziologické modely patří model Quantitative Human Physiology (nyní také šířený pod názvem „Digital Human“), jehož podrobná struktura je šířena jako „open source“ (viz http://physiology.umc.edu/themodelingworkshop).

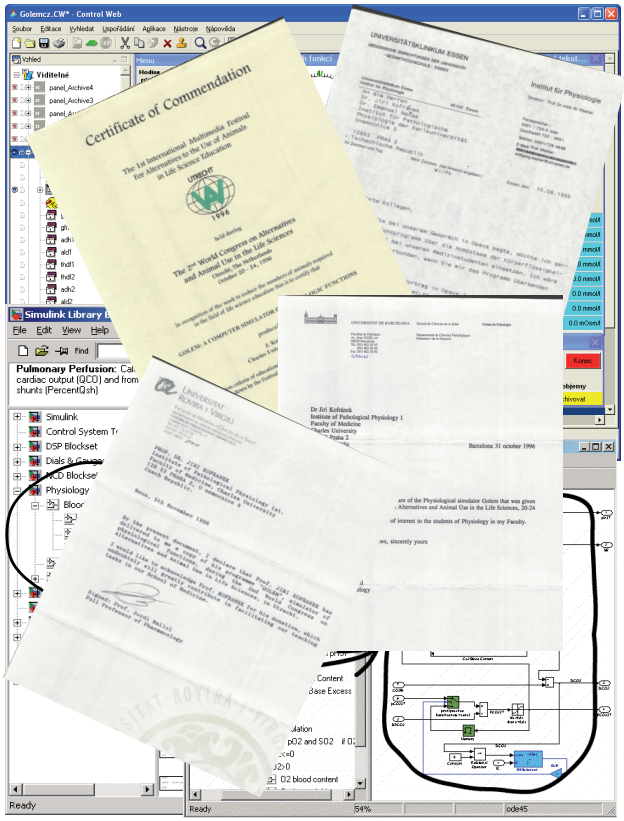
3 Zkušenosti s tvorbou a využitím lékařského výukového simulátoru a cíle další práce

V polovině devadesátých let jsme se pustili do vývoje vlastního lékařského simulátoru. Jeho podkladem byl komplexní model fyziologických regulací, zaměřený především na modelování poruch homeostázy vnitřního prostředí.

Modelováním poruch vnitřního prostředí se zabýváme již poměrně dlouhou dobu.

Na počátku osmdesátých let jsme se věnovali tvorbě modelu acidobazické rovnováhy a přenosu krevních plynů (Kofránek, 1980). Později jsme jej rozšířili na komplexní model homeostázy vnitřního prostředí (Kofránek, Pokorný, Wünsch, Brelidze, Gondžilašvili & Verigo, 1982a-c; Kofránek, Brelidze & Gonžilašvili, 1984; Gondžilašvili, Kofránek, Pokorný & Brelidze, 1987). Vytvořený model byl mimo jiné součástí širšího projektu využití matematických modelů lidského organismu v rámci sovětského kosmického výzkumu (Verigo, 1987), obdobného americkému projektu NASA „Digital Astronauts“. Některé subsystémy modelu našly své praktické využití i pro výpočty některých klinicko-fyziologických funkcí prostřednictvím identifikace modelu na měřená data (Kofránek, a další, 1988).

Námi vytvořený komplexní model vnitřního prostředí jsme se v polovině osmdesátých let pokoušeli využít i ve výuce. Jeho komplexnost však vyžadovala implementaci (v jazyce Fortran) na vzdáleném hardwaru, komunikace s modelem se vedla přes alfanumerický terminál, interaktivita byla proto poněkud těžkopádná a navíc výstupy byly pouze číselné a pseudografy vytvářené ze znakových sad nahrazovaly grafický výstup. Pedagogický efekt proto nakonec neodpovídal úsilí, které jsme věnovali jeho implementaci.



Obr. 20 – Naším příspěvkem k rozšíření simulátorů ve výuce medicíny byl námi vytvářený simulátor GOLEM. Jeho teoretickým podkladem byl matematický model fyziologických vztahů, vytvářený v prostředí Matlab/Simulink. Na základě tohoto modelu byl vlastní výukový simulátor naprogramován v prostředí Control Web. Simulátor byl zdarma šířen po lékařských fakultách jako „public domain“, zkušenosti z jeho nasazením do výuky pak vedly k dalším verzím modelu a příslušného simulátoru. Mezinárodní ohlasy kolegů z různých zemí, využívající tento simulátor ve výuce byly, zvláště zpočátku, velkou motivací pro naši další práci.

3.1 Golem jako výuková hračka pro mediky a lékaře

Růst výkonu osobních počítačů a nové možnosti jejich grafického uživatelského rozhraní nás vedly k tomu, že jsme se k vytvoření výukového simulátoru fyziologických funkcí v druhé polovině devadesátých let vrátili. Vytvořili jsme ***výukový simulátor „Golem“***, (Kofránek, Velan & Kerekeš, 1997; Kofránek & Velan, 1999), který se využíval ve výukové praxi na některých našich i zahraničních lékařských fakultách.

Od samotného počátku byl simulátor vytvářen jako volně šiřitelná výuková pomůcka, Jeho teoretickým podkladem byl rozsáhlý matematický model fyziologických vztahů homeostázy vnitřního prostředí vytvářený ve vývojovém prostředí Matlab/Simulink. Strukturu modelu v Simulinku jsme šířili jako „open source“ (poslední verze tohoto modelu je součástí naší knihovny PHYSIOLIBRARY – http://www.physiome.cz/simchips). Praktickým využitím teoretických výsledků, tj. formalizovaného popisu fyziologických subsystémů organismu pomocí matematického modelu, byla výuková pomůcka – simulátor Golem. Jeho tvorba ovšem vyžadovala další nemalé

programátorské úsilí, spočívající ve vytvoření uživatelského rozhraní a jeho propojení s verifikovaným modelem implementovaným na pozadí simulátoru. Vytvořený výukový simulátor byl volně distribuován na CD ROM discích všem zájemcům z různých lékařských fakult a základě praktických zkušeností ve výuce postupně vznikaly další verze modelu a simulátoru (viz obr. 20).

Simulátor „Golem" byl zaměřen především pro výuku patofyziologie poruch vnitřního prostředí. Umožňuje simulovat zejména smíšené poruchy iontové, osmotické a acidobazické rovnováhy, poruchy transportu krevních plynů, respirační selhání i poruchy funkce ledvin. Umožňoval také sledovat vliv nejrůznější infúzní terapie (Kofránek J., Snášelová, Anh Vu & Svačina, 2001; Kofránek, Snášelová, Anh Vu & Svačina, 2001; Kofránek, Vrána, Velan & Janicadis, 2001).

Simulátor byl koncipován tak, aby ***jeho ovládání bylo pokud možno co nejjednodušší***. Proto se v něm hojně užívaly menu, tlačítka a přepínače (ovladatelné počítačovou myší), knoflíky (kterými lze pomocí myši snadno otáčet a nastavovat tak příslušné hodnoty), myší ovladatelná posuvná šoupátka, okénka pro numerické zadávání či zobrazování jednotlivých hodnot atd.

Vstupy simulátoru byly:

1. Vnější podmínky (jako je barometrický tlak, koncentrace plynů v inspirovaném vzduchu apod.) a počáteční hodnoty některých veličin před začátkem simulace (velikost zásob jednotlivých látek).
2. Rychlosti vnějších příjmů či ztrát jednotlivých látek (iontů, vody, glukózy) do/z organismu (přes gastrointestinální trakt nebo pomocí infúzí). Zadáváním rychlostí příjmů/ztrát bylo možné sestavovat různé scénáře patofyziologických poruch (např. průjem, zvracení...) a jejich infúzní léčbu (pomocí infúzních roztoků nejrůznějšího druhu).
3. Některé „vnitřní" parametry fyziologických regulátorů v simulačním modelu, jejichž změnou bylo možné simulovat různé patofyziologické regulačníporuchy (např. diabetes).

Výstupy simulátoru se objevovaly:

1. V okénkách s animovanými schématy fyziologických subsystémů: pro rychlou orientaci byla barva okénka s číselným výstupem zelená v případě normálních hodnot, červená v případě jejího zvýšení nad fyziologický limit, či světle modrá při snížení pod hranici rozmezí normálních hodnot; změna hodnot jednotlivých veličin byla často provázena změnou animovaného schématu (např. změnou „hladiny", změnou „odkapávání z kohoutku", změnou směru animovaných šipek apod.).
2. V okénkách s grafickými výstupy: mezi okny s fyziologickým schématem a oknem s grafy bylo možné snadno přepínat pomocí příslušných tlačítek.

Simulátor se osvědčil zejména při výuce patofyziologie a klinické fyziologie, kde přispíval k lepšímu pochopení dynamických souvislostí rozvoje nejrůznějších patologických stavů a vlivu příslušné terapie.

Simulátor byl vhodným doplňkem klasických učebnic patofyziologie a klinické fyziologie. V učebnicích se, z didaktických důvodů, obvykle probírají jednotlivé fyziologické poruchy odděleně po jednotlivých fyziologických subsystémech a po jednotlivých poruchách. U reálných pacientů se však nezřídka setkáváme s kombinací jednotlivých poruch. Simulátor umožnil ve virtuální realitě sledovat vývoj jednoduchých i kombinovaných poruch a příslušné regulační odpovědi fyziologických subsystémů na jednoduché i kombinované poruchy.

Tak např. při klasickém výkladu poruch vnitřního prostředí se postupně probírají poruchy jednotlivých subsystémů vnitřního prostředí (tj. poruchy acidobazické, iontové, objemové osmotické homeostázy) samostatně.

V klinické praxi jsou ale u pacientů poruchy vnitřního prostředí často kombinované (např. se vyskytuje smíšená acidobazická, iontová a objemová porucha) a lékař musí pro správnou diagnostiku a volbu adekvátní terapie dobře porozumět návaznostem regulačních smyček jednotlivých subsystémů. Z pedagogického hlediska je proto výhodné, že simulátor díky své komplexnosti umožnil názorně demonstrovat, ***jak jednotlivé fyziologické sybsystémy spolu navzájem souvisejí*** a ***jak se tyto souvislosti projevují v jednotlivých patofyziologických stavech***.

3.2 Golem místo pacienta

Simulátor umožňoval nastavováním jednotlivých vstupních veličin sledovat rozvoj nejrůznějších patologických stavů a vlivu příslušné terapie na stav virtuálního pacienta. Simulátor umožňuje změnou hodnot svých vstupů simulovat příslušné ***patologické stavy*** a některé ***terapeutické zásahy.***

Pro ilustraci si uveďme příklad postupného rozvoje a léčení metabolické acidózy.

Pootočením knoflíku na jednom z panelů simulátoru (na panelu acidobazické metabolické bilance – viz obr. 21) zvýšíme metabolickou produkci silných kyselin a tak vyvoláme metabolickou acidózu, způsobenou zvýšením tvorby silných kyselin (obdobně jako při laktátové acidóze, diabetické ketoacidóze aj.).

Na dalším panelu zobrazujícím celkovou acidobazickou bilanci (viz obr. 22) okamžitě vidíme důsledek: Metabolická produkce silných kyselin, disociujících na příslušné anionty a vodíkové ionty (H^+), převýší jejich exkreci ledvinami. V ledvinách se exkretované vodíkové ionty se váží na fosfáty a amoniak a vylučují se jako tzv. titrovatelná acidita a amonné ionty (NH_4^+). V daném případě je bilance mezi metabolickým přísunem vodíkových iontů do extracelulární tekutiny a jejich vylučováním kladná (vodíkové ionty se hromadí v extracelulární tekutině rychlostí 942 mmol/24hod.).

Organismus na to reaguje tím, že vodíkové ionty jsou v extracelulární tekutině pufrovány bikarbonátovými i nebikarbonátovými pufry. a zároveň se zvýší výměna vodíkových iontů za draselné a sodíkové ionty na membráně buněk. Ionty H^+ vstupují do buněk, kde jsou pufrovány intracelulárními nárazníky. Neblahým důsledkem ale je postupné vymývání zásob draslíku v buňkách. Déletrvající acidémie proto může vést k nebezpečnému snížení zásob draslíku.

Na panelu acidobazické rovnováhy krve (viz obr. 23) můžeme sledovat reakci pufračních systémů krve na akutně vzniklou metabolickou acidózu. Nadbytečné vodíkové ionty se v krvi vážou na bikarbonát a nebikarbonátové nárazníkové báze, tj. fosfáty, plazmatické bílkoviny a hemoglobin (na panelu jsou všechny nebikarbonátové báze označeny souhrnně jako BUF^-), pH plazmy klesá, rozvíjí se acidémie, Base Excess a hladina aktuálních bikarbonátů klesá.

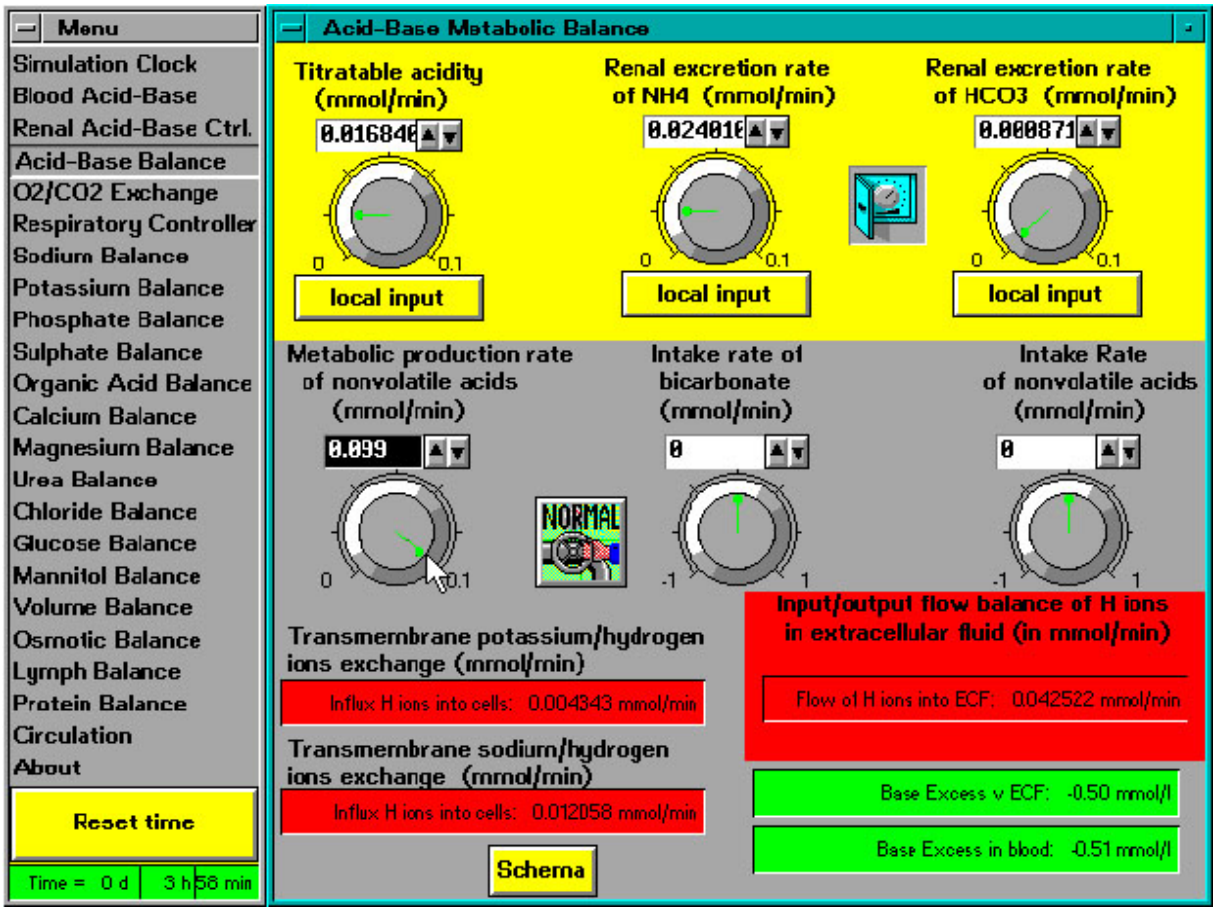
V klinické praxi se pro vyhodnocování poruch acidobazické rovnováhy využívají kompenzační diagramy. Kompenzační diagram je v simulátoru znázorňován na speciálním panelu (viz obr. 24). Na tomto diagramu můžeme sledovat, jak se acidobazická porucha vyvíjí a jak se postupně uplatňují kompenzační a korekční mechanismy. V daném simulovaném případě vidíme, že hodnoty acidobazické rovnováhy jsou na kompenzačním diagramu v pásmu akutní metabolické acidózy, hodnota pH je 7,1.

Pomalu (během dvanácti hodin) se začíná rozvíjet kompenzační odpověď respirace na metabolickou acidózu – PCO_2 pomalu klesá (křivka na kompenzačním diagramu se otáčí směrem dolů).

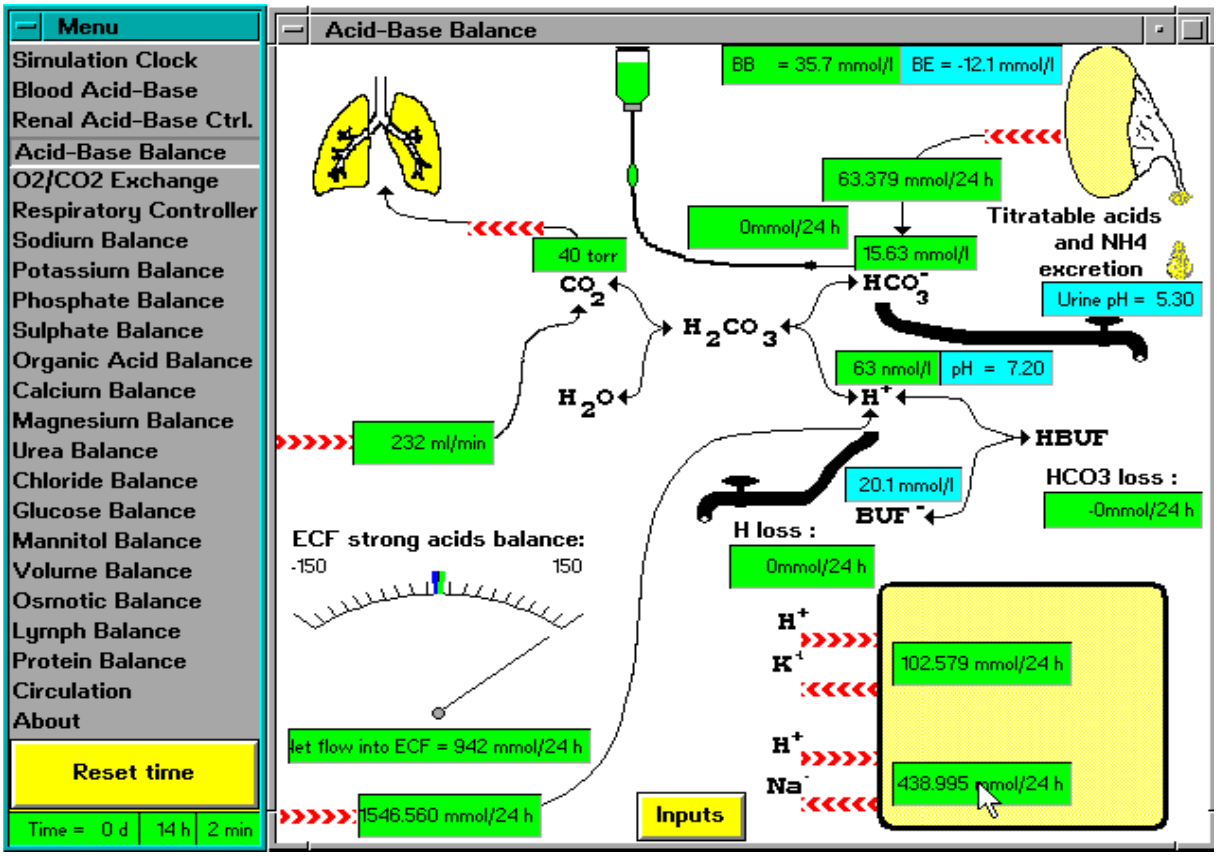
Příčinou postupného zvyšování alveolární ventilace je reakce dechového centra na metabolickou acidózu. Reakci respiračního regulátoru můžeme sledovat na dalším panelu (viz obr. 25). Hladina bikarbonátů v krvi se snížila díky vazbě nadbytečných se vodíkových iontů při pufrační reakci. Protože, na rozdíl od CO_2, bikarbonáty pronikají přes hematoencefalickou bariéru pomalu, vzniká mezi mozkomíšním mokem a krevní plazmou koncentrační gradient bikarbonátů. Na panelu můžeme sledovat, jak se bikarbonáty zvolna přesouvají z moku do krve, a jak jejich hladina v mozkomíšním moku klesá. Ztráta bikarbonátů postupně posouvá rovnováhu v bikarbonátovém systému moku doprava, což vede k postupnému zvyšování koncentrace vodíkových iontů a k poklesu pH moku. Pokles pH mozkomíšního moku dráždí dechové centrum, důsledkem je pozvolna se zvyšující alveolární ventilace. Během 6-12 hodin (podle tíže poruchy) se hladiny bikarbonátů v moku a v krevní plazmě vyrovnají a respirační odpověď na metabolickou acidózu dosáhne svého maxima.

Další vývoj metabolické acidózy na panelu obsahujícím kompenzační diagram (obr. 26). Zvýšení alveolární ventilace vede k poklesu hodnot PCO_2 v arteriální krvi (a vlivem pufrační reakce) ke zvýšení pH z původní nízké hodnoty 7,1 (viz obr. 24) na hodnotu 7,19. Acidobazické hodnoty simulovaného pacienta se nacházejí v pásmu maximální respirační kompenzace.

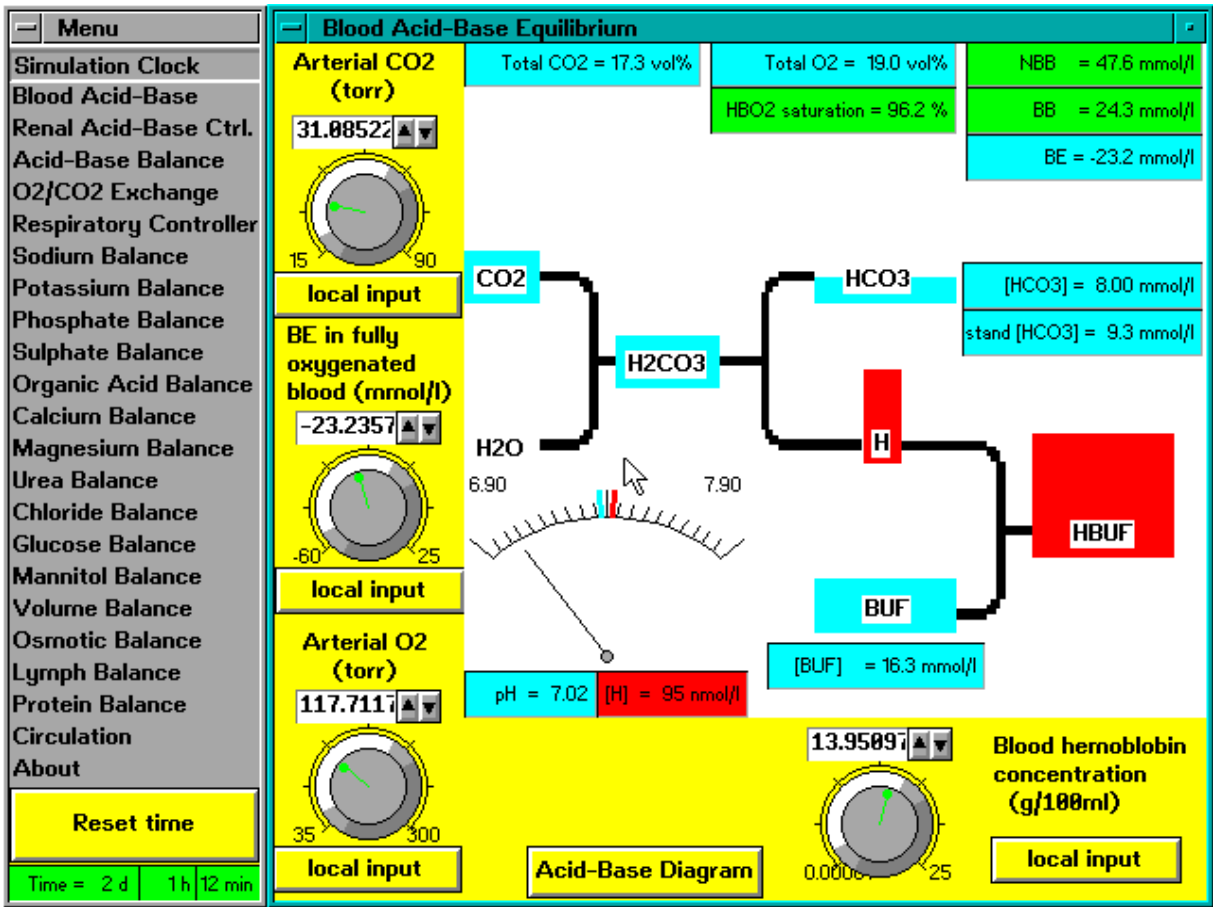
Respirační systém sám osobě ale nedokáže plně korigovat kyselé pH na normální hodnotu 7,4. K plné korekci pH je zapotřebí rozvinutí renální korekce, kdy ledviny zvýší (během 3-5 dnů) vylučování vodíkových iontů vázaných na fosfáty a amoniak, tj. ve formě tzv. titrovatelné acidity a amonných iontů NH_4^+ a postupně vyloučí nahromaděné silné kyseliny.



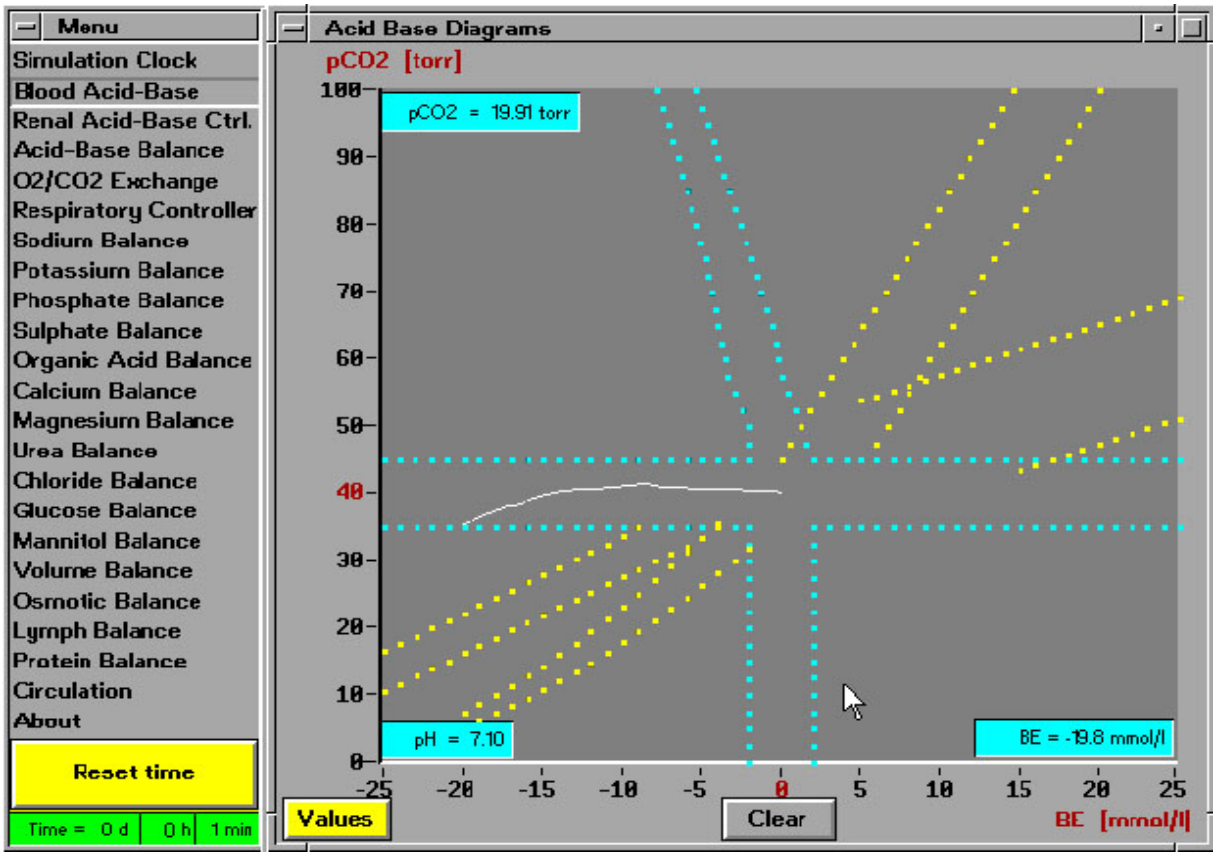
Obr. 21 – Ukázka práce se simulátorem GOLEM při sledování patogeneze metabolické acidózy.



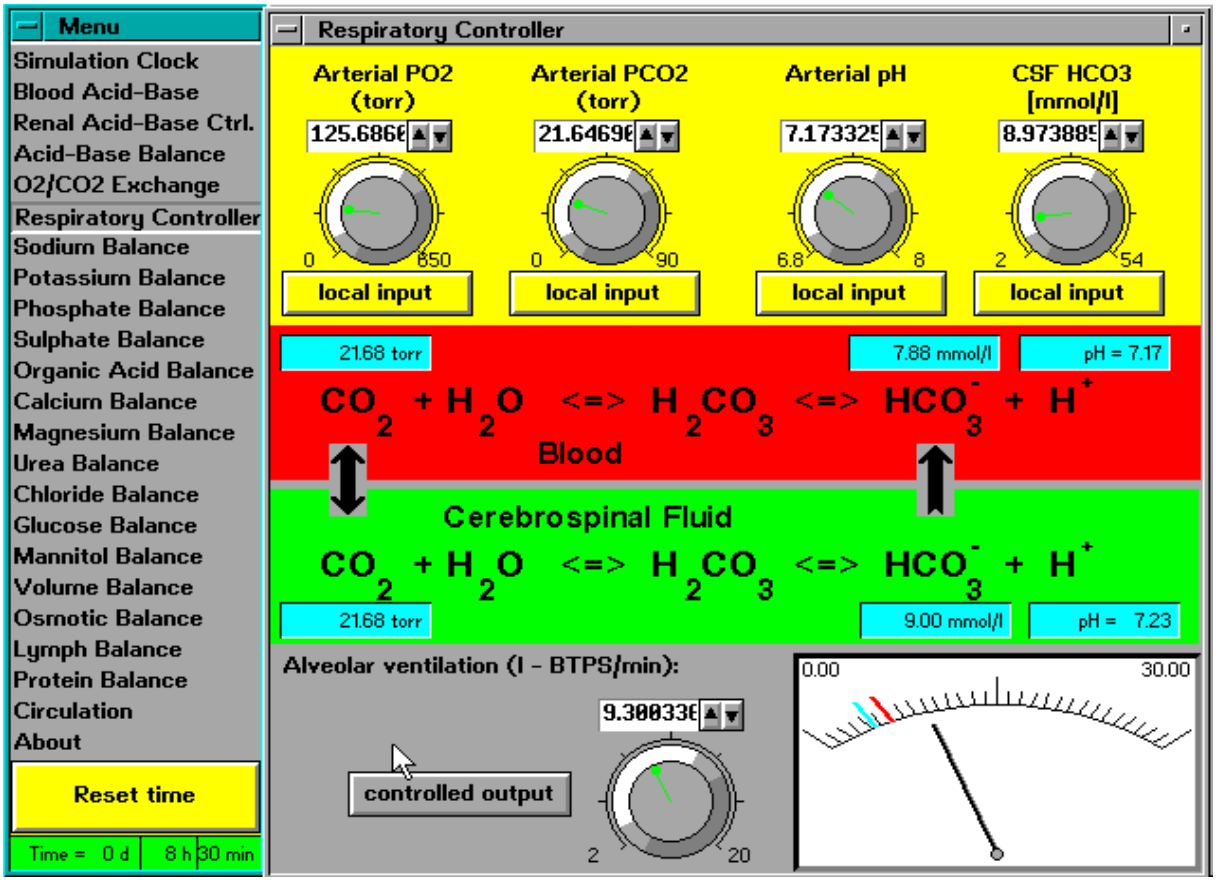
Obr. 22 – Porucha bilance mezi tvorbou silných kyselin a jejich vylučováním v ledvinách při metabolické acidóze.



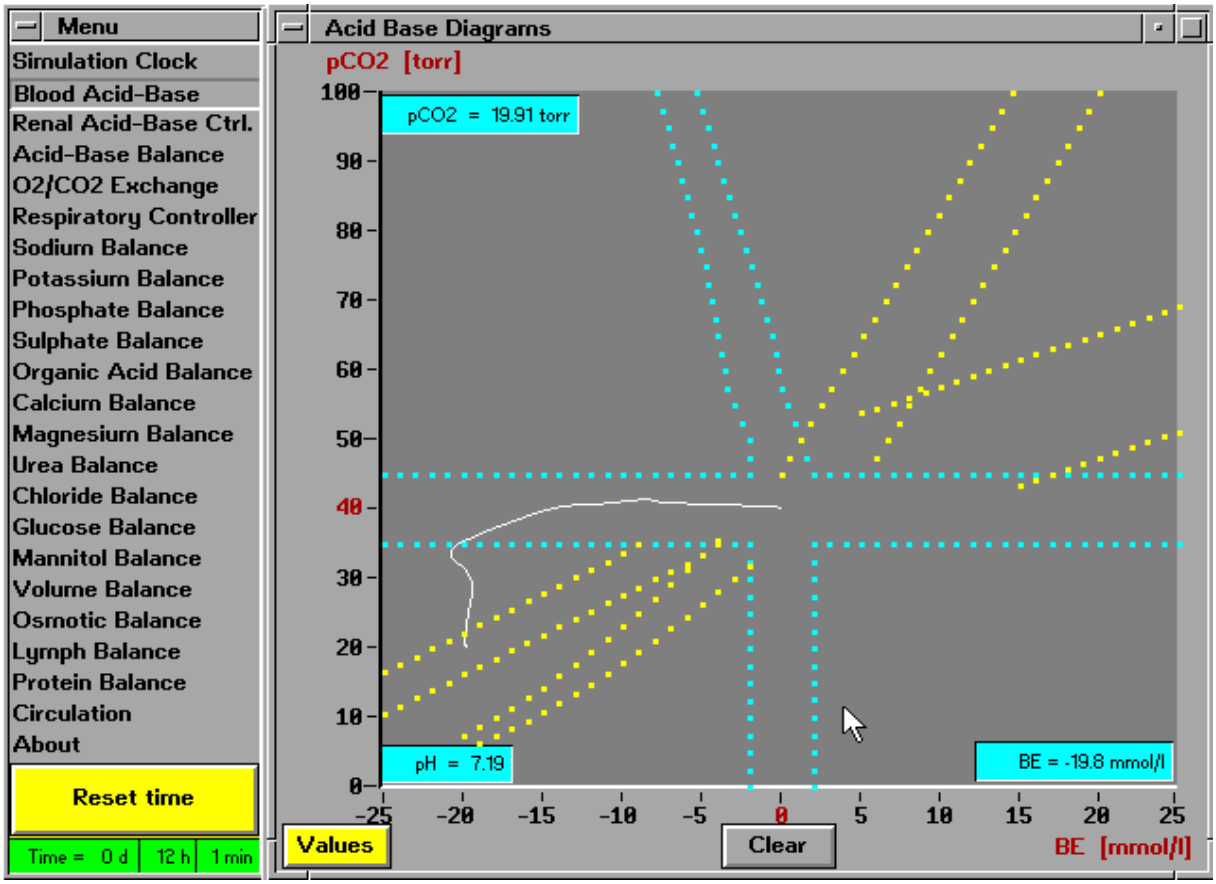
Obr. 24 – Reakce pufračních systémů krve na metabolickou acidózu.



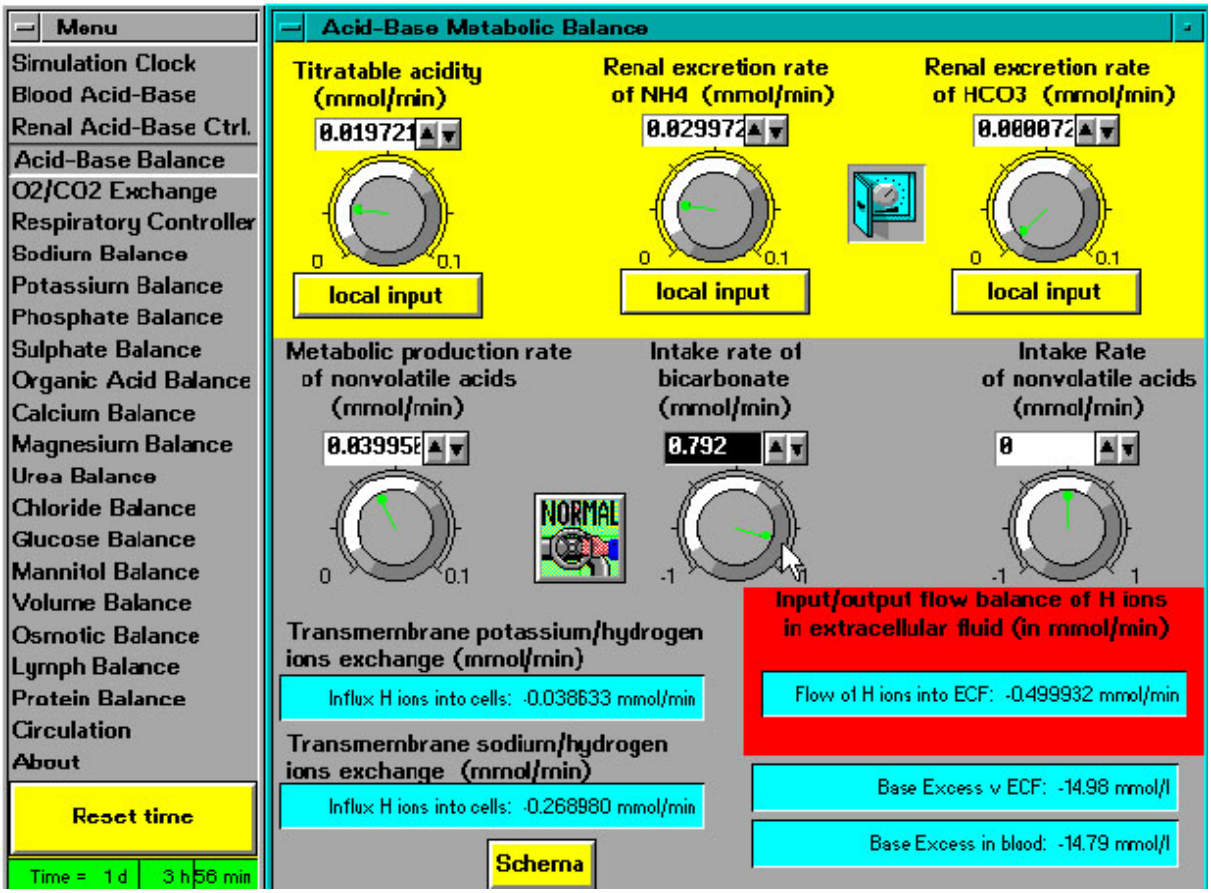
Obr. 23 – Akutní metabolická acidóza na kompenzačním diagramu.



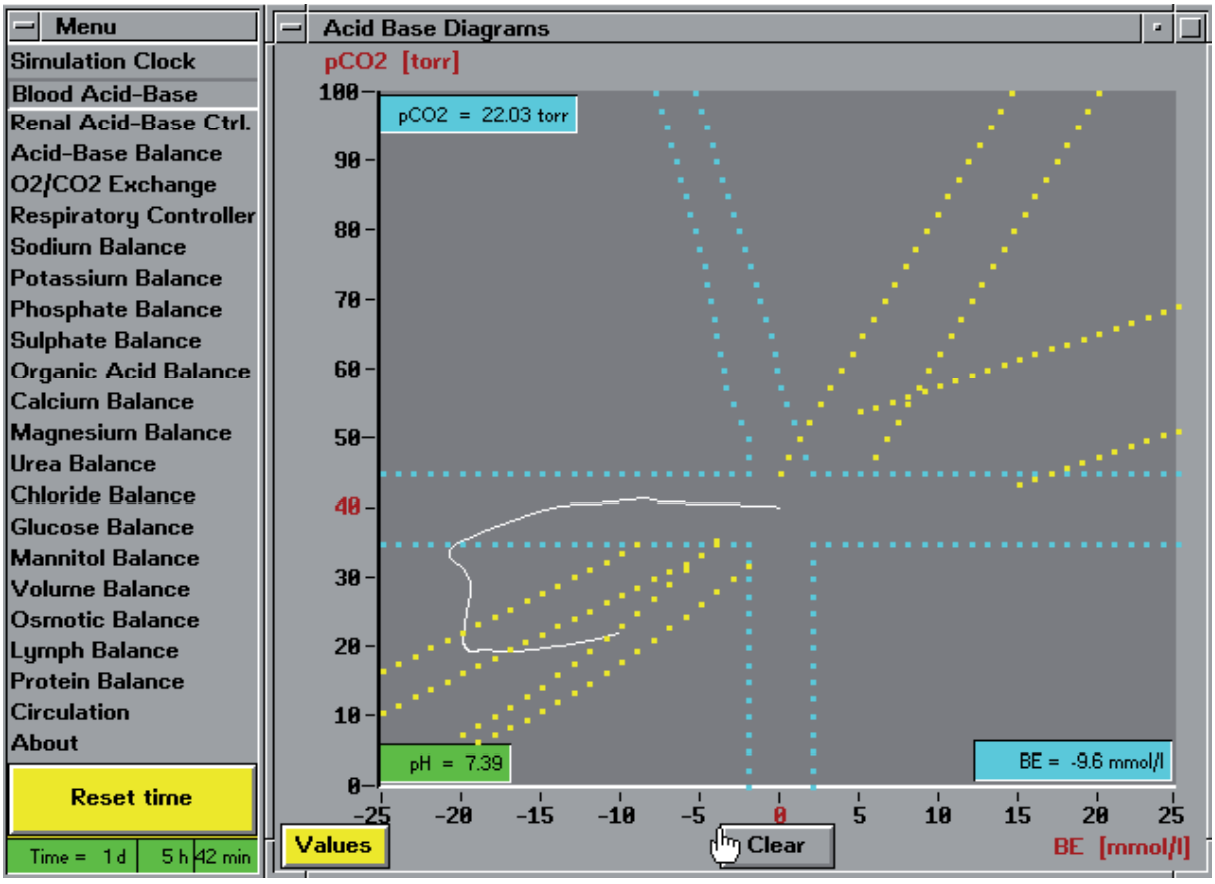
Obr. 26 – Reakce dechového centra na metabolickou acidózu.



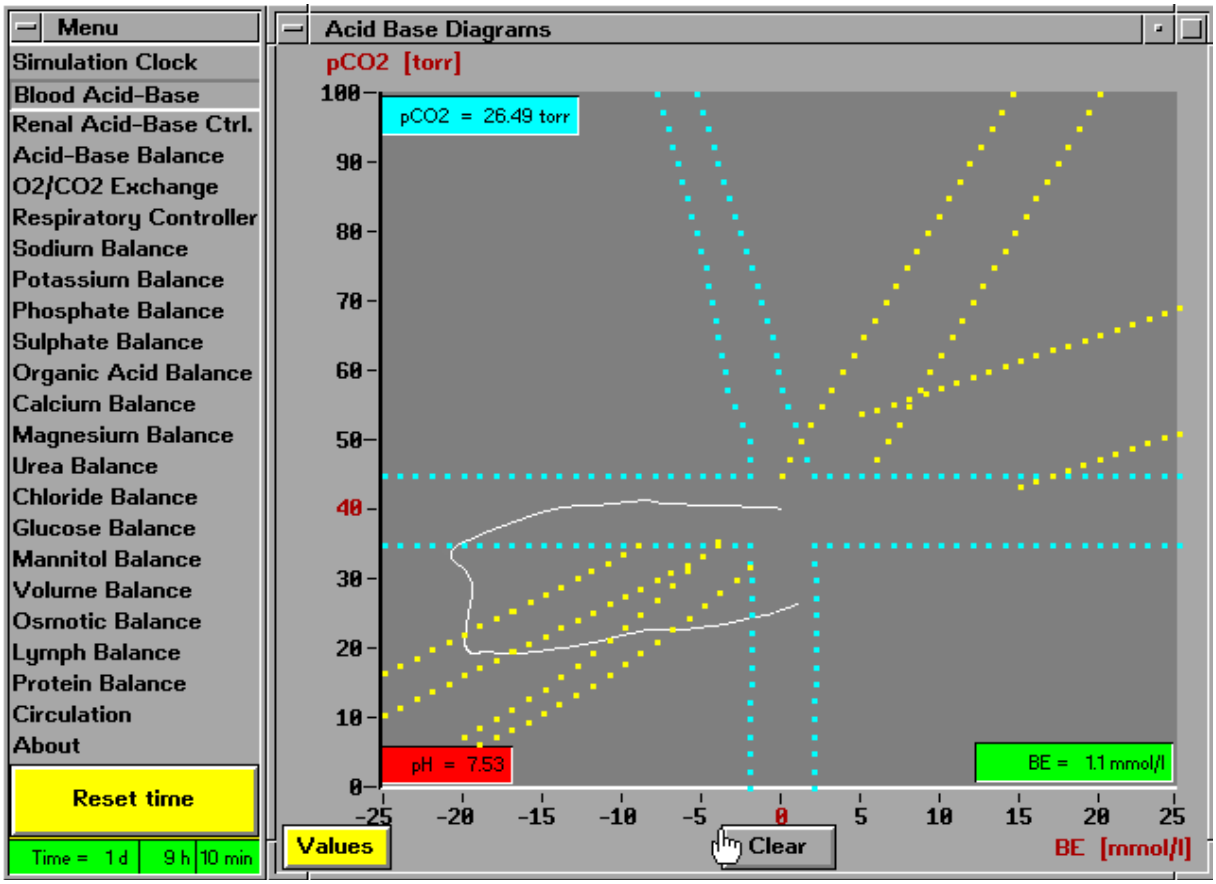
Obr. 25 – Chronická metabolická acidóza na kompenzačním diagramu.



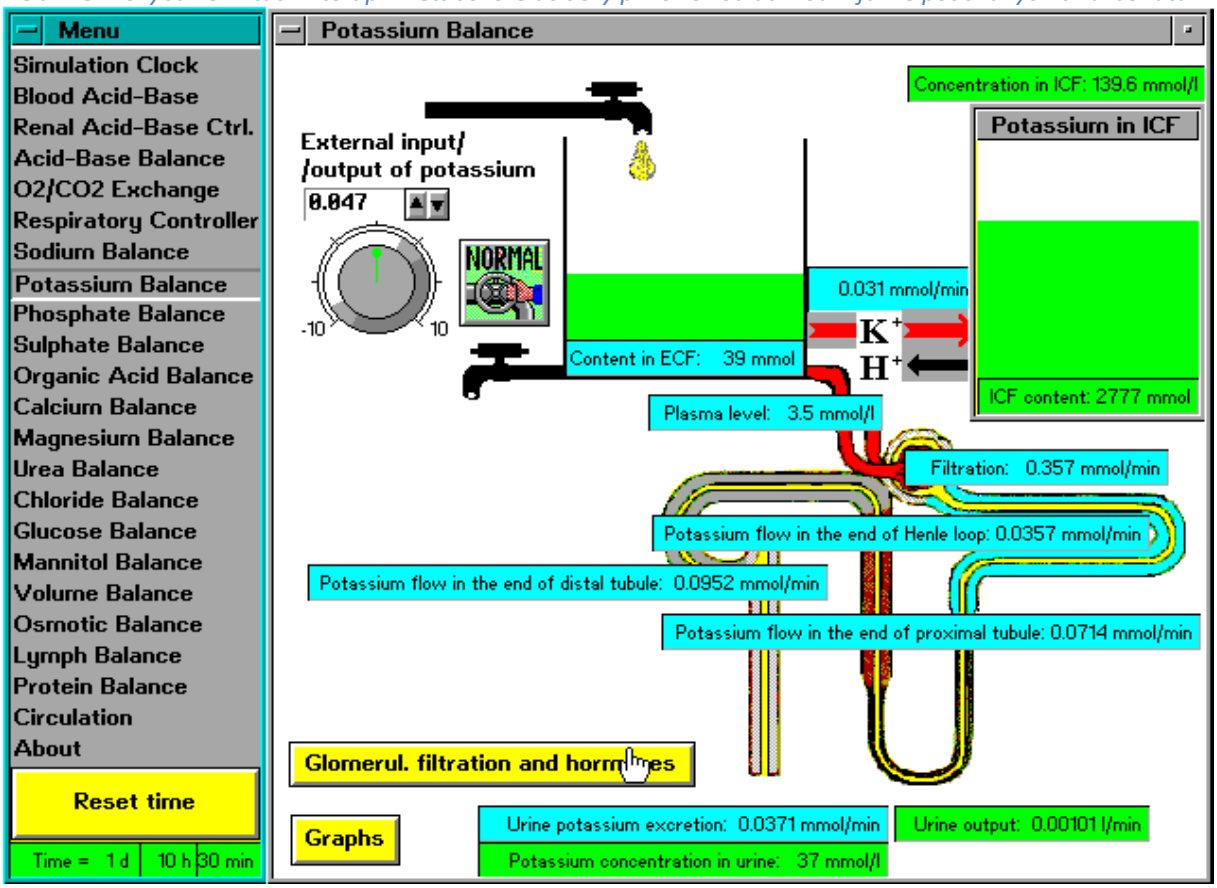
Obr. 28 – Renální korekce metabolické acidózy a její léčení infúzí bikarbonátů.



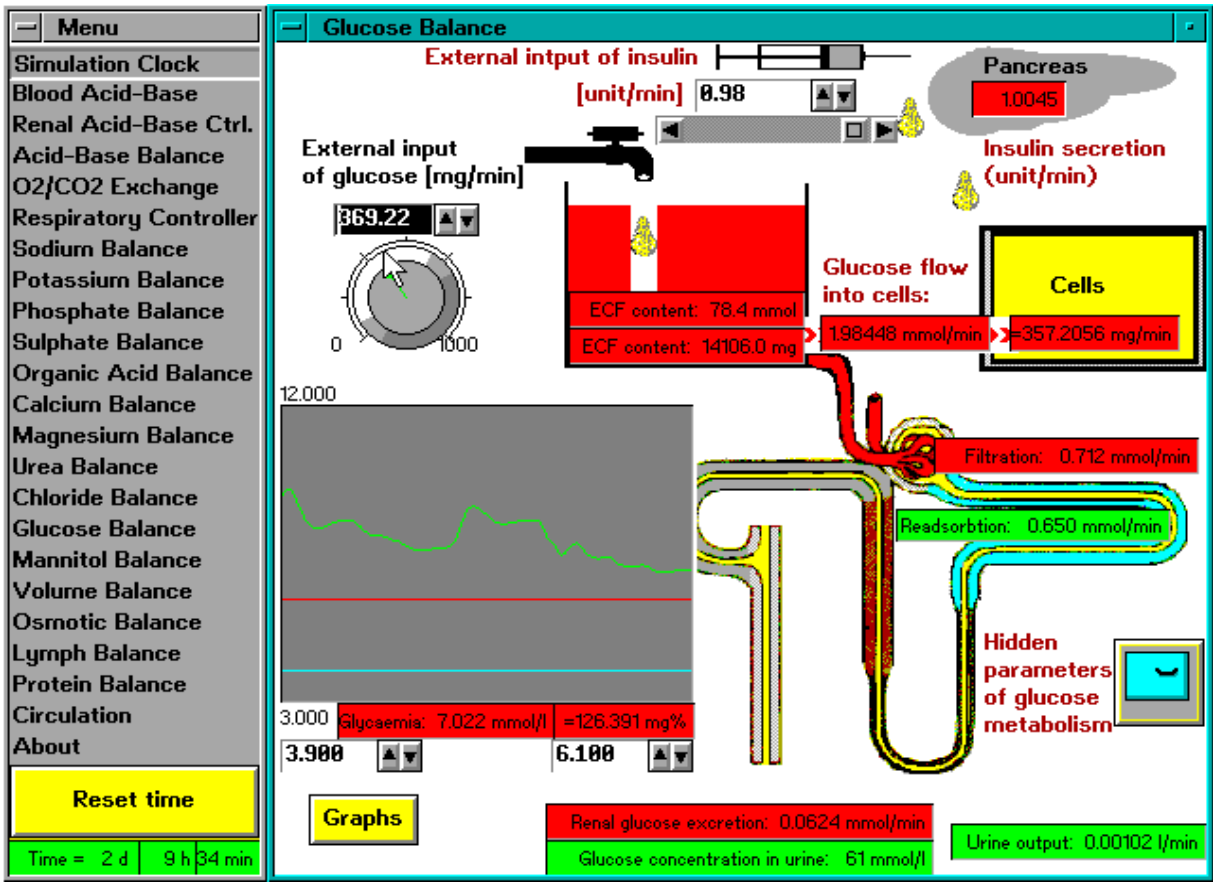
Obr. 27 – Korekce metabolické acidózy pomocí infúze bikarbonátů.



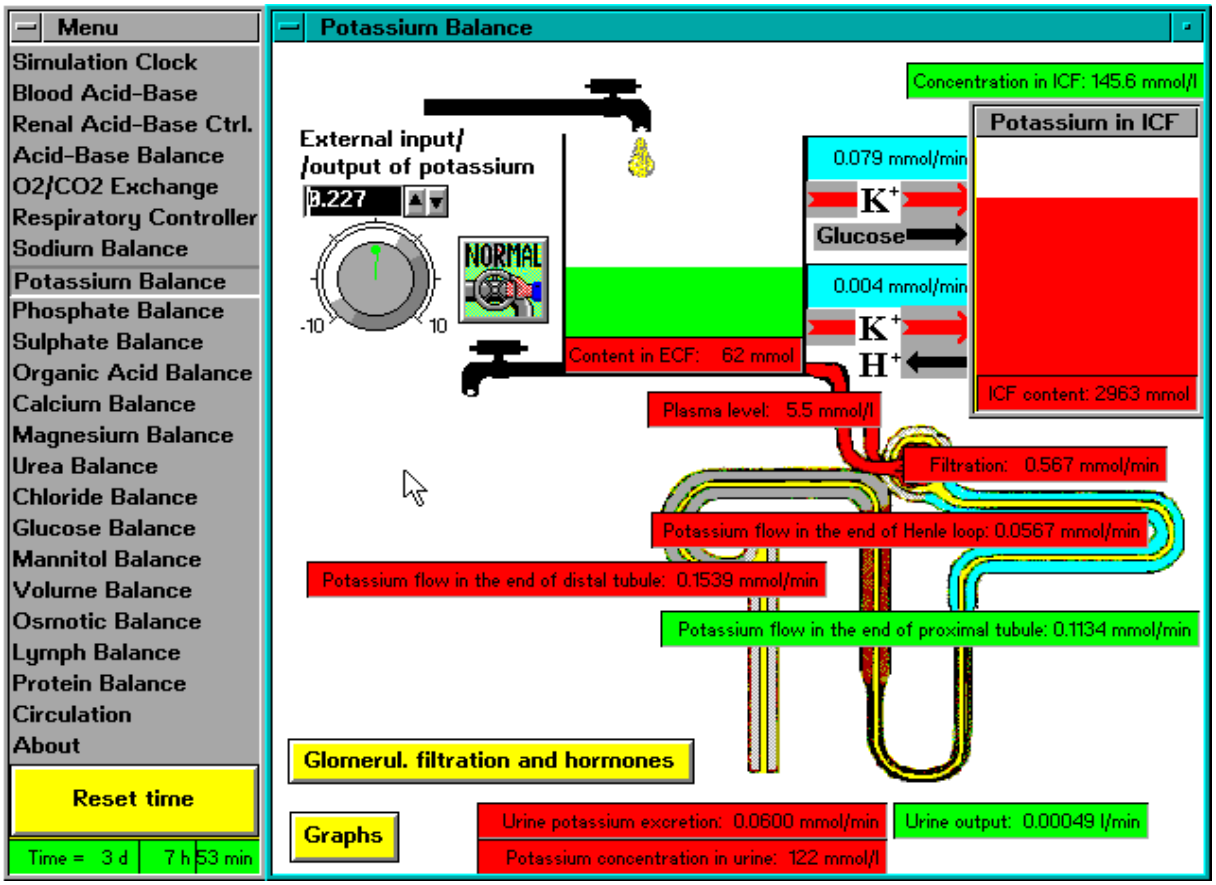
Obr. 29 – Chyba ve virtuální terapii metabolické acidózy příliš velkou dávkou infúzně podávaných bikarbonátů.



Obr. 30 – Nebezpečné následky rychlé alkalizace po dlouhodobější metabolické acidóze.



Obr. 31 – Vliv inzulínu na přesun draslíku z extracelulární tekutiny do buněk.



Obr. 32 – Infúze draselných roztoků spolu s glukózou a inzulínem pro korekci deplece draslíku.

Hodnoty pacienta by se na kompenzačním diagramu pak pomalu pohybovaly v pásmu maximální respirační kompenzace směrem k normálním hodnotám (tj. směrem ke středu kompenzačního diagramu (samozřejmě za předpokladu, že není přítomna žádná další patologie).

Renální korekci metabolické acidózy můžeme sledovat na příslušném panelu (viz obr. 27). Vidíme, že virtuální pacient zvýšil vylučování titrovatelné acidity a NH_4^+.

Aby se rychleji metabolická acidóza korigovala, byla pacientovi podávána infúze bikarbonátů. Nejjednodušším způsobem, jak tuto virtuální terapii v simulátoru realizovat, je vybrat na speciálním panelu příslušný infúzní roztok, stanovit jeho celkovou dávku a zvolit rychlost infúze. Ovládací prvek rychlosti příjmu bikarbonátů („intake rate of bicarbonate mmol/min“) na panelu acidobazické metabolické bilance se pak automaticky nastaví podle zvoleného dávkování (na obrázku 27 je u tohoto ovládacího prvku kurzorová šipka). Jiná možnost je nastavit zvolenou rychlost infúze přímo posunem ovládacího kolečka myší.

Výsledek korekce metabolické acidózy pomocí infúze bikarbonátů můžeme postupně sledovat na panelu kompenzačního diagramu (obr. 28). Vidíme, že acidobazické hodnoty se během podávání infúze v kompenzačním diagramu pomalu přesunuly z oblasti chronické metabolické acidózy do oblasti normálního pH, avšak sníženého parciálního tlaku CO_2. V tomto okamžiku by bylo vhodné infúzi přerušit. Pokud tak neučiníme, můžeme tímto razantním způsobem léčby u pacienta vyvolat nepříjemné komplikace.

Pacient je ale virtuální (i jeho smrt je jen virtuální), a proto si můžeme na simulátoru zkusit, co se stane, když pacienta předávkujeme infúzí bikarbonáty (viz obr. 29). Infúzi bikarbonátů nepřerušíme a budeme mu podávat bikarbonát až do vyrovnání hodnoty BE na normální hodnotu (0 mmol/l). Kdyby byla hodnota PCO_2 normální, hodnota pH by se normalizovala (na pH=7.4). Jenomže pacient byl v pásmu maximální respirační kompenzace (ta se vyvíjela cca 12 hodin), stejnou dobu bude ovšem tato kompenzace odeznívat. V důsledku pomalu odeznívající respirační kompenzace je hodnota PCO_2 nízká a hodnota pH se přesunula na alkalickou stranu (pH = 7.53).

To může mít neblahé důsledky pro draselný metabolismus. Ty můžeme na simulátoru sledovat na panelu bilance draslíku (viz obr. 30).

Vidíme, že rychlá alkalizace, zejména po hlubší a déle trvající acidémii, vedla k přesunu draslíku do buněk. Draslík, který se při předchozí acidémii přesunul do extracelulární tekutiny (výměnou za vodíkové ionty) se ale z podstatné části vyloučil do moči a v extracelulární tekutině již není k dispozici. Přesun draslíku do buněk proto vede k nebezpečnému poklesu hladiny draslíku v plazmě. Život pacienta je ohrožen. Situaci je nutno řešit dodáním draslíku.

Draslík ovšem musíme podávat velmi opatrně. Pokud by koncentrace draslíku v extracelulární tekutině díky jeho přísunu infúzním roztokem příliš stoupla, pacientovi by se zastavilo srdce.

Infúzi draselných iontů je zapotřebí podávat tak, aby se hladina draslíku v extracelulární tekutině příliš nezvýšila, podávaný draslík se stačil přesouvat do buněk a doplňovat tak snížené zásoby draslíku v buňkách. Pro rychlejší doplnění snížených zásob draslíku v buňkách je vhodné při podávání draselných iontů v infúzi zároveň zvýšit přesun draslíku do buněk.

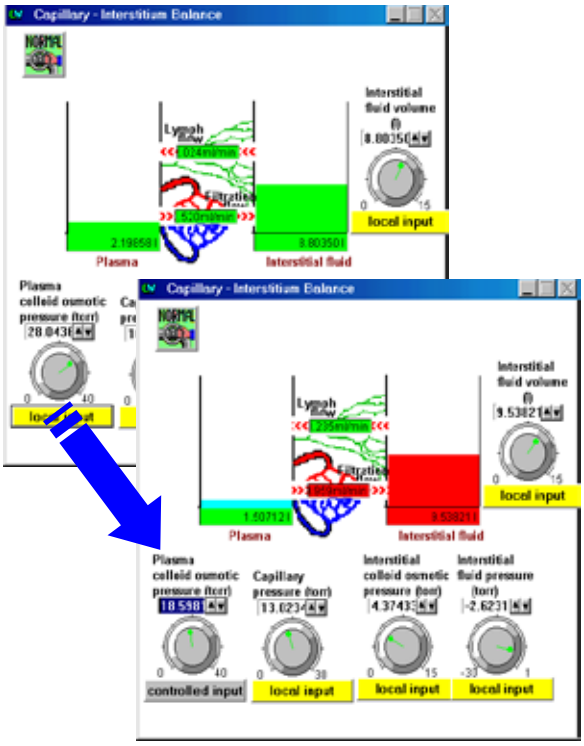
Přesun draslíku do buněk urychluje inzulín. Proto pacientovi podáme do draselné infúze inzulín spolu s glukózou (která se vlivem inzulínu nasává do buněk). Na panelu glukózové bilance (obr. 31) můžeme sledovat, jak se při infúzi glukózy s inzulínem zvýší rychlost přesunu glukózy do buněk. Dávkování glukózy a inzulínu volíme takové, abychom nevyvolali u pacienta hypoglykémii (hladinu glykémie můžeme průběžně sledovat na grafu).

Na panelu bilance draslíku (obr. 32) můžeme sledovat, jak se vlivem inzulínu se zároveň s glukózou do buněk přesouvá ve vyšší míře i v infúzi podávaný draslík. Snížené zásoby draslíku v buňkách se rychle doplňují a hladina plazmatického draslíku je i při podávané draselné infúzi držena v bezpečných mezích a stav virtuálního pacienta, kterého na životě ohrozilo předávkování alkalizačních infúzí, se zlepšil.

Simulaci bylo možné kdykoli přerušit a znovu na definovaný simulovaný čas rozběhnout. Časové průběhy hodnot jednotlivých veličin bylo možné také sledovat na speciálních panelech zobrazujících příslušné grafy časových průběhů hodnot všech relevantních veličin. Simulaci bylo možné také stiskem tlačítka „Reset“ ukončit a vše uvést do normálního stavu. Takové tlačítko, které by tímto jednoduchým

způsobem umožnilo zvládnout těžké život ohrožující stavy, ovšem u reálných pacientů neexistuje a proto simulátory slouží jako užitečná výuková pomůcka pro nácvik lékařského diagnostického a terapeutického rozhodování bez nebezpečí pro pacienta. Vše se odehrává ve virtuálním světě, kde i smrt je jen virtuální…

Pro simulátor byla připravena celá sada scénářů, realizujících nejrůznější kombinované poruchy homeostázy vnitřního prostředí – poslední verze simulátoru obsahovaly například „receptář“, kde bylo možné zadat velikost, rychlost a časový interval příslušných ztrát tělních tekutin, a nastavení hodnot příslušných parametrů, kterými bylo možné simulovat příslušné poruchy fyziologických regulací při nejrůznějších onemocněních. Byl vytvořen i receptář pro dávkování příslušné terapie - studenti si např. mohli vybírat pro infúzní terapii z řady nabídnutých infúzních roztoků a simulátor také umožňoval do příslušného receptáře přidávat další roztoky.



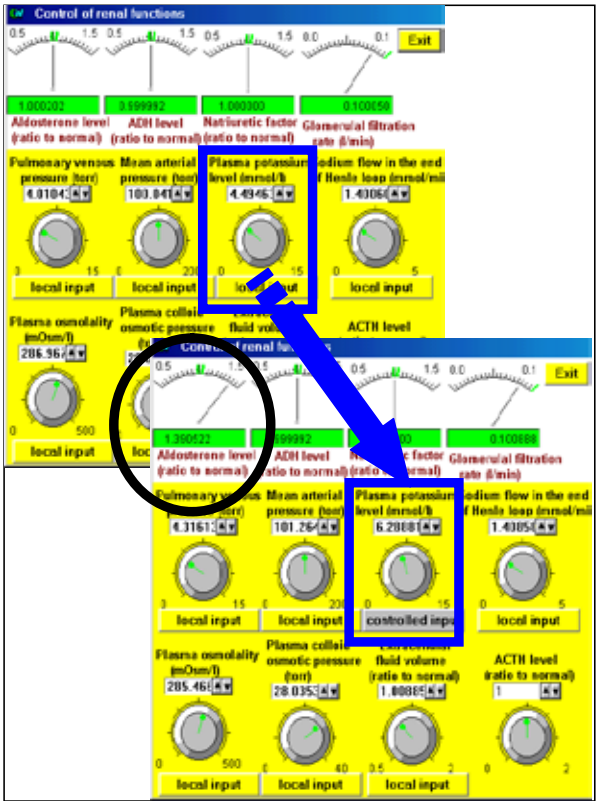
Obr. 33 – Sledování vlivu koloidně-osmotického tlaku plazmatických bílkovin na rovnováhu filtrace a zpětné resorpce vody mezi kapilárou a intersticiální tekutinou na panelu kapilárně-intersiciální rovnováhy. Hodnoty všech veličin zobrazených na tomto panelu jsou regulované veličiny zobrazované v příslušných okénkách. Jejich hodnoty nelze na panelu měnit (příslušné regulační knoflíky pod displejem na počítačovou myš nereagují). Stiskem příslušného tlačítka však můžeme regulační smyčku rozpojit a hodnotu příslušné veličiny můžeme zadat jako lokální vstup a sledovat jak se její změna projeví. V daném příkladě byla v stiskem tlačítka přepnuta hodnota koloidně osmotického tlaku plazmatických bílkovin do režimu lokálního vstup. V simulátoru můžeme pozorovat, jak snížení její hodnoty vedlo ke zvýšenému přestupu vody z plazmy do intersticiální tekutiny a k následnému vzestupu objemu intersiciální tekutiny a vznik otoků. Na dalších panelech simulátoru Golem můžeme sledovat jak na tuto situaci budou reagovat další fyziologické subsystémy organismu nebo můžeme celý subsystém kapilárně-intersticiální rovnováhy vody na odpojit z regulací a sledovat jaký význam mají jednotlivé proměnné na vznik a resorpci otoků.

3.3 Golem na pitevním stole – přerušování a opětovné zapojování regulačních vztahů

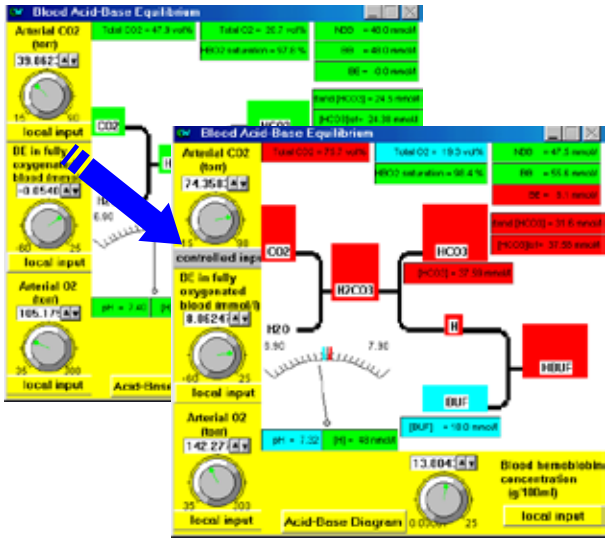
Při výuce anatomie je pitva užitečným výukovým prostředkem, umožňujícím studentům medicíny postupným odstraňováním jednotlivých anatomických struktur si ozřejmit základní anatomické souvislosti stavby lidského těla. Pro podporu výuky anatomie existuje též i řada na internetu dostupných nástrojů, umožňujících vizualizovat anatomické poměry na obrazovce počítače a lidské tělo tak pitvat ve virtuálním prostředí (viz např. http://www.visiblebody.com/).

Obdobně, jako při anatomické pitvě pro ozřejmení anatomické stavby organismu odstraňujeme jednotlivé části orgánů tak i při sledování funkčnosti organismu na simulátoru se z pedagogického hlediska ukázalo jako velice výhodné objasňovat fyziologický význam jednotlivých regulačních okruhů pomocí ***rozpojování a opětovného zapojování jednotlivých regulačních vazeb***. Rozpojení regulačních vazeb v simulátoru umožní ***lokálně sledovat odezvu jednotlivých fyziologických subsystémů*** na změnu hodnot některých proměnných, které jsou ale sami v organismu regulovány.

Do simulátoru Golem jsme proto zavedli ***možnost „odpojení regulace“*** některých regulovaných fyziologických proměnných a jejich „přepnutí na lokálně zadávaný vstup“. Stačí pouze pod okénkem příslušné proměnné (která je vstupem do daného subsystému a jejíž hodnota je zvnějšku subsystému regulována) stisknout příslušné tlačítko (v anglické verzi označené nápisem *„local input“*). Proměnná se rázem odpojí z vnější



Obr. 34 – Sledování vlivu některých fyziologických proměnných na hormony ovlivňující funkci ledvin. V panelu zobrazujícím pomocí ručkových měřících přístrojů hodnotu glomerulární filtrace a hladiny hormonů ovlivňujících renální funkce (tj. aldosterunu, ADH, natriuretického hormonu) jsou pod měřícími přístroji zobrazeny hodnoty některých regulovaných fyziologických proměnných. Tyto hodnoty můžeme (stiskem tlačítka s nadpisem „lokální vstup", resp. v anglické verzi simulátoru „local input") odpojit od regulace a přepojit je do režimu lokálního vstupu a nastavit novou hodnotu těchto proměnných. Tak se například na zobrazeném příkladě se odpojila od regulace hodnota plazmatické koncentrace draslíku a z původní (regulavané) hodnoty 4,49 mmol/l se ručně přenastavila na hodnotu 6,29 mmol/l. Na panelu můžeme sledovat, že nadledviny následně zvýší sekreci aldosteronu. Na jiném panelu pak např. můžeme sledovat, jak se zvýšená hladina aldosteronu projeví na zvýšení vylučování draslíku ledvinami. Tímto způsobem simulátor Golem umožňuje studentům sledovat význam jednotlivých fyziologických regulačních okruhů a lépe pochopit odezvy jednotlivých fyziologických subsystémů na nejrůznější patologické i terapeutické podněty.

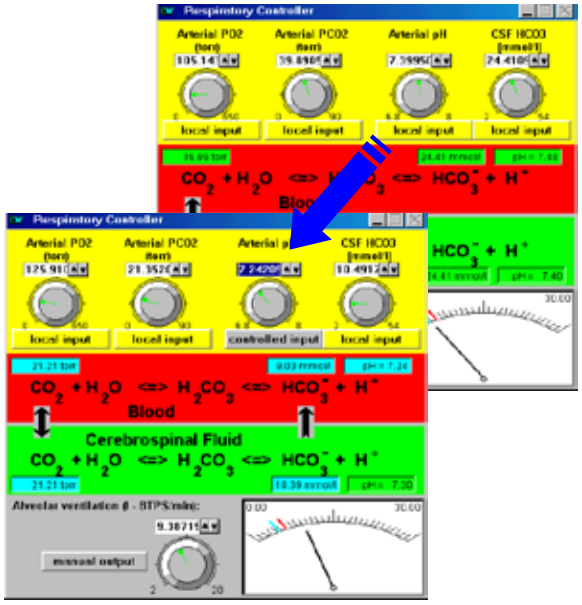


Obr. 35 – Přerušení regulační smyčky do pufračních subsystémů krve. Stiskem tlačítka byla rozpojena regulační smyčka řídící hodnotu pCO_2 v arteriální krvi – nyní na ní již nemá vliv respirační systém a hodnotu arteriálního PCO_2 můžeme zadávat jako lokální vstup. Můžeme např. jeho hodnotu zvýšit a sledovat jak na zvýšení hladiny pCO_2 bude reagovat pufrační systém krve. Vidíme, že hladina bikarbonátů stoupá a zároveň hladina nebikarbonátových bazí (BUF^-) klesá, hodnota pH (zobrazená na ručičkovém měřícím přístroji) se přechýlí na kyselou stranu. Stiskem dalších (žlutě vybarvených) tlačítek můžeme z regulačních smyček odpojit i všechny ostatní vstupy do systému a podrobně sledovat chování samotného pufračních systémů krve při změnách jednotlivých vstupních veličin. V daném případě jsme další vstupy neodpojili a můžeme proto sledovat, jak se vlivem postupného zapojení renálního regulátoru organismus postupně (s maximem odpovědi během 3 - 5 dnů simulovaného času) snaží o kompenzaci kyselého pH zvýšením vylučování silných kyselin ledvinami (ve formě titrovatelné acidity a NH_4^+).

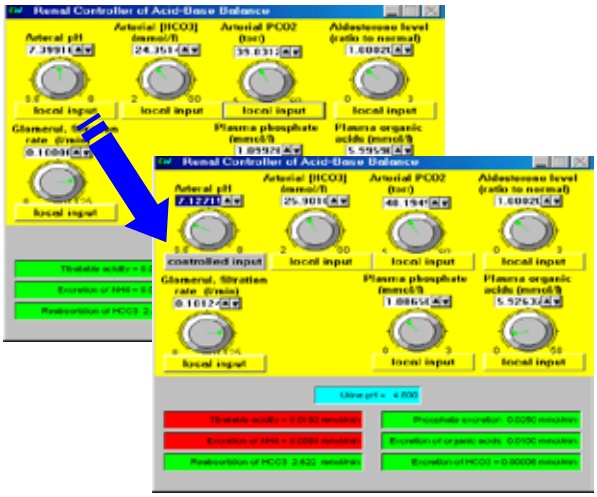
regulační smyčky. Její hodnotu pak můžeme měnit a sledovat odezvy subsystému na zadávané hodnoty. Jakmile opět stiskneme příslušné tlačítko (jehož nadpis se změnil na *„controlled input"*) proměnnou opět propojíme do regulační smyčky.

Rozpojení regulačních smyček umožní omezit se při simulaci na jednotlivý fyziologický subsystém a zkoumat jeho chování nezávisle na spletitých regulačních vztazích uvnitř celého organismu a sledovat tak chování jednotlivých fyziologických regulačních vztahů odděleně.

Tak např. můžeme sledovat, jaký vliv na přesuny vody mezi kapilárou a intersticiální tekutinou má koloidně osmotický tlak plazmatických bílkovin, kapilární tlak, nebo koloidně osmotický tlak intersticia či hydraulický tlak intersticiální tekutiny (obr. 33). Všechny tyto veličiny jsou organismem regulovány. Pokud odpojíme od regulace koloidně osmotický tlak v plazmě (který závisí na organismem regulované koncentraci plazmatických bílkovin), můžeme sledovat, jak se při jeho poklesu přesune část tekutiny z plazmy do intersticiálního prostoru.



Obr. 36 – Sledování chování respiračního regulátoru na poruchy acidobazické rovnováhy. Na schématu jsou znázorněny bikarbonátové pufrační systémy v krvi a v mozkomíšním moku, oddělené hematoencefalickou bariérou. CO_2 proniká přes hematoencefalickou bariéru snadno (proto jsou jeho hladiny v moku a v krevní plazmě prakticky vždy vyrovnané) zatím co bikarbonát (HCO_3^-) proniká hematoencefalickou bariérou pomalu (a proto mohou být jeho hodnoty a krevní plazmě a v moku rozdílné). Rozpojením hodnoty arteriálního pH od regulace můžeme tuto hodnotu zadávat jako lokální vstup a sledovat reakci respiračního regulátoru na změnu arteriálního pH. Pokud hodnotu pH snížíme z 7.4 na 7.24, jako je ukázáno na obrázku, v krevní plazmě okamžitě poklesne hladina bikarbonátů (HCO_3^-). Hladina CO_2 v mozkomíšním moku je prakticky neustále ekvilibrována s hladinou CO_2 v krvi, zatímco hladiny bikarbonátů v krvi a moku se vyrovnávají pomalu. Na simulátoru můžeme sledovat, jak bikarbonát z mozkomíšního moku pomalu přechází do krve, jeho hladina v moku postupně klesá. Tím se ale posouvá chemické ekvilibrium pufrační reakce v bikarbonártovém systému moku směrem doprava, koncentrace vodíkových iontů v moku stoupá, pH moku klesá. Tím se ale dráždí respirační centrum a hodnota alveolární ventilace (zobrazená na ručkovém měřidle v dolní části obrázku) postupně (během 12 hodin simulovaného času) stoupá.



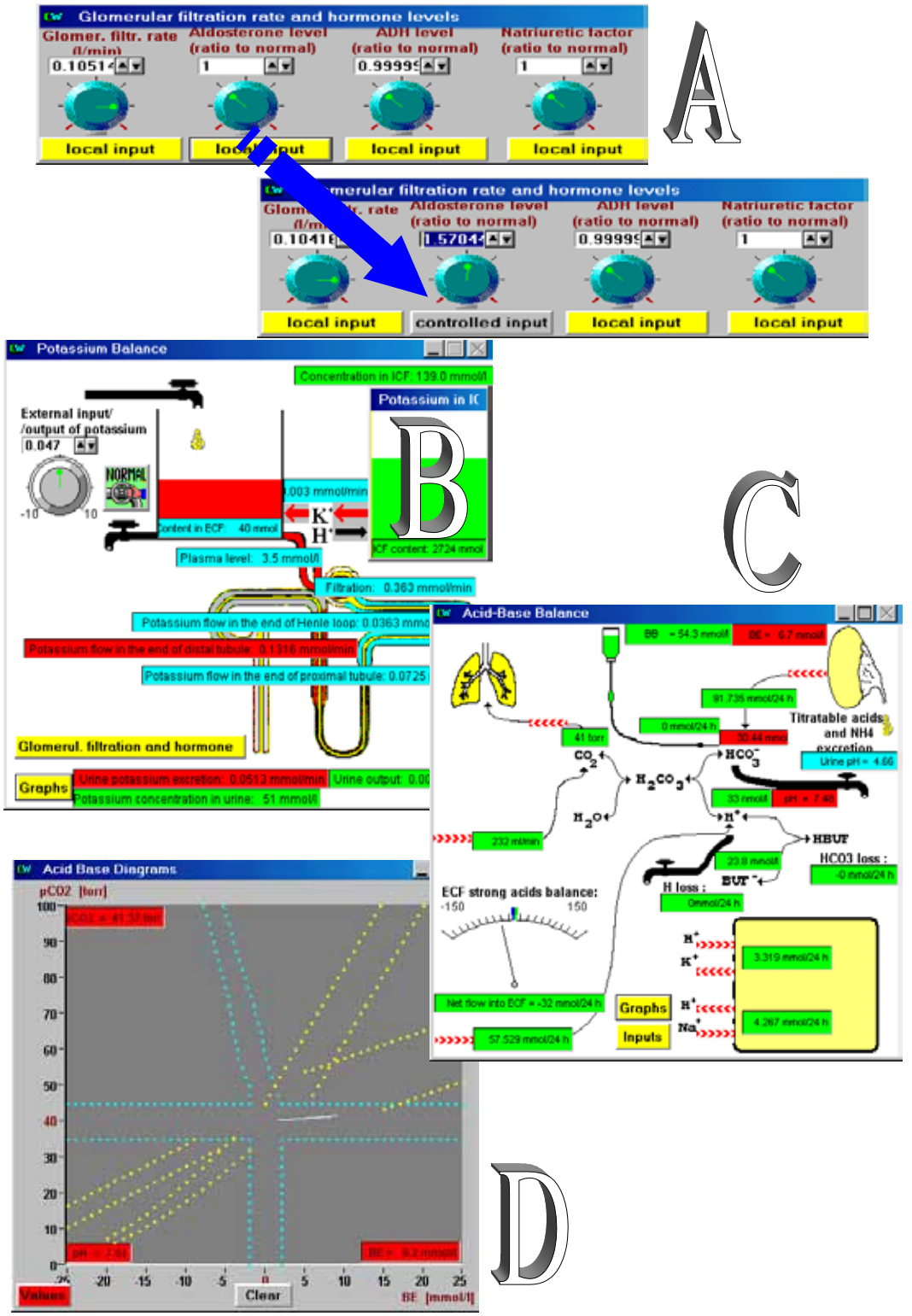
Obr. 37 – Sledování chování renálního regulátoru acidobazické rovnováhy. Renální regulátor má význam pro regulaci bilance mezi tvorbou a vylučováním silných kyselin. Stiskem tlačítka se přeruší vazba renálního regulátoru na jeho vnější okolí přes hodnotu arteriálního pH, kterou je pak možné zadávat jako lokální vstup do renálního regulátoru. Snížíme-li hodnotu arteriálního pH můžeme sledovat, jak se ledviny budou snažit posunout bilanci směrem ke zvýšení vylučování silných kyselin. Vidíme, jak se postupně (během 3-5 dní simulovaného času) zvyšuje exkrece vodíkových iontů vázaných na fosfáty (ve formě titrovatelné acidity) a na amoniak (ve formě NH_4^+). Hodnota pH v moči se sníží.

Na simulátoru můžeme mimo jiné také zkoumat vliv jednotlivých proměnných na výdej příslušných hormonů – tak například rozpojením regulační smyčky řídící koncentraci plazmatické hladiny draslíku můžeme hladinu draslíku v plazmě zadávat jako vstup a sledovat, jak zvýšení hladiny draslíku změní výdej aldosteronu (viz obr. 34) a na panelu ledvin pak sledovat jak zvýšená hladina aldosteronu ovlivní jeho vylučování v ledvinách.

Od vnějších vazeb můžeme dokonce odpojit celý subsystém a sledovat dynamiku chování daného subsystému při postupných změnách jediného vstupu, zatímco jiné vstupy jsou nastaveny na zvolené konstantní hodnoty (tzv. princip „ceteris paribus“).

Tak například, pro pochopení mechanismu regulace acidobazické rovnováhy můžeme sledovat chování pufračních systémů krve, odpojených z vnější regulace, na změnu hladiny oxidu uhličitého (viz obr. 35). Rozpojováním regulačních smyček můžeme také podrobně zkoumat, čím jsou ovlivňovány respirační a renální regulátor acidobazické rovnováhy (obr. 36 a 37).

Odpojováním jednotlivých regulátorů můžeme tedy odděleně sledovat chování samostatných regulačních smyček a dílčích fyziologických subsystémů. Tím simulátor Golem nahrazuje celou škálu modelů dílčích fyziologických subsystémů a může přispět k pochopení jednotlivých fyziologických regulačních vazeb.



Obr. 38 – Příklad rozpojení fyziologických subsystémů z vnějších regulačních smyček v simulátoru GOLEM demonstrující vazby subsystému acidobazické rovnováhy a elektrolytové rovnováhy přes regulační vliv hormonální regulace. Stisknutím tlačítka odpojíme řízení hladiny hormonu aldosteron z regulace. Pootočením knoflíku pak hladinu aldosteronu násilně zvýšíme (A). Po čase vidíme na schématu vylučování draslíku v ledvinách, že draslík se ve zvýšené míře vylučuje močí, jeho zásoby v plazmě jsou však malé, ledvinné ztráty vedou k tomu, že draslík se přesouvá do mimobuněčné tekutiny (a plazmy) z buněk (B). Výstup draslíku z buněk je však provázen vstupem vodíkových iontů do buněk (C), důsledkem je rozvoj extracelulární metabolické acidózy (D).

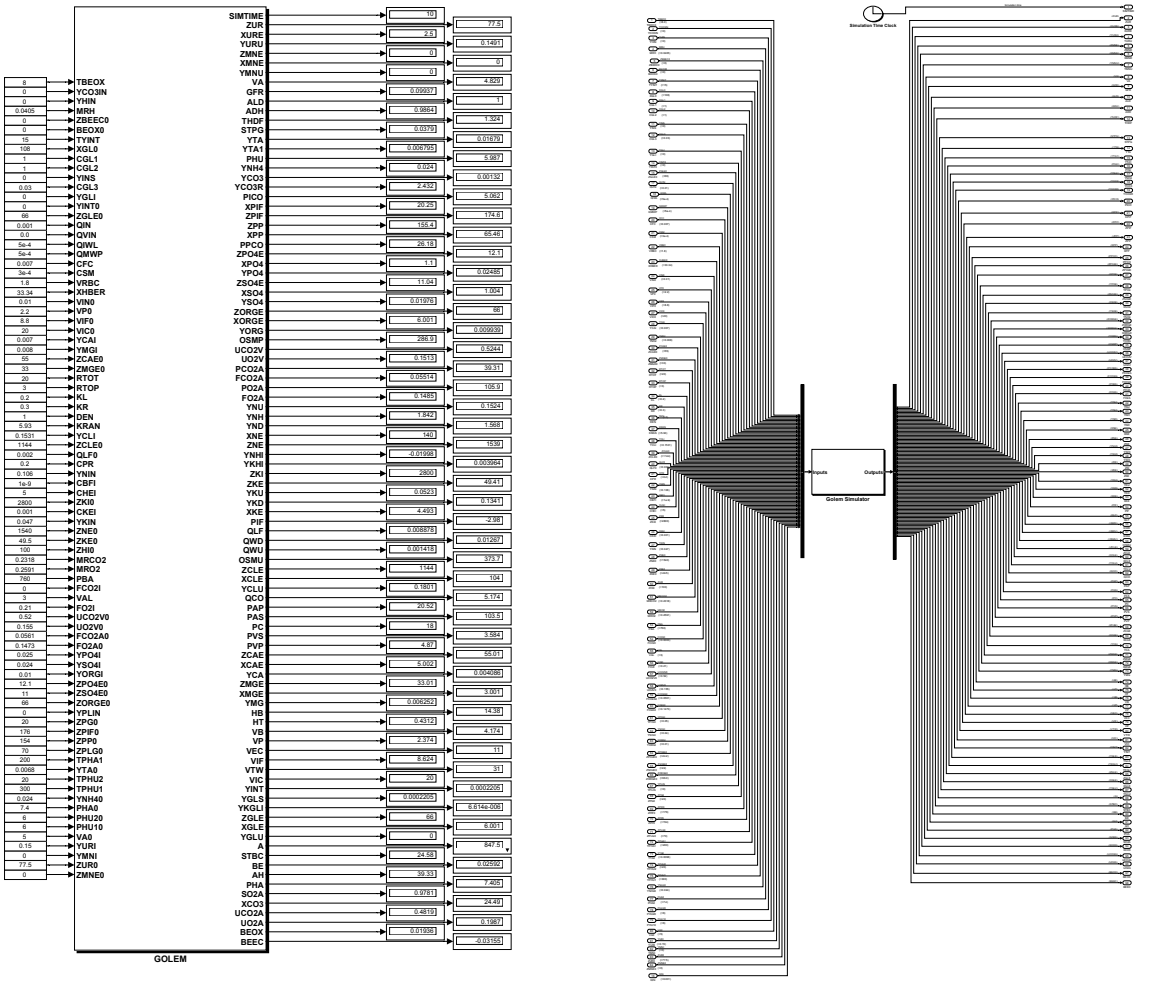
Pro pochopení patofyziologie řady klinických stavů je však důležité pochopení vzájemných souvislostí jednotlivých fyziologických subsystémů. Rozpojením regulační smyčky řízení výdeje aldosteronu můžeme například sledovat hormonální ovlivnění subsystémů jak acidobazické tak i elektrolytové rovnováhy (viz obr. 38).

Možnost rozpojování a zapojování regulačních smyček umožňuje studentům pomocí simulátoru si ozřejmit význam jednotlivých regulačních okruhů a studovat vliv (rozpojených a zprvu ručně řízených) regulačních vazeb na chování organismu při nejrůznějších patologických poruchách a reakcích na příslušnou terapii.

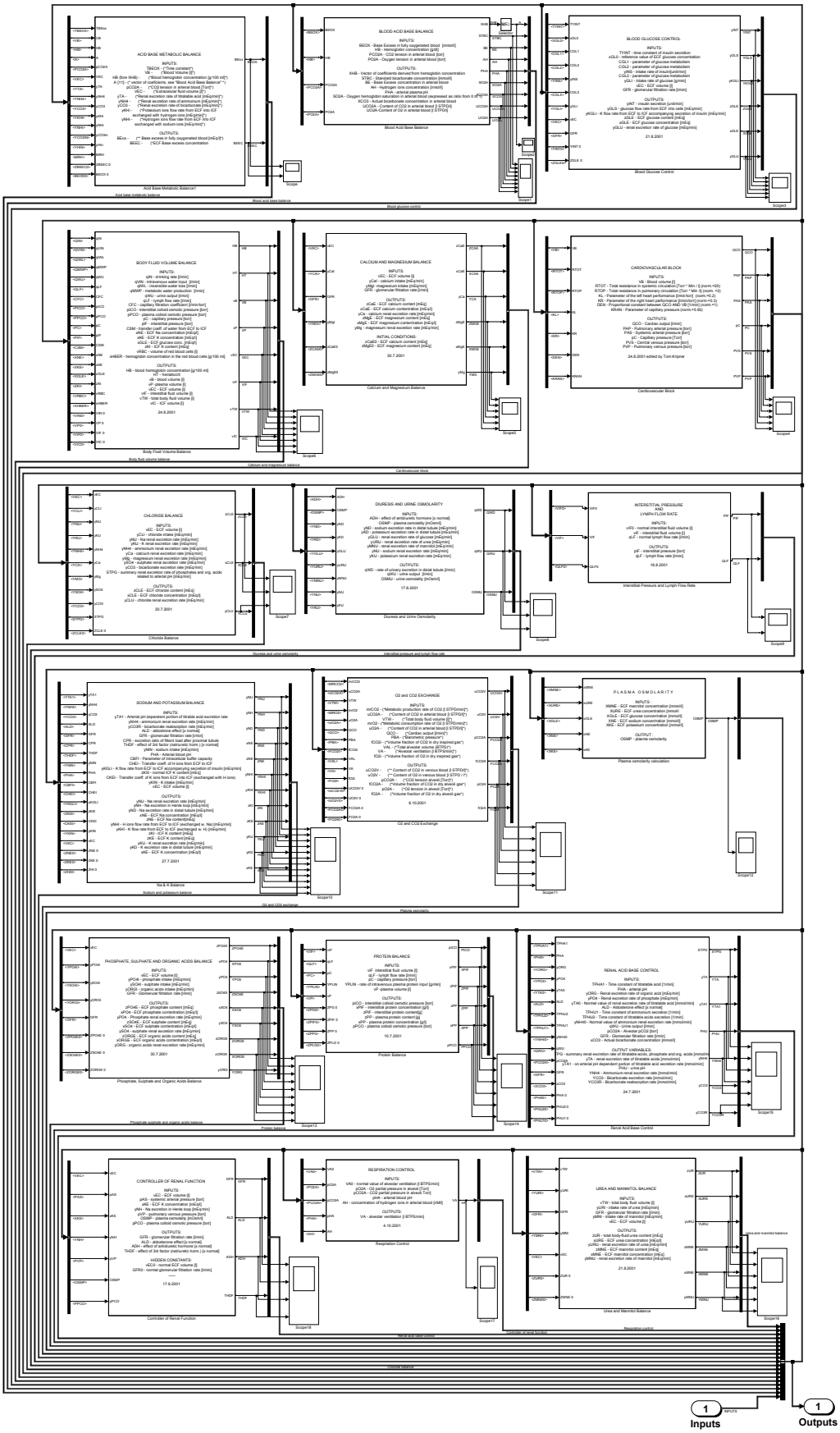
Podle našich zkušeností právě tento přístup vede ***k lepšímu pochopení významu jednotlivých regulačních smyček,*** k porozumění jejich úlohy v patogeneze nejrůznějších onemocnění a chápání patofyziologických principů příslušných léčebných zásahů.

3.4 Šém pro Golema – matematický model v Simulinku

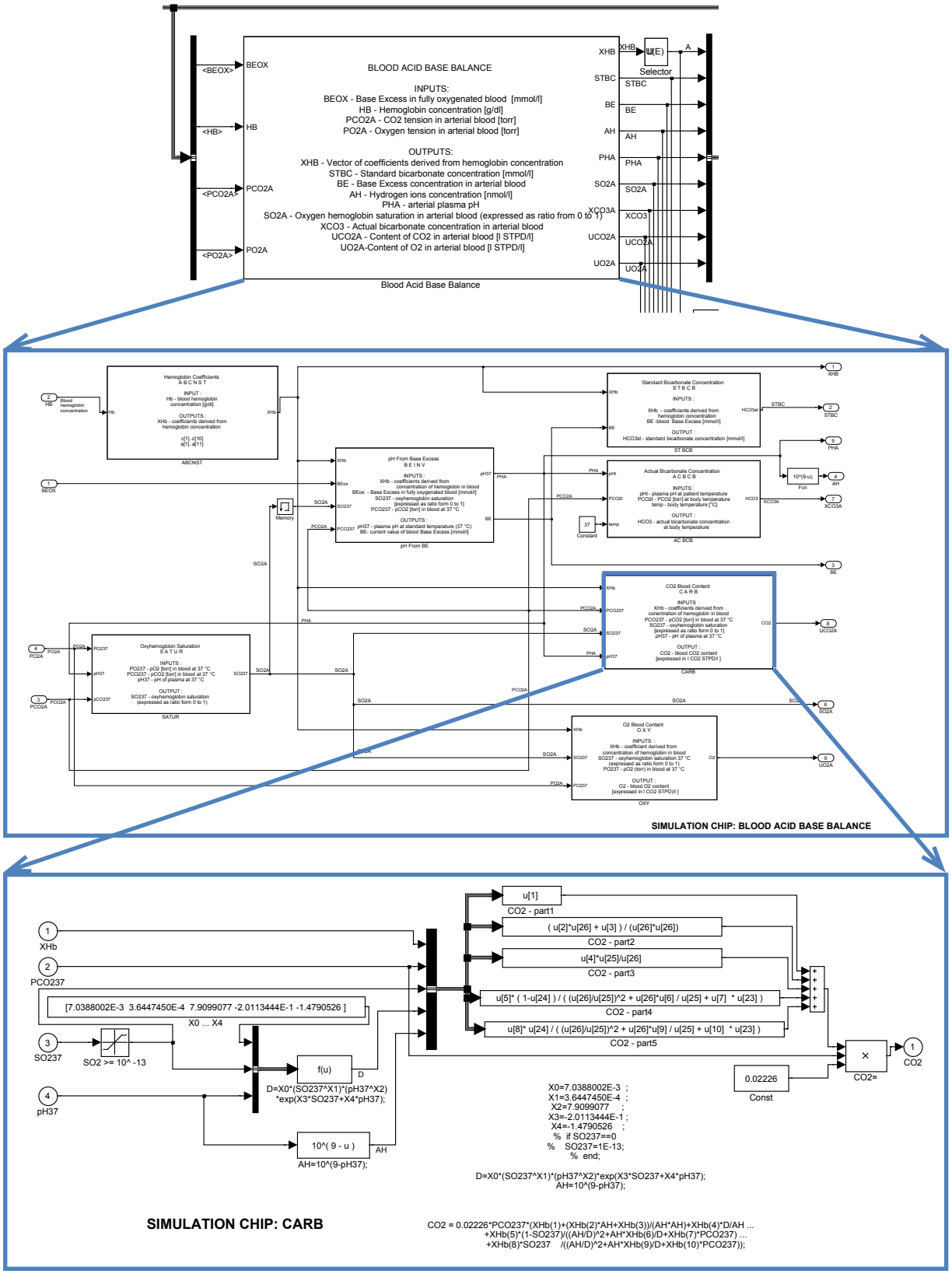
Mýtického Golema – hliněného umělého člověka, kterého podle legendy stvořil Jehuda Löw, oživoval „šém ha-meforaš“ – znamení Božího jména zapsané na vnějším hliněném nosiči. Šém po vložení do Golema hliněnou bytost údajně probouzel k životu. Magickým „šémem“, který oživuje simulátor Golem je matematický model. Matematický model, který je podkladem k simulátoru, byl vytvářen v prostředí Matlab/Simulink od firmy Mathworks.



Obr. 39 – Matematický model, který je podkladem pro simulátor Golem byl implementován jako speciální, hierarchicky uspořádaný simulinkový blok, který je součástí veřejně přístupné knihovny Physiolibrary (http://www.physiome.cz/simchips). Svojí strukturou připomíná elektronický čip.



Obr. 40 – Vnitřní struktura simulinkového bloku, realizujícího simulační model Golem, se skládá z 18 bloků jednotlivých fyziologických subsystémů, jejichž vstupy a výstupy jsou propojeny přes společnou sběrnici.



Obr. 41 – Celý model simulátoru Golem je vytvořen z hierarchicky uspořádaných, navzájem propojených simulinkových bloků. Jejich propojení zobrazuje příslušné závislosti a vazby mezi jednotlivými fyziologickými subsystémy. Struktura simulinkové počítací sítě, vyjadřující matematické vztahy, se objevuje teprve na nejnižší hierarchické úrovni. Tak například unvitř bloku BLOOD ACID BASE BALANCE jeden z propojených bloků - (blok CARB), počítající celkovou koncentraci oxidu uhličitého v krvi, již obsahuje vlastní počítací bloky.

Zdrojový text modelu v Simulinku je součástí námi vyvinuté knihovny Physiolibrary a je jako „open source“ volně k dispozici na adrese: http://www.physiome.cz/simchips. Model v Simulinku má hierarchickou strukturu – složenou z jednotlivých vnořených a propojených bloků připomínajících elektronické čipy (Kofránek, Andrlík, Kripner & Mašek, 2002b).

Celý model je realizovaný jako jeden hierarchicky uspořádaný simulinkový blok (viz obr. 39). Po jeho „rozkliknutí“ se objeví celá struktura modelu tvořená 18 propojenými simulinkovými bloky, které reprezentují jednotlivé subsystémů (obr. 40).

Hierarchická struktura jednotlivých bloků i propojení byla volena tak, aby modelované vzájemné souvislosti byly srozumitelné i pro fyziology. V masce každého bloku jsou popsány významy všech vstupních a výstupních i proměnných i fyzikálních jednotek, v nichž jsou vyjádřeny jejich hodnoty. Propojení jednotlivých bloků tak názorně ukazuje, které proměnné a jaké fyziologické souvislosti jsou v modelu uvažovány.

Příslušné matematické vztahy jsou skryty uvnitř propojených bloků. K těmto vztahům je možné se dostat postupným „rozklikáváním“ hierarchicky organizovaných simulinkových bloků. Terve na nejhlubší úrovni se nakonec zobrazí struktura simulinkové počítací sítě, realizující příslušné matematické výpočty podle rovnic modelu (obr. 41).

Celý model i všechny vnořené bloky jsou zároveň součástí volně šiřitelné simulinkové knihovny Physiolibrary. Jednotlivé bloky se dají využít samostatně i v jiných modelech. Snažili jsme se, aby model byl svou strukturou samodokumentující.

Připojením definovaných vstupů k bloku a sledováním průběhu výstupů ke každému samostatně vytaženému bloku z knihovny může fyziolog zkoumat adekvátnost jeho chování. V připojené nápovědní stránce k bloku je popsán modelovaný subsystém podrobněji, včetně příslušných matematických vztahů.

3.5 Jen Simulink nestačí

Simulátor ve formě hierarchicky uspořádaného simulinkového bloku je sice vhodným nástrojem pro odbornou komunikaci prostřednictvím volně stažitelné knihovny, nicméně pro vytvoření výukového simulátoru pro výuku mediků to samo o sobě nestačí. Simulační nástroje firmy Mathworks jsou totiž určeny pro specialisty a pro běžného uživatele, který si chce se simulačním modelem jen „pohrát“, se příliš nehodí. I když v prostředí těchto nástrojů je možné naprogramovat poměrně příjemné uživatelské rozhraní k ovládání vytvořeného modelu, ***pro účely uplatnění simulačního modelu ve výuce medicíny je toto rozhraní až příliš komplikované.***

Kromě toho – za pohodlí prostředí, určeného především pro vytváření (a nikoli provozování) simulačních modelů se platí tím, že u rozsáhlých modelů (a námi vytvořený model mezi ně patří) jsou nároky na výpočetní výkon počítače poměrně vysoké. Na méně výkonných počítačích pak simulace probíhá neúměrně pomalu.

Proto bylo nutno na základě odladěné a verifikované struktury simulačního modelu zvlášť naprogramovat vlastní simulátor včetně jeho uživatelského rozhraní. Implementovaný simulační model v něm počítá rychleji a možnosti uživatelského ovládání jsou pro netechniky (tj. studenty medicíny a lékaře) podstatně přirozenější než v prostředí Matlabu a Simulinku.

Pro vývoj simulátoru jsme proto využili jiné vývojové prostředky než Matlab a Simulink: po určitém váhání, spíše než po obecných softwarových nástrojích (typu Delphi, Visual C++ aj.) jsme se nakonec poohlédli po ***nástrojích využívaných při tvorbě průmyslových aplikací*** (měřících ústředen a velínů). Vedly nás k tomu především tři důvody:

1. ***Se simulačním modelem chceme v simulátoru fyziologických funkcí zacházet obdobně jako se v průmyslu řídí složité technologické zařízení***: chceme číst (a v nejrůznější grafické či číselné podobě zobrazovat) množství nejrůznějších měřených dat (jako v průmyslové měřící ústředně) a zároveň chceme jednoduchým způsobem (stiskem tlačítek, otáčením knoflíků popotahováním táhel apod.) simulační model ovládat (obdobně, jako se z velínu řídí nějaká technologie).

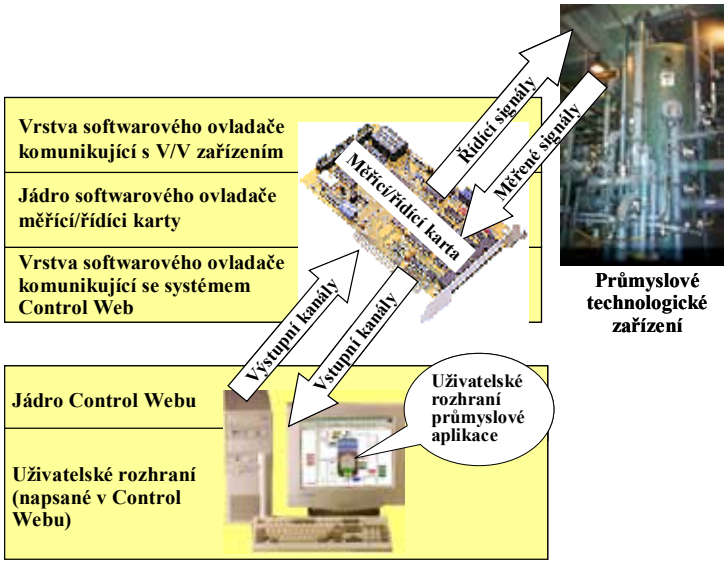
2. Průmyslové řídící a vizualizační aplikace jsou ***náročné jak na grafický, tak i na numerický výkon***. Na rozdíl od kancelářských aplikací, kde se numerický výpočet může na chvíli pozastavit, dokud se nevykreslí grafika, mají potřebné numerické výpočty v průmyslových řídících a vizualizačních aplikacích obvykle vyšší prioritu než grafické vykreslování výsledků. Obdobně, numericky náročný běh modelu na pozadí simulátoru je stejně důležitý jako vizualizace chování modelu v jeho grafickém uživatelském rozhraní.
3. Posledním, ale důležitým důvodem, proč jsme sáhli po softwarovém nástroji z průmyslu, je ***spolehlivost***. Požadavky spolehlivosti, kladené na nástroje, jejichž pomocí se vyvíjejí průmyslové řídící aplikace, jsou obvykle řádově vyšší než u obecných programovacích nástrojů.

Nástrojů pro design průmyslových aplikací je na světovém trhu nemálo. Jejich ceny jsou ovšem, na rozdíl od obecných softwarových nástrojů, zpravidla velmi vysoké. Jedním z rozšířených vývojových nástrojů pro práci v prostředí Windows je vývojové prostředí LabView od amerického výrobce National Instruments, často využívané pro vizualizaci měřených dat z připojeného experimentálního zařízení (http://www.ni.com/labview). LabView se ale využívá i v některých lékařských výukových simulátorech. Jejich podkladem je matematický model, naprogramovaný v prostředí LabView (např. Jackson & Gnadt, 1999, Davis, 2001, Davis & Gore, 2001, Lipovszki & Aradi, 2006, Felipini, de Adrande, Lucchi, da Fonseca & Nicolosi, 2008) nebo také připojený fyzikální model. Tak např. nedávno Anderson a spol. vytvořili výukovou aplikaci, jejíž podstatou je mechanický simulátor imitující plíce. Fyzikální model je připojen k počítači, v prostředí LabView je naprogramováno jeho řízení a vizualizace snímaných dat. Model je využíván pro výuku fyziologie a patofyziologie plicní mechaniky (Anderson, a další, 2009).

LabView je určitým, svého druhu průmyslovým, standardem pro řízení a vizualizaci technologických procesů prostřednictvím personálního počítače. Nicméně i v České republice existuje výrobce softwarového nástroje, obdobného jako je LabView. Jeho výrobcem je zlínská akciová společnost ***„Moravské přístroje,“*** která vyvinula systém pro tvorbu průmyslových aplikací s názvem **„Control Web“**, který dokonce v některých parametrech americký software svými vlastnostmi převyšuje, a to při ceně, která je mnohem nižší. Proto jsme pro vývoj výukových simulačních aplikací zvolili softwarový nástroj zlínské společnosti.

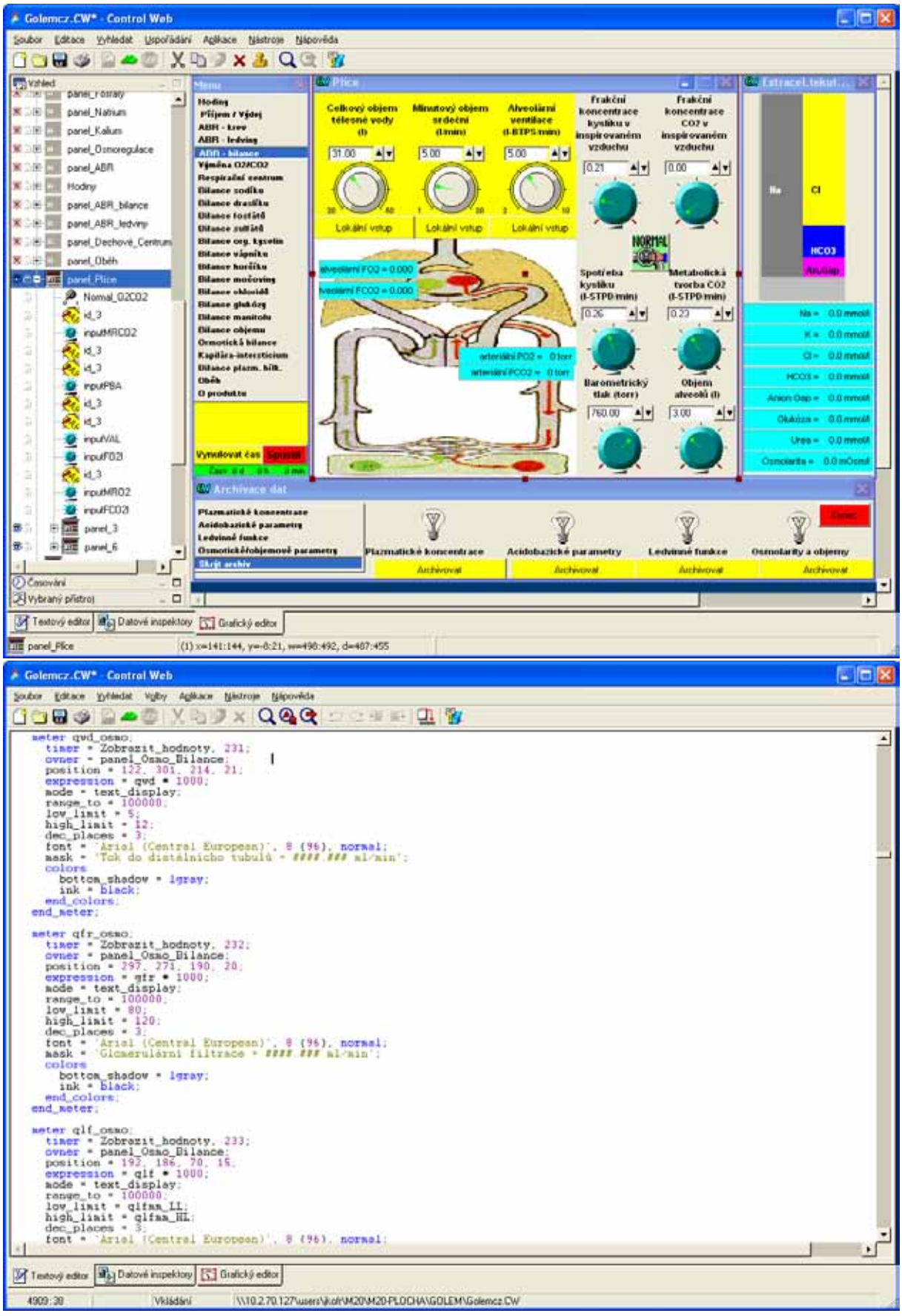
3.6 Součástky pro Golema – virtuální přístroje vývojového nástroje Control Web

„Svensk kvalitet (till Tjeckiska priser)“ – „Švédská kvalita (za české ceny)“, je napsáno v jednom reklamním letáčku skandinávského distributora výkonného softwarového nástroje z Valašska. Švédové jsou velmi domýšliví na kvalitu svých výrobků, a pokud o vývojovém nástroji ***Control Web*** z dílny zlínské společnosti Moravské přístroje veřejně prohlašují, že dosahuje „švédské kvality“, je to velké ocenění skupiny tvůrců, kteří na tomto produktu v potu tváře usilovně pracují již od počátku devadesátých let. Control Web patří k jedněm z mála softwarových produktů českých firem, které si našlo cestu i na zahraniční trhy, včetně Japonska.



Obr. 42 – Komunikace systému Control Web s ovladačem řídící/měřící karty při tvorbě průmyslových aplikací.

Control Web je především určen pro ***vývoj průmyslových vizualizačních a řídících aplikací na platformě WIN32*** – sběr, ukládání a vyhodnocování dat, tvorba rozhraní člověk-stroj aj. (http://www.mii.cz).



Obr. 43 – Vývojové prostředí Control Web umožňuje programování simulátoru jak v texrtovém tak i v grafickém režimu, což podstatně usnadňuje tvorbu uživatelského rozhraní simulátoru.

Objektově-orientovaná komponentová architektura zajišťuje vyvíjeným aplikacím široký rozsah nasazení: od prostých časově nenáročných vizualizací až po řídicí aplikace reálného času, od jednoprocesorových aplikací až po rozsáhlé distribuované síťové aplikace.

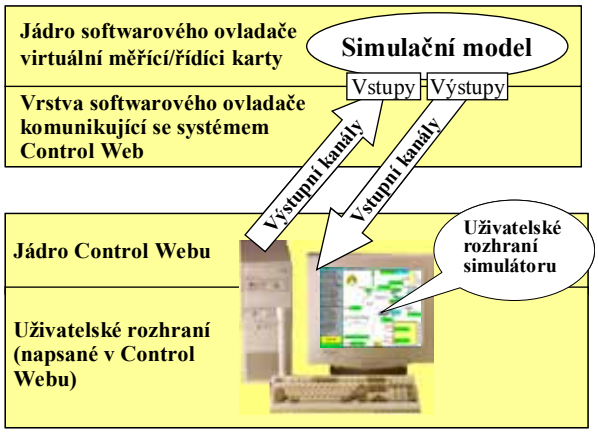
V průmyslových aplikacích jádro systému Control Web komunikuje přes ovladač měřící/řídící karty s průmyslovým zařízením. Na jádro systému je napojena uživatelská aplikace vytvořená ve vývojovém prostředí Control Web pomocí vizualizačních nástrojů. Vyvinutá a odladěná aplikace pro svůj běh pak potřebuje pouze instalovat runtime verzi Control Webu s příslušným ovladačem měřící/řídící karty připojené k technologickému zařízení.

Obr. 44 – Paleta virtuálních přístrojů v systému Control Web. Přístroje je možno pomocí myši z palety přístrojů rozmísťovat do vytvářené aplikace a napojovat na vstupní či výstupní kanály. Tímto způsobem lze jednoduše naprogramovat řídící panel technologického zařízení či interaktivní grafické uživatelské rozhraní simulátoru.

Základními stavebními kameny uživatelské aplikace jsou ***virtuální přístroje*** (komunikující mezi sebou pomocí proměnných a zpráv). Každý virtuální přístroj je komponenta, jejíž vlastnosti jsou pomocí vizualizačních a ovládacích prvků snadno modifikovatelné. Ale nejenom to – každá komponenta systému má k dispozici mocné programátorské nástroje, jako jsou lokální proměnné a libovolně definovatelné procedury, reagující na události. Control Web má pro tyto účely vlastní programovací jazyk syntaxí podobný jazyku Modula 2.

Měřené hodnoty okolního světa jsou virtuálním přístrojům zprostředkovány přes ***vstupní kanály***, ***řídící signály*** do okolí mohou virtuální přístroje posílat pomocí ***výstupních kanálů*** (obr. 42). Důležité je, že systém umožňuje práci v reálném čase – každý vstupně/výstupní kanál je čten v době, kdy jej nějaký virtuální přístroj (nebo skupina virtuálních přístrojů) potřebuje. Real-time časování je přesně monitorováno a řízeno. Virtuální přístroje mohou být časovány v přesně definovaném čase a v přesně definované sekvenci.

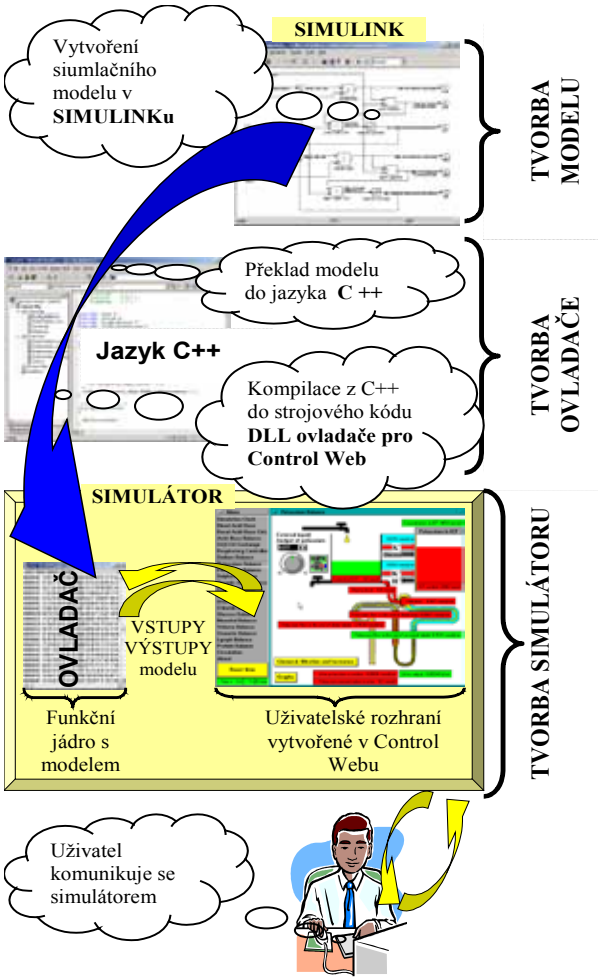
Vývojové prostředí umožňuje ***dvojcestné programování*** v textovém nebo grafickém režimu (obr. 43). Zdrojový tvar aplikace je textový – a textová podoba je jediná, kterou Control Web používá pro ukládání. Control Web umožňuje snadno překlopit zdrojový text aplikace z textové do grafické podoby (překlad), v níž systém poskytuje mocné vizuální prostředky pro snadný a přehledný vývoj aplikace. Obdobně snadno se dá grafická podoba aplikace překlopit do textové (generování).



Obr. 45 – Začlenění simulačního modelu do ovladače „virtuální karty" při tvorbě simulátoru v prostředí Control Web.

Pro vývoj uživatelského rozhraní simulátoru Golem systém Control Web poskytuje velmi výkonné prostředky. Tak např. z palety virtuálních přístrojů (obr. 44) je možno snadno tažením myši vytáhnout potřebný přístroj a umístit ho na příslušný panel a v interaktivním dialogu mu nastavit hodnoty jeho příslušných atributů, nadefinovat jeho lokální proměnné, či individuální procedury (metody objektu) apod.

Abychom mohli využít vývojářské pohodlí systému Control Web, ***bylo nutno napsat speciální ovladač***, který byl schopen komunikovat (přes softwarové kanály) s objekty systému Control Web. Na rozdíl od ovladačů ke skutečným měřícím a řídícím kartám však tento ***ovladač nekomunikuje s hardwarem těchto karet, ale se simulačním modelem***, který je součástí ovladače (Obr 45). Pokud se ovla-



Obr. 46 – Postupná tvorba simulátoru Golem v prostředí Control Web.

dač napíše dobře, je systém Control Web „ošálen“: vstupní kanály (k měřícím přístrojům) považuje za skutečné měřené signály někde v technologickém okolí počítače, zatímco ve skutečnosti to jsou výstupní proměnné simulačního modelu. Výstupní kanály, odcházející od řídících prvků systému Control Web, nenastavují přes příslušný ovladač nějaké aktivní prvky technologie, ale mění vstupy simulačního modelu. Vzájemný vztah mezi simulačním modelem ve virtuálním ovladači a vizualizačním rozhraním systému Control Web je realizován jako ***klient – server technologie***: v určitých časových okamžicích Control Web (klient) žádá virtuální ovladač se simulačním modelem (server) o načtení hodnot ze vstupních kanálů a umožnění změny hodnot výstupních kanálů.

Jádrem simulátoru Golem je tedy ***virtuální ovladač***, který obsahuje vlastní simulační model.

Modelové jádro simulátoru jako ovladač virtuální měřící/řídící karty, bylo zprvu implementováno v jazyce Modula (Kofránek, Velan & Kerekeš, 1997) a později v C++. (Kofránek, Velan & Janicadis, 2000, Kofránek, Velan, Janicadis, Kerekeš, 2001; Kofránek J., Snášelová, Anh Vu, Janicadis & Velan, 2001; Kofránek, Anh Vu, Snášelová, Kerekeš & Velan, 2001).

Pro implementaci ovladače jsme využívali simulační model, o jehož adekvátnosti jsme se (se vším pohodlím systému **MATLAB** a **SIMULINK**) již přesvědčili (obr. 46).

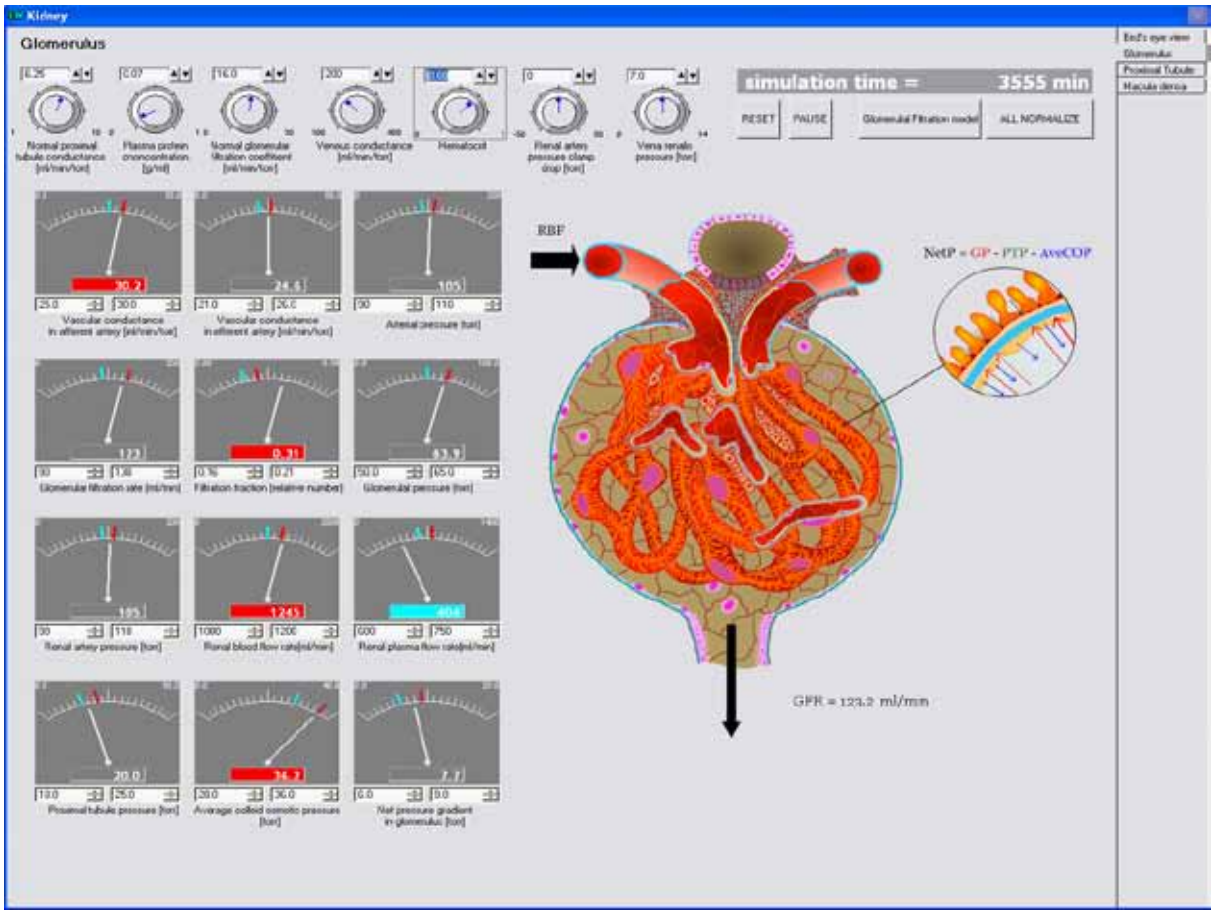
Při změnách matematického modelu, odlaďovaného v prostředí Matlab/Simulink (tj. v prostředí pro vývoj matematických modelů) bylo ale nutné tyto změny vždy promítnout i do simulačního jádra simulátoru implementovaného v C++. Proto jsme postupně hledali cesty jak tuto transformaci usnadnit a nemuset tento ovladač psát v C++ „ručně“ (Kofránek, Andrlík, Kripner & Mašek, 2002a; Kofránek, Andrlík, Kripner & Mašek, 2002b; Kofránek, Andrlík & Kripner, 2003).

Vyvinuli jsme speciálního nástroj, který ***umožní vývoj ovladače automatizovat***. To nám umožnilo ze simulinkového schématu přímo generovat zdrojový text příslušného virtuálního ovladače v C++. Tím bylo možné jednoduše modifikovat ovladač v prostředí Control Web při nejrůznějších úpravách simulačního modelu v prostředí Simulink (Kofránek, Andrlík, Kripner & Mašek, 2002a).

3.7 Modelem řízené interaktivní animace

Vnější vzhled aplikace v Control Webu byl dlouho omezen nabídkou dodávaných virtuálních přístrojů. Paleta těchto přístrojů je dostatečně velká, avšak je určena především pro technické aplikace. Z toho vyplýval poněkud technický vzhled vlastního simulátoru.

Z pedagogického hlediska je ale vhodnější, když výukový simulátor připomíná spíše obrázky a schémata z lékařských učebnic, než velín atomové elektrárny. Pro snadnější realizaci tohoto záměru jsme na přelomu tisíciletí navázali úzkou spolupráci s výtvarnou školou Václava Hollara a postupně naučili výtvarníky vytvářet interaktivní animace v prostředí Flash. Naším cílem bylo ve výukovém simulátoru využít animace, které by byly řízené podle momentálních hodnot výstupů modelu.



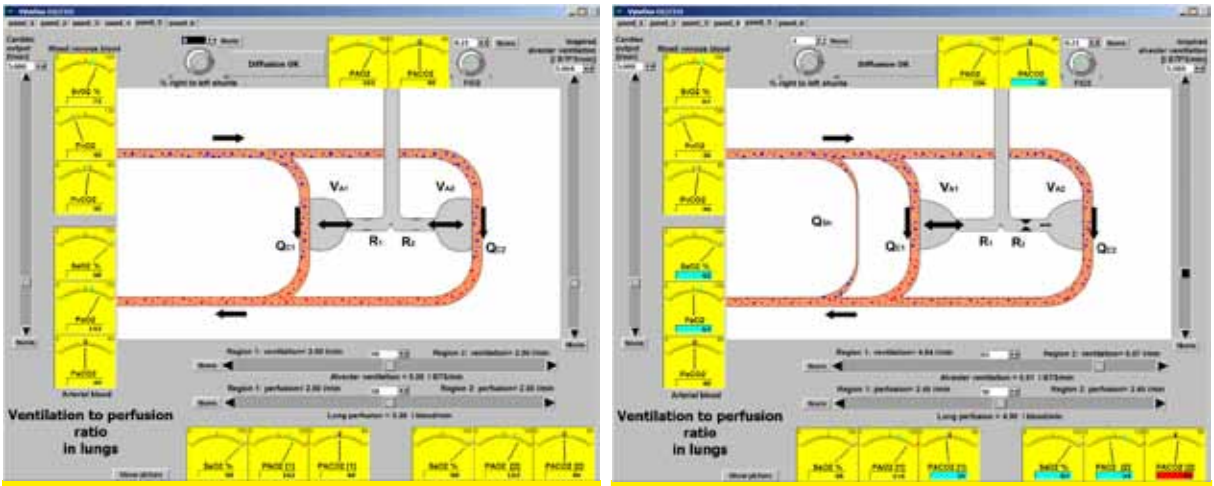
Obr. 47 – Vizuální rozhraní ve výukovém simulátoru funkce ledvin vytvořeného v prostředí ControlWeb. Výstupy modelu jsou zobrazovány na ručkových měřících přístrojích a zároveň ovlivňují i tvar animovaného obrázku ledvinného glomerulu (průměry cév, tloušťku šipek a číselnou hodnotu aj.), vytvořeného pomocí programu Adobe Flash.

Animace by tak byly zavěšeny jako „loutky“ na výstupních hodnotách modelu a obrázková schémata tvořící výstupní rozhraní simulátoru by vyjadřovala stav simulované reality.

Začali jsme proto využívat jeden z virtuálních přístrojů z nabídky Control Webu: ***kontejner pro komponenty ActiveX***, který představuje most mezi systémem Control Web a vlastnostmi a metodami (OLE Automation) ActiveX komponent. To znamená, že do aplikace jsme mohli zabudovat ActiveX komponenty a programově je ovládat – nastavovat jejich vlastnosti a volat metody z procedur jakýchkoliv přístrojů. To nám umožnilo pomocí systému Macromedia Flash (nyní Adobe Flash) vytvořit interaktivní výukové animace a ty pak uložit jako ActiveX komponenty do kontejneru. Animace pak mohly být řízeny na základě výstupů implementovaného simulačního modelu – např. přívodní a odvodní arterioly v ledvinném glomerulu se mohly roztahovat nebo komprimovat (viz obr. 47), plicní sklípek mohl hlouběji či mělčeji „dýchat“, barva proudících erytrocytů v cévách se měnila od modré po červenou podle hodnoty saturace hemoglobinu kyslíkem apod. (viz obr. 48).

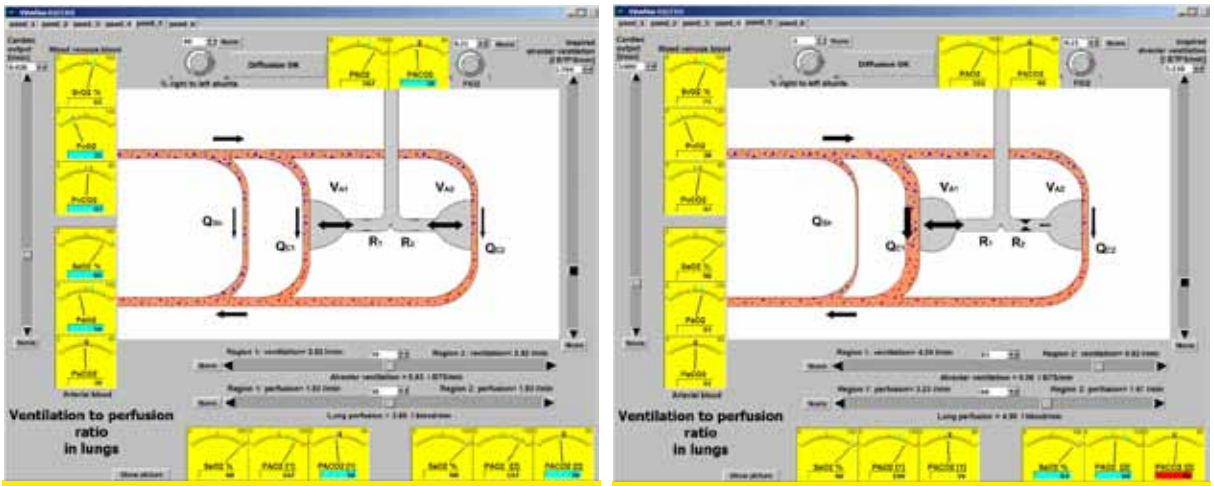
Interaktivní animované obrázky, propojené se vstupy a výstupy simulačního modelu na pozadí, více přiblížily uživatelské rozhraní simulátoru schematickým ilustracím z lékařských učebnic. Na rozdíl od statických knižních ilustrací, jsou obrázky propojené se simulačním modelem interaktivní a umožňují využívat obrázek pro nejrůznější výukové simulační hry.

Nicméně se ukázalo, že prostředí Control Web, v němž jsme implementovali simulátor, není pro tyto nové požadavky vzhledu uživatelského rozhraní, nejvhodnější. Prostředí Control Web sice umožňuje rychle vytvořit a modifikovat uživatelské prostředí simulátoru, avšak jeho vzhled je, i přes využití animovaných obrázků propojených přes ActiveX se simulačním jádrem, příliš svázán s komponentami původního vývojového nástroje pro průmyslové aplikace a výukový simulátor má příliš „technický“ vzhled (jako např. obrázek glomerulu obklopený množstvím ručkových měřících a přístrojů a ovládacích prvků na obr. 47).



Normální výchozí stav

Nerovnoměrný poměr mezi ventilací a perfúzí, kompenzace celkovým zvýšením ventilace



Zvýšení procenta pravo-levých plicních zkratů

Kompenzace ventilačně-perfúzní nerovnoměrnosti celkovým zvýšením ventilace a omezením perfúze v hypoventilovaných alveolech

Obr. 48 – Ukázka různých stavů vizuálního uživatelského rozhraní v simulační hře ve výukovém programu poruch respirace. Uživatelské rozhraní, vytvořené v prostředí Control Web s využitím flashové animace, schematicky zobrazuje různé stupně ventilace a perfúze dvou částí plic. Student si „hraje" s pohyblivým obrázkem a model na pozadí propočítává příslušné fyziologické parametry, zobrazované jako hodnoty virtuálních měřících přístrojů (v Control Webu). Zároveň se na schematickém obrázku tvořeném flashovou animací mění „hloubka dýchání" a „velikost prokrvení" plic, saturace hemoglobinu kyslíkem ovlivňuje barvu (od modré po červenou) pohybujících se erytrocytů v cévách apod.

Kromě toho při hojnějším využití mnoha flashových komponent propojených přes ActiveX se simulačním jádrem, mělo jejich ovládání a vykreslování pomocí neúměrně velkou režii a zpomalovalo výpočet. Tím se pak ale ztrácela výhoda rychlého vykreslování grafiky při využívání standardních virtuálních přístrojů z nabídky vývojového prostředí Control Web.

3.8 Golem na síti

Jak již název „Control Web" napovídá, systém ***umožňuje tvorbu skutečně distribuovaných řešení***. Pokud je na všech komunikujících počítačích nainstalován Control Web runtime, pak je možné zpřístupnit libovolný datový element všem spojeným aplikacím po libovolne TCP/IP síti včetně internetu. Po síti mohou být aktivovány virtuální přístroje, po síti mohou být volány i metody dynamického rozhraní vir-

tuálních přístrojů. Síťová komunikace může být samozřejmě precizně časována a řízena k dosažení optimálního výkonu. Kromě toho Control Web obsahuje ***plnohodnotný HTTP server*** dynamicky tvořící stránky podle stavu technologie (pracující i na starých obyčejných Windows 95). Navíc dokáže prostřednictvím HTTP a HTML technologii i řídit.

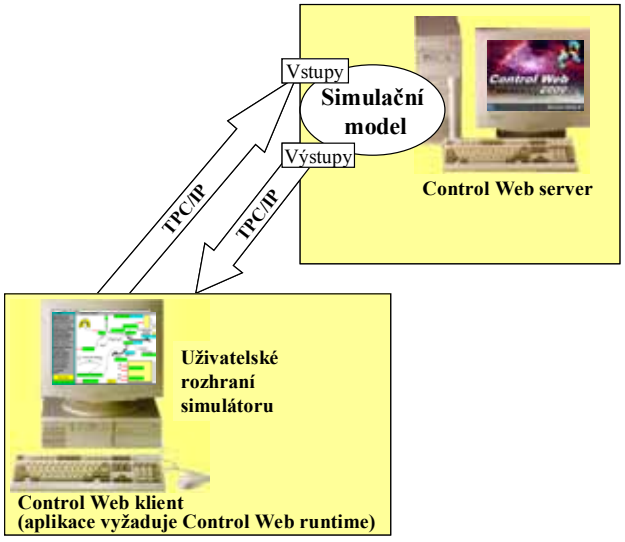
V zásadě existují ***tři způsoby tvorby distribuovaných aplikací v prostředí Control Web***. Všechny tři způsoby jsou plně postaveny na internetových technologiích, protokolech TCP/IP a HTTP, formátu HTML, standardu Java apod.

1. Aplikace systému ***Control Web*** dokáží mezi sebou komunikovat po libovolné TCP/IP síti, tedy i po internetu. Samozřejmě aplikace komunikující po rychlé lokální síti mohou přenášet častěji větší bloky dat než aplikace komunikující přes půl zeměkoule. Takto distribuovaná aplikace přináší nejlepší uživatelskou zkušenost, neboť lze využít všech vlastností systému ***Control Web*** i operačního systému Windows. Na druhé straně aplikace pracuje jen pokud je na všech počítačích nainstalován alespoň ***Control Web Runtime***, který pracuje pouze pod systémem Windows.

2. Při použití čistého HTML lze pomocí zabudovaného HTTP serveru prezentovat data libovolnému WWW klientovi pracujícímu na velkém množství počítačů, včetně přenosných počítačů do dlaně. Za tuto obecnost se ale platí nejchudšími vyjadřovacími prostředky.

3. Data lze prezentovat pomocí Java appletů zabudovaných do HTML stránek. Java applety zdaleka neposkytují takové možnosti jako samotný ***Control Web***, ale na druhé straně pracují ve všech WWW prohlížečích podporujících JVM, tedy i mimo systémy Windows. Pomocí Java appletů je možno vyjádřit všechny základní přístroje Control Webu (tyto aplety jsou součástí dodávky Control Webového vývojového prostředí). Obdobně jako Java lze dnes pro komunikaci klientského počítače s Control Web serverem využít i aplikace využívající Adobe Flash a nainstalovaný Flash Player v prohlížeči. To umožňuje vytvářet distribuované aplikace, kdy na komunikaci s aplikací vytvořenou v Control Webu stačí obyčejný WWW prohlížeč (není nutný Control Web klient).

Můžeme tedy vytvářet distribuovanou aplikaci, kdy vlastní numericky náročné výpočty simulátoru probíhají na serveru (Control Web serveru) a vizualizace probíhá na klientech (obr. 49). Na každém z klientů je nutno nainstalovat ControlWeb runtime. Takové uspořádání má ale význam především tehdy, pokud na simulačnímu serveru probíhá společný simulační výpočet určený pro více klientů, které ve skutečnosti slouží jen jako vizualizační a řídící terminály, například v počítačové učebně při semináři, kdy všichni studenti pod vedením pedagoga mohou sledovat stav modelovaného pacienta a zároveň mají možnost ovlivňovat jeho vstupy (Kofránek, Vrána, Velan & Janicadis, 2001).



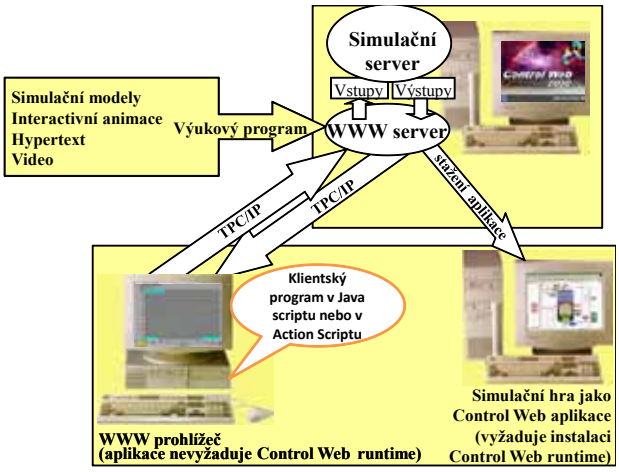
Obr. 49 – Možnost vytváření distribuované výukové simulační aplikace v intranetových a internetových sítích pomocí Control Web serveru.

Pokud by ale více klientů chtělo spouštět vlastní samostatnou simulaci, pak je výhodnější aby jak vizualizace, tak i vlastní simulační jádro, běželo na klientském počítači jako samostatná aplikace a webový server by se stal jen místem, odkud by se simulační aplikace dala stahovat. Pro její spouštění na klientovi je ovšem vyžadováno nainstalování programu Control Web runtime. Pro každý takový program je ale nutno mít platnou licenci a program aktivovat prostřednictvím licenčního serveru. Pro účely výuky nám firma Moravské přístroje na zakázku vytvořila speciální volně šířitelný runtime (pro verzi Control Web 2000), který bylo možné distribuovat přes internet.

Simulátor Golem vyžadoval instalaci (byl dis-

tribuován na CD ROM). Naším cílem bylo umožnit šíření simulátoru prostřednictvím internetu (zejména pro usnadnění aktualizace simulátoru při jeho dalším vývoji) a zejména umožnit snadné propojení simulací s výukovými vysvětlovacími texty a interaktivními animacemi.

Řešili jsme proto otázku, zda zvolit Control Web jako hlavní platformu pro internetové aplikace. Zvažovali jsme, zda okolo Control Webu, pracujícím jako www server, by bylo také možné vystavit výukové interaktivní aplikace, využívající simulační hry (obr. 50). Vlastní simulace by běžely buď na serveru a na klientech v internetovém prohlížeči by byly vizualizovány (s využitím Java Scriptů s instalovaným javovým pluginem v prohlížeči, nebo s využitím Action Scriptu pro zobrazení ve Flashovém prohlížeči). V tomto případě by ale mohly nastat problémy s tím, že každá vizualizační klientská aplikace by vyžadovala spuštění své vlastní simulační aplikace na internetovém Control Web serveru, což by při náročnějších výpočtech mohlo vést k přetížení jeho kapacity. Navíc bychom tím přišli o možnost využití všech výhod využití virtuálních přístrojů, které nabízí vývojové prostředí Control Web. Proto jsme nakonec hledali jiná řešení.



Obr. 50 – Control Webová aplikace může poskytnout i plnohodnotný HTTP server. Výukové aplikace tak mohou komunikovat se vzdáleným výukovým simulačním serverem vytvořeným v systému Control Web pomocí obyčejného WWW prohlížeče bez nutnosti mít nainstalován Control Web runtime na klientské straně. Výukový simulátor vytvořený v prostředí Control Web může také být stažen ze serveru a na klientském počítači pracovat jako samostatná aplikace.

3.9 Golem na rozcestí

Výhodou při vývoji simulátoru Golem bylo oddělení prostředí pro vývoj simulačního modelu a vývojového prostředí pro vytvoření vlastního simulátoru. Při vývoji a modifikacích matematického simulátoru jsme mohli využívat všech výhod nástroje určeného pro tvorbu a ladění modelů (Matlab/Simulink) a vývoj a úpravy simulátoru nám urychlovalo vývojové prostředí Control Web.

Vytvořili jsme si nástroj, který umožňoval generovat zdrojový text simulačního jádra pro Control Webovou aplikaci přímo ze Simulinku, což usnadnilo průběžnou aktualizaci simulátoru podle nových verzí modelu (viz obr. 46).

Zároveň to ale přinášelo potíže.

Stálým problémem bylo udržovat soulad poloautomatického generátoru simulačního ovladače mezi novými verzemi Simulinku a Control Webu. Vývoj nových verzí Simulinku je rychlý. Jeho výrobce, firma Mathworks, stejně tak jako i Moravské přístroje pravidelně inovují svůj nástroj. Každá nová verze Simulinku nebo Control Webu nezřídka znamenala nutnost přepracovat vývojový nástroj, který generuje ze simulinkového modelu ovladač virtuální řídící/měřící karty v C++ se simulačním modelem.

Ne zcela optimální se ukázala i realizace distribuce simulátoru prostřednictvím webových aplikací a jeho provázání s interaktivními multimédii a výkladovými výukovými aplikacemi spustitelnými přes internetový prohlížeč.

Rozšíření vysokorychlostního internetu a rozvoj nástrojů pro tvorbu interaktivní animované grafiky, přináší nové možnosti pro vytváření výukových aplikací dosažitelných přes internetový prohlížeč. Uživatelský vzhled výukových aplikací začíná stále více hrát svoji úlohu. Od roku 2003 jsme proto navázali úzkou spolupráci se Střední uměleckou školou Václava Hollara. V roce 2007 jsme společně stáli u zrodu Vyšší odborné školy Václava Hollara se zaměřením na interaktivní grafiku (a dodnes na ní vyučujeme předmět „ovládání interaktivity“ a vedeme studentské praxe). Získali jsme proto možnosti pro naše výukové aplikace vytvářet uživatelské rozhraní s interaktivními animacemi spustitelnými v internetovém prohlížeči.

Hojné využívání možností vkládat flashové animace do aplikací vytvářených v prostředí Control Web mělo svou stinnou stránku v poměrně velké časové režii, která zpomalovala chod celé aplikace. Tím se ztrácela výhoda rychlého vykreslování grafických prvků při využívání standardní Control Webové knihovny virtuálních přístrojů.

Tyto potíže vedly k tomu, že jsme další vývoj simulátoru Golem v prostředí Control Web v roce 2005 přerušili a začali se více věnovat tvorbě webově instalovatelných simulačních her, propojených s výkladovou částí. Simulační model, který byl podkladem simulátoru Golem z roku 2005 (Kofránek, Andrlík & Kripner, 2005) je součástí Simulinkové knihovny „Physiolibrary" volně dostupné z adresy http://physiome.cz/simchips.

3.10 Od Golema k internetové učebnici - výklad pomocí simulačních her

K přehodnocení strategie výstavby složitého simulátoru v prostředí Control Web a naši následnou orientaci na jednodušší „webové" výukové simulátory nás, kromě nutnosti změnit implementační platformu, vedl i další důvod. Naše zkušenosti i zkušenosti jiných pracovišť s nasazením komplexních modelů do výuky totiž ukazují, že velké a složité modely mají z didaktického hlediska značnou nevýhodu ve složitém ovládání (Lane, 2001; de Freitas, 2006; Kofránek, Privitzer, Matoušek, Vacek & Tribula, 2009).

Velké množství vstupních proměnných i široká paleta možností sledování výstupních proměnných, vyžadují od uživatele důkladnější porozumění vlastní struktury simulačního modelu, i znalost toho, jaké procesy je zapotřebí při simulacích určitých patologických stavů sledovat. V opačném případě se složitý sofistikovaný model uživateli jeví jen jako „složitá a málo srozumitelná technická hračka" (obdobně, jako když ho posadíte před složitý simulátor dopravního letadla bez předchozího teoretického kurzu).

Nebyl to jen problém simulátoru Golem. Např. určitou brzdou většího rozšíření komplexních simulátorů QCP (obr. 11) a QHP (obr. 15), je jejich složité ovládání. Rozvětvené uživatelskému rozhraní, umožňující najednou sledovat hodnoty stovek proměnných se nakonec paradoxně ukazuje jako omezující prvek pro jejich nasazení v běžné výuce na lékařských fakultách. Přesto, že tyto simulátory jsou volně stažitelné z internetu, ve výuce se zatím příliš nerozšířily.

Výukové modely (a zřejmě nejen komplexní modely se stovkami proměnných) pro efektivní využití ve výuce proto sami o sobě nestačí. Musí být provázeny výkladem jejich využití – nejlépe pomocí interaktivních výukových aplikací na internetu.

Teprve ***spojení výkladu a se simulační hrou*** dává možnost využít všech výhod virtuální reality pro vysvětlení složitých patofyziologických procesů.

Pedagogická praxe ukázala, že simulační hra s jednoduchými agregovanými modely (s možností sledování pouze několika proměnných) je někdy vhodnějším nástrojem pro vysvětlení složitých procesů než rozsáhlý výukový model. Při výkladu je vhodné postupovat od jednoduššího ke složitějšímu, tj. nejprve s využitím jednoduchých modelů vysvětlit základní principy a teprve potom se věnovat složitějším detailům a využívat simulační hry se složitějšími modely.

Naše zkušenosti s vývojem a využitím simulátoru Golem a tvorbou výukových aplikací využívajících simulační hry nás také vedli k tomu, abychom věnovali více pozornosti technologiím a metodikám výstavby výukových programů a simulátorů, i požadavkům jejich uživatelů – tj. v našem případě pedagogům a studentům medicíny.

Je zřejmé, že sebelepší simulátor bez dostupného výkladového textu a jasného scénáře jeho využití ve výuce, je jen poloviční řešení. Na druhé straně je zřejmé, že i vynikající interaktivní multimediální výukové programy s mnoha ozvučenými animacemi, ale bez simulačních her s modely na pozadí, jako je např. Interactive Physiology (Branstrom, a další, 2008), využívají možnosti počítače pro výuku jen napůl.

Z toho vyplývá jasný požadavek na skloubení interaktivní multimediálního výkladového textu se simulátorem. Ukazuje se, že jakousi „kostrou" výukové aplikace, o níž lze opřít interaktivní multimediální animace propojené se simulačními hrami, je scénář výukového programu. Nejlépe hned na počátku musí být jasné, jaké simulátory budeme potřebovat a jakým způsobem je chceme ve výukovém programu využívat. Z toho vyplynou i požadavky na návrh struktury simulátoru a modelu na jeho pozadí.

Je zřejmé, že vzhled a přívětivé uživatelské rozhraní výukové aplikace je pro pedagogický úspěch

velmi podstatné. Použité animované ilustrace by měly mít profesionální vzhled, a měl by je proto vytvářet profesionální výtvarník.

Požadavkem doby je dostupnost výukových aplikací „na kliknutí“ přes internet – nejlépe přímo v internetovém prohlížeči.

Pro skloubení možností interaktivních multimédií a simulačních modelů pro lékařskou výuku jsme proto koncipovali projekt internetového počítačového ***Atlasu fyziologie a patofyziologie*** jako multimediální výukovou pomůcku, která názornou cestou prostřednictvím internetu s využitím simulačních modelů by měla pomoci vysvětlit funkci jednotlivých fyziologických subsystémů, příčiny a projevy jejich poruch – http://physiome.cz/atlas.

V atlasu se snažíme kombinovat výkladové kapitoly, doprovázené interaktivními animacemi (synchronizované se zvukovým doprovodem), se simulačními hrami s modely jednotlivých fyziologických subsystémů. Vše je volně dostupné z internetu.

Jestliže „kostrou“ výukové aplikace je kvalitní scénář, a jejími „svaly“, kterými se aplikace může pyšnit navenek, jsou multimediální animace, pak jakýmsi „mozkem“ výukového programu je verifikovaný simulační model, na požadované úrovni přesnosti odrážející chování modelované reality. Simulační model představuje formalizovaný popis reality implementovaný na počítači. Jeho tvorba je proto zpravidla více výzkumný než čistě vývojový problém.

Během tvorby simulátoru Golem se využitelné technologie doslova měnily pod rukama. Objevovaly se nové technologie usnadňující tvorbu multimediálních animací, propojitelných se simulačním modelem na pozadí. Zároveň se objevily i simulační nástroje, usnadňující vytváření složitých hierarchicky organizovaných modelů. Tyto nástroje jsme se rozhodli využít pro vytvoření nové složitější verze simulátoru fyziologických funkcí s pracovním názvem ***eGolem.***

3.11 Golemem to nekončí - zkušenosti a cíle další práce

Naše zkušenosti s výstavbou a využitím simulátoru Golem můžeme shrnout do následujících bodů:

- Z didaktického hlediska se ukázalo velmi výhodné, že jsme do simulátoru Golem zavedli **možnost rozpojovat některé regulační smyčky** a umožnit studentům v simulační hře sledovat reakce zvoleného fyziologického subsystému na změny vstupních veličin (které jsou ovšem v reálném organismu samy regulovány). Podle našich zkušeností tento přístup vede k lepšímu pochopení významu jednotlivých regulačních smyček a porozumění jejich úlohy v patogeneze nejrůznějších onemocnění a pochopení patofyziologických principů příslušných léčebných zásahů (Kofránek, Anh Vu, Snášelová, Kerekeš, & Velan, 2001).
- Na druhé straně ale naše zkušenosti s uplatněním simulátoru Golem ve výuce ukázaly, že velké a složité simulátory mají z didaktického hlediska značnou nevýhodu v poměrně komplikovaném ovládání. Obdobnou zkušenost s didaktickým využitím složitých modelů ve výuce potvrzují i zahraniční autoři (Lane, 2001; de Freitas, 2006). Velké množství vstupních a výstupních proměnných totiž vyžaduje od uživatele důkladnější porozumění struktury modelu i znalost toho, jak složitý model ovládat a jaké proměnné je vhodné při simulaci nejrůznějších patologických stavů sledovat (v opačném případě se složitý model uživateli jeví jako složitá a málo srozumitelná technická hračka). Výukové modely pro jejich efektivní didaktické využití ve výuce proto musí být provázeny s interaktivním výkladem – teprve **spojení výkladu se simulační hrou** dává možnost využití všech výhod virtuální reality pro vysvětlení složitých regulačních procesů ve zdravém i nemocném organismu.
- Z didaktického hlediska je vhodné při výkladu postupovat **od jednoduchého ke složitějšímu**, a proto je vhodné při výkladu využívat nejprve jednodušší agregované modely (s několika proměnnými), s jejich pomocí vysvětlit základní principy a poté model (a popisovanou fyziologickou realitu) postupně zesložiťovat. I jednoduchý interaktivní model může být dobrým pomocníkem pro vysvětlení patogenetických řetězců rozvoje nejrůznějších patologických stavů.
- Z hlediska metodologie tvorby simulátoru se ukázalo jako výhodné používat **odlišné ná-**

stroje pro modelování (v případě simulátoru Golem – vývojové prostředí Matlab/Simulink) a **jiné nástroje pro tvorbu simulátoru** (Control Web).

- Výhodnou se jevila i **výstavba modelů ve formě hierarchicky uspořádaných kompone**nt (jakýchsi „simulačních čipů“), které zpřehledňují strukturu modelu a usnadňují mezioborovou spolupráci.
- Ukázalo se také, že ideálním prostředím pro distribuci i vlastní provozování výukových simulátorů je prostředí **internetu**.
- Jako didakticky účinné se ukázalo **využití interaktivních animací, které jsou řízené výstupy modelu**. Pro tyto požadavky však dosud používaná technologie výstavby simulátoru prostřednictvím vývojového prostředí Control Web se nejevila zcela optimální.

Během postupné tvorby nových verzí výukového simulátoru Golem přicházely nové technologie, které slibovaly nové možnosti pro vytváření multimediálních výukových simulačních aplikací dosažitelných přes internetový prohlížeč. Objevily se i nové technologie zefektivňující výstavbu simulačního jádra výukových simulátorů.

Na základě dosavadních zkušeností s výstavbou a využitím simulátoru Golem byly proto stanoveny ***cíle další práce***:

- při ***tvorbě simulačních modelů*** využít ***nové technologie***, zefektivňující tvorbu, testování a ladění složitých hierarchicky uspořádaných modelů;
- zavézt ***novou technologii tvorby výukových simulátorů*** propojujících výklad se simulační hrou;
- vytvořit nástroje umožňující ***propojení interaktivních animací se simulačním modelem*** na pozadí;
- vytvořit ***nástroje usnadňující týmovou multidisciplinární spolupráci*** při vytváření výukových simulačních aplikací (pedagogů, tvůrců simulačních modelů, výtvarníků a programátorů);
- zajistit ***dostupnost výukových aplikací přes internetový prohlížeč***;
- jako konkrétní výsledek vytvořit internetovou výukovou aplikaci ***„Atlas fyziologie a patofyziologie“*** kombinující výklad a simulační hry;

Využít nové technologie tvorby simulačních modelů pro vytváření ***nové složitější verze komplexního simulátoru fyziologických funkcí*** (pracovní název ***„eGolem“***).

4 Od „umění k průmyslu“ - dosažené výsledky v technologii tvorby a využití výukových simulátorů

Navzdory tomu, že se využití počítačů ve výuce stalo tématem řady konferencí, odborných i popularizačních článků, přesto, že hardwarové možnosti i softwarové nástroje dnes umožňují vytvářet náročná interaktivní multimedia, k výraznému rozšíření multimediálních výukových programů ve výuce medicíny zatím nedošlo. Příčin je několik.

- Za prvé, ukazuje se, že tvorba výukových programů je podstatně náročnější na čas, lidské i materiální zdroje, než je obvykle plánováno.
- Za druhé – tvorba kvalitních výukových programů vyžaduje týmovou multidisciplinární spolupráci zkušených pedagogů, výtvarníků i programátorů. Nároky stoupají, pokud na pozadí výukového programu má běžet simulační program, umožňující interaktivní simulační hry – ve vývojovém týmu pak musí být i odborníci, kteří jsou schopni navrhnout, formalizovat a odladit příslušné modely.
- Konečně, pro kreativní propojení různých profesí, podílejících se na tvorbě výukové multimediální aplikace, musí být k dispozici vhodně zvolené vývojové nástroje (jejichž cena není malá a jejichž ovládnutí vyžaduje určité úsilí a čas), vyřešen způsob jejich propojení (což nezřídka znamená i vytvoření specializovaných propojovacích softwarových prostředků) a v interdisciplinárním týmu musí být vytvořen, zaveden a dodržován dohodnutý vývojový cyklus.

Zdá se, že pomalu končí doba, kdy vytváření výukových programů bylo otázkou entuziasmu a píle skupin nadšenců. Tvorba moderních výukových aplikací je náročný a komplikovaný projekt, vyžadující ***týmovou spolupráci*** řady profesí – od zkušených učitelů, jejichž scénář je základem kvalitní výukové aplikace, přes systémové analytiky, kteří jsou ve spolupráci s profesionály daného oboru odpovědni za vytvoření simulačních modelů pro výukové simulační hry, výtvarníky, kteří vytvářejí vnější vizuální podobu, až po programátory, kteří celou aplikaci „sešijí“ do výsledné podoby.

Aby mezioborová spolupráce byla účinná, je zapotřebí pro každou etapu vývoje mít k dispozici řadu ***specifických vývojových nástrojů*** a ***metodologi***í, které práci jednotlivých členů týmu usnadní a pomohou jim překonat mezioborové bariéry. Propojením různých profesí a technologií se ***tvorba výukového softwaru*** stává efektivnější, pozvolna přestává být výsledkem kreativity a pracovitosti jedinců a stále více získává ***rysy inženýrské práce*** (Kofránek, Andrlík, Kripner & Stodulka, 2005).

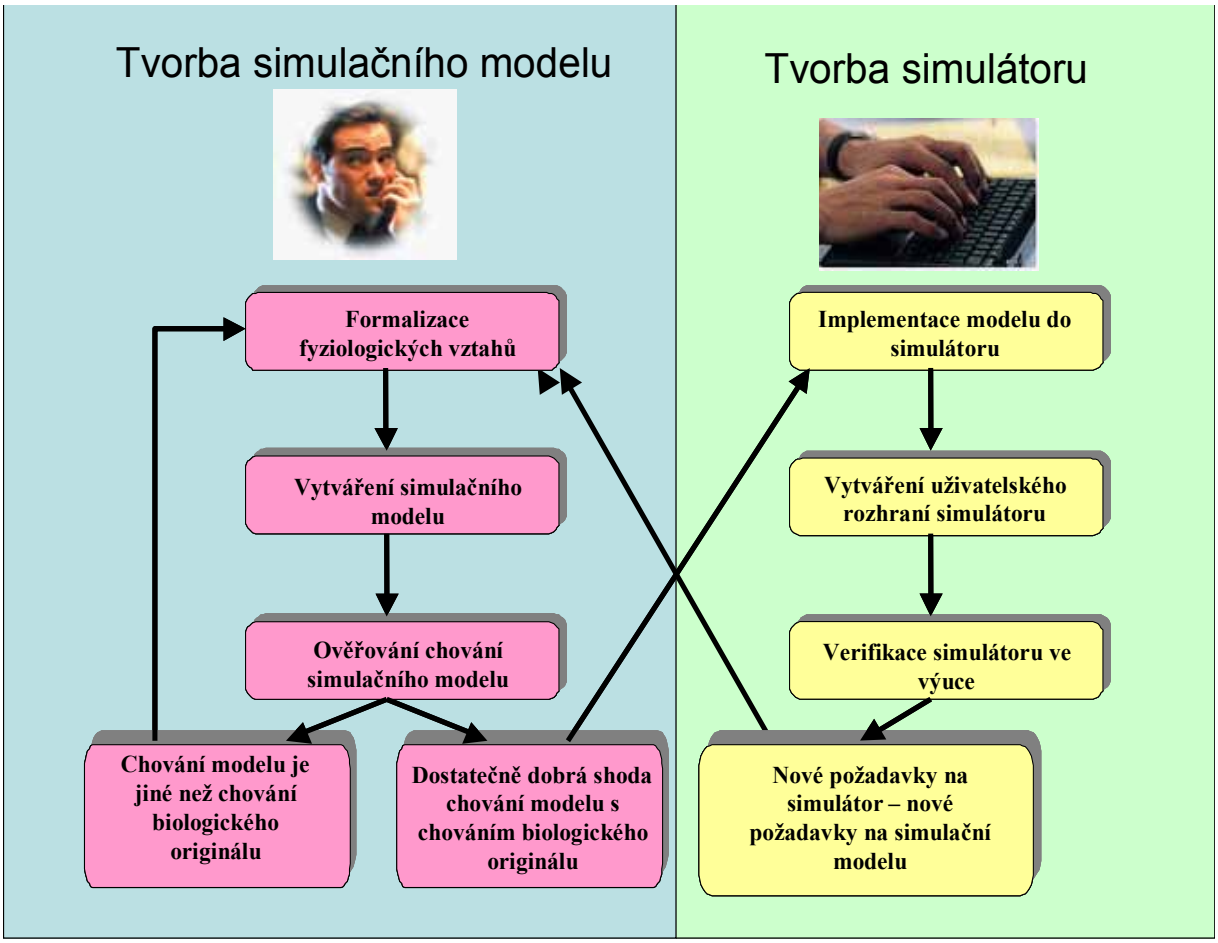
4.1 Dva typy problémů při tvorbě výukových simulátorů

Při vytváření simulátorů a výukových simulačních her je nutno řešit dva typy problémů:

1. ***Tvorba simulačního modelu*** – vlastní teoretická výzkumná práce, jejíž podstatou je formalizované vyjádření reality vyjádřené matematickým modelem. Výsledkem by měl být verifikovaný simulační model, který na zvolené úrovni přesnosti dostatečně věrně odráží chování modelované reality.
2. ***Tvorba vlastního multimediálního simulátoru***, resp. tvorba výukového programu využívajícího simulační hry – je praktická aplikace teoretických výsledků, která navazuje na výsledky řešení výzkumu. Podkladem simulátoru jsou vytvořené (a verifikované) matematické modely. Zde jde o náročnou vývojovou práci, vyžadující skloubit nápady a zkušenosti pedagogů, vytvářejících scénář výukového programu, kreativitu výtvarníků, vytvářejících interaktivní multimediální komponenty a úsilí programátorů, kteří „sešijí“ výsledné dílo do konečné podoby.

Každý z těchto problémů má své zvláštnosti, a vyžaduje proto použít zcela odlišné vývojové nástroje.

Zatímco vytvoření vlastního simulátoru je spíše vývojářskou a programátorskou prací, tvorba simulačního modelu není vývojářský, ale (poměrně náročný) výzkumný problém související s hledáním adekvátního formalizovaného popisu modelované reality (viz obr. 51). Na základě formalizovaného popisu je vytvořen simulační model, který (řešením příslušných rovnic matematického modelu) na počítači



Obr. 51 – Dva typy problémů při tvorbě výukových simulátorů.

simuluje chování modelované reality. Chování modelu je porovnáváno s chováním reálného systému. Rozdíly v chování vedou ke korekcím formalizovaného popisu (např. stanovením nových hodnot některých koeficientů matematického modelu nebo přímo i ke změnám rovnic modelu) do té doby, dokud chování modelu v daných mezích přesnosti se neshoduje s chováním modelované reality. Hovoříme o tzv. verifikaci modelu.

V minulosti se simulační modely vytvářely přímo ve stejném vývojovém prostředí jako i vlastní simulátor (např. v programovacím jazyku Fortran, C++ či Java). V současné době se pro tvorbu a testování simulačních modelů stále častěji využívají specializované vývojové nástroje – např. Maltab/Simulink od firmy Mathworks aj. Zvlášť perspektivní jsou tzv. akauzální simulační prostředí (využívající např. speciální jazyk Modelica), o nichž pojednávají další kapitoly.

Vývojové nástroje pro tvorbu simulačních modelů nejsou příliš vhodné pro vytváření vlastního výukového simulátoru. I když v prostředí těchto nástrojů je možné naprogramovat poměrně příjemné uživatelské rozhraní k ovládání vytvořeného modelu, je toto rozhraní (zvláště pro uplatnění ve výuce medicíny) až příliš komplikované a navíc často vyžaduje zakoupení dalších (poměrně drahých) licencí. Student medicíny a lékař vyžaduje uživatelské rozhraní simulátoru připomínající spíše obrázky a schémata obdobná, s jakými se setkává v lékařských knižních publikacích, než jen suchý výpis množství grafů a hodnot jednotlivých proměnných modelu.

Proto je nutné výukový simulátor včetně jeho multimediálního uživatelského rozhraní naprogramovat zvlášť. Možnosti uživatelského ovládání simulátoru jsou pak pro cílovou skupinu uživatelů podstatně přirozenější.

Tvorba výukových simulátorů a výukových programů využívajících simulační hry je především programátorská vývojové práce, která z jedné strany navazuje na scénář výukového programu, vytvořeného zkušeným pedagogem a z druhé strany na výsledky řešení výzkumu, tj. na vytvořené (a verifikova-

né) matematické modely.

Nezastupitelnou komponentou výukového simulátoru je i část programu, která realizuje simulační model. Známe-li strukturu simulačního modelu (který jsme si vytvořili v některém z vývojových nástrojů pro tvorbu simulačních modelů), pak zbývá přetvořit strukturu modelu do podoby počítačového programu ve zvoleném programovacím jazyce (např. v Javě, C++, C# apod.). Simulační jádro je možné naprogramovat i „ručně“, ale u složitějších modelů se vyplatí, pokud máme k dispozici některý nástroj, který tuto činnost zautomatizuje.

Pro vytvoření uživatelského rozhraní simulátoru je třeba nakreslit interaktivní multimediální komponenty. Tyto komponenty je pak nutno propojit se simulačním modelem na pozadí simulátoru. V moderních výukových aplikacích jsou často tyto multimediální komponenty tvořeny interaktivními animovanými obrázky. Při tvorbě profesionálních výukových aplikací proto musí s programátory ještě spolupracovat výtvarníci, kteří animované obrázky vytvářejí.

Při vytváření vlastního simulátoru se obvykle využívají nejrůznější standardní vývojová prostředí pro tvorbu softwarových a webových komponent (např. Visual Studio .NET, prostředí pro vývoj v jazyce Java jako třeba NetBeans aj.). Pro návrh a programování interaktivní grafiky je možno využít další specializovaná prostředí (jako např. Adobe Flash a příslušný jazyk ActionScript aj.). Výukové simulátory jsou také vyvíjeny pomocí nástrojů pro vizualizaci průmyslových aplikací, např. my jsme dlouho pro tento účel používali vývojhové prostředí Control Web.

Tvorba simulačního modelu i tvorba simulátoru spolu úzce navzájem souvisejí (obr. 51) – předpokladem pro tvorbu výukového simulátoru je dostatečně dobře verifikovaný model, na druhé straně, využití simulátoru ve výuce přináší nové požadavky na vytvoření nových, či modifikaci stávajících simulačních modelů.

Pokud pro tvorbu simulačních modelů a pro vytváření vlastního simulátoru používáme odlišné vývojové nástroje, musíme pak zajistit dostatečně flexibilní přenos výsledků z jednoho vývojového prostředí do druhého.

Tak např. modifikujeme-li simulační model v některém nástroji pro tvorbu simulačních modelů (např. v prostředí Matlab/Simulink), je výhodné zajistit, aby se změny v modelu bez větších potíží mohly rychle promítnout do aktualizace těchto změn ve vlastním simulátoru (vyvíjeném třeba v prostředí Visual Studio .NET)

K usnadnění tohoto přenosu je vhodné si vytvořit i vlastní softwarové pomůcky nebo využít specializované integrované vývojové nástroje.

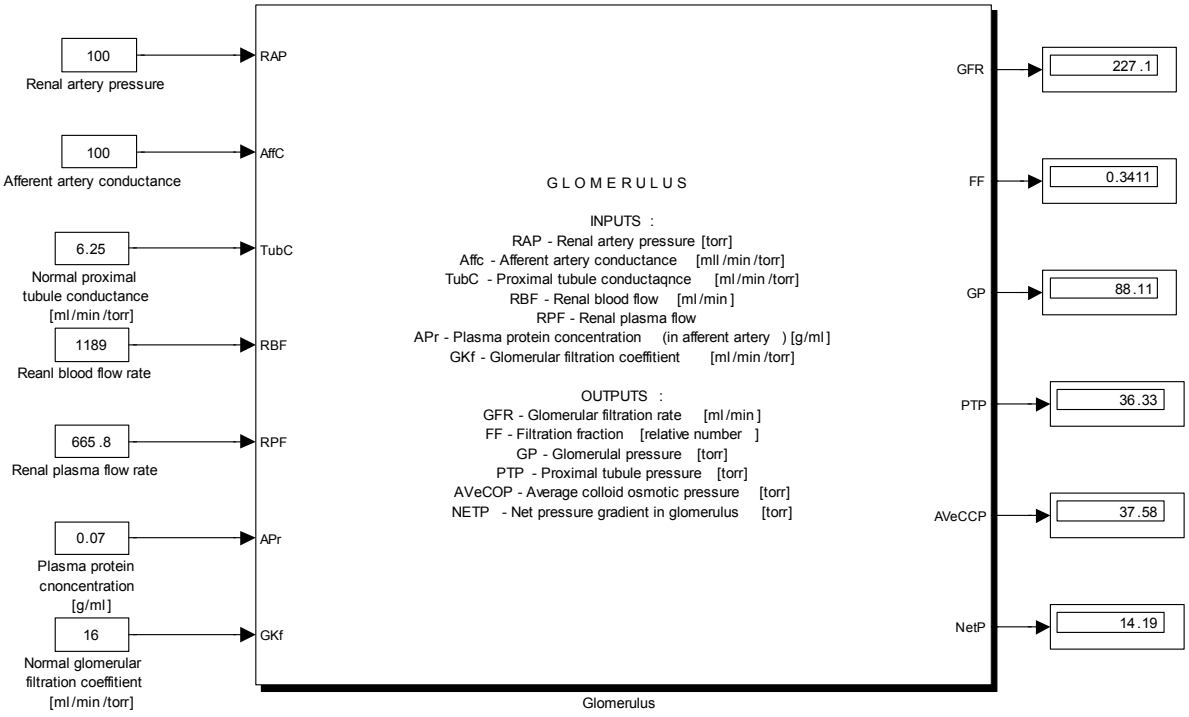
4.2 Dřinu strojům – moderní softwarové nástroje pro tvorbu simulačního jádra výukových programů

4.2.1 Blokově orientované simulační sítě

Guytonovo rozsáhlé blokové schéma modelu řízení krevního oběhu (obr. 1 na straně 3) předznamenalo vznik vizuálních blokově orientovaných simulačních jazyků. Grafická notace tohoto schématu silně připomíná simulinkové bloky používané ve vývojovém prostředí Matlab/Simulink od firmy Mathworks. Sám Guyton a jeho spolupracovníci ovšem tenkrát, v roce 1972, model implementovali v jazyku Fortran – Simulink ve verzi 1 byl uveden až trh až o osmnáct let později (v roce 1990).

Blokově orientované simulační jazyky, jejichž typickým představitelem je právě Simulink, umožňují sestavovat počítačové modely z jednotlivých bloků, s definovanými vstupy a výstupy. Bloky jsou seskupeny v knihovnách a pomocí počítačové myši se při tvorbě modelu vytvářejí jejich jednotlivé instance, jejichž vstupy a výstupy se propojují pomocí vodičů, kterými „proudí“ informace.

Simulinkovou síť je možné hierarchicky uspořádat. Bloky, je možno seskupovat do jednotlivých subsystémů, které s jejich vnějším okolím komunikují prostřednictvím definovaných vstupních a výstupních „pinů“ a představují tak jakési „simulační čipy“. Simulační čip skrývá před uživatelem strukturu simulační sítě, obdobně jako elektronický čip ukrývá před uživatelem propojení jednotlivých tranzistorů a dalších elektronických prvků. Uživatel se pak může zajímat pouze o chování čipu a nemusí se starat



Obr. 52 – Ukázka simulačního čipu „GLOMERULUS“ (realizovaného simulinkovým blokem) počítajícího glomerulární filtraci. Struktura výpočtu je před uživatelem skryta. V prostředí Simulinku je možno snadno otestovat jeho chování – k jednotlivým „vstupním pinům“ lze přivést vstupní hodnoty (nebo průběhy hodnot) a od „výstupních pinů“ na virtuálních displejích či osciloskopech odečítat výstupy, resp. časové průběhy výstupů.

o vnitřní strukturu a algoritmus výpočtu. Chování simulačního čipu pak může testovat pomocí sledování výstupů na připojených virtuálních displejích či na virtuálních osciloskopech (obr. 52).

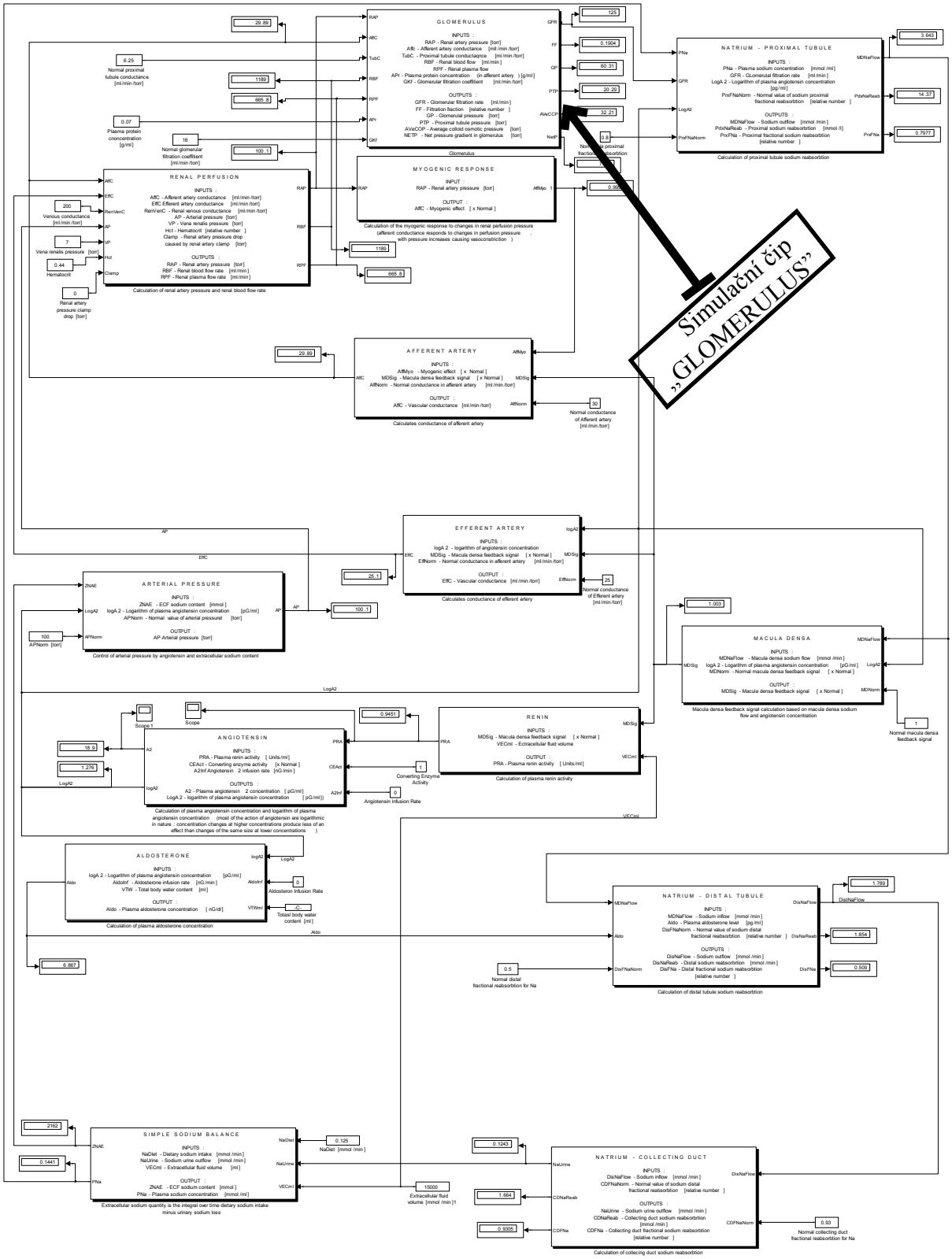
Pomocí simulačních čipů lze snadněji testovat chování modelu a zejména přehledněji vyjádřit vzájemné závislostí mezi proměnnými modelovaného systému. Celý složitý model pak můžeme zobrazit jako propojené simulační čipy a ze struktury jejich propojení je jasné, jaké vlivy a jakým způsobem se v modelu uvažují (obr. 53).

To je velmi výhodné pro mezioborovou spolupráci – zejména v hraničních oblastech jakým je např. modelování biomedicínských systémů (Kofránek, Andrlík, Kripner & Mašek, 2002c). Experimentální fyziolog nemusí dopodrobna zkoumat, jaké matematické vztahy jsou ukryty „uvnitř“ simulačního čipu, z propojení jednotlivých simulačních čipů mezi sebou však pochopí strukturu modelu a jeho chování si může ověřit v příslušném simulačním vizualizačním prostředí. Pro fyziologa je např. mnohem přehlednější Guytonův model graficky vyjádřený ve formě propojených simulačních čipů (obr. 54 a 55) než jeho implementace spletitou simulinkovou sítí – stačí porovnat implementaci klasického Guytonova modelu na obr. 54 s implementací na obr. 1 na straně 3.

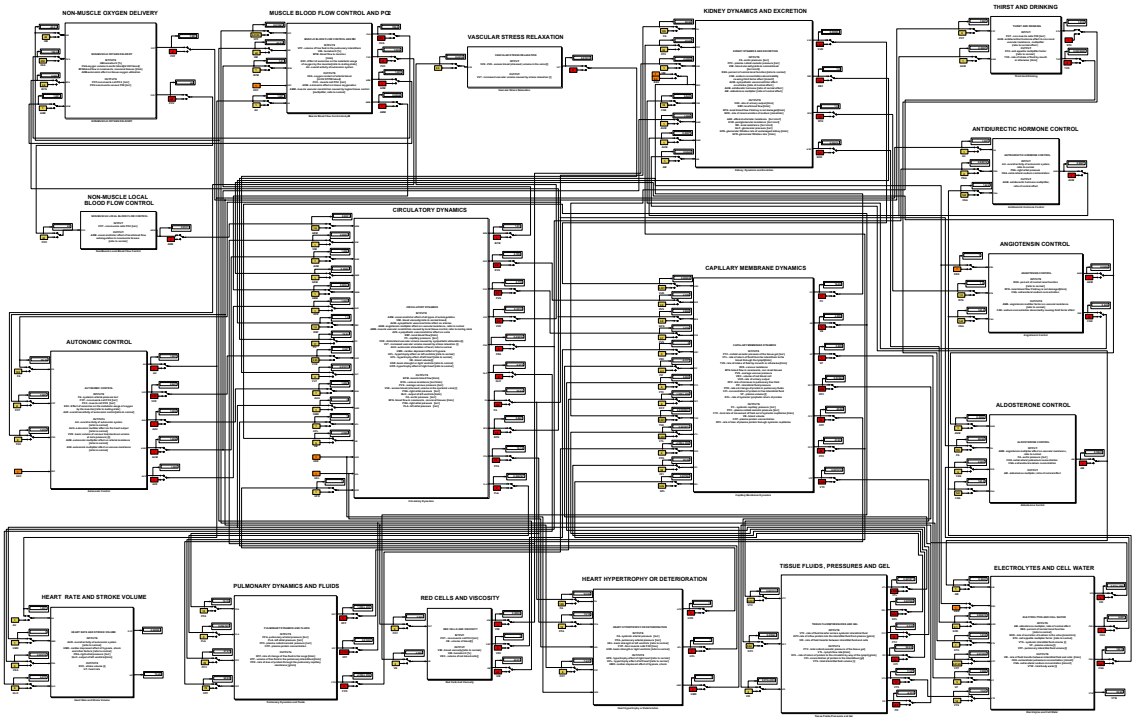
V masce simulinkových bloků z nichž vytváříme simulační čipy, popisujeme stručně význam všech vstupů a výstupů (včetně fyzikálních jednotek), případně k blokům vytvoříme i příslušnou dokumentaci dosažitelnou na kliknutí na připojenou nápovědní stránku. Maska bloku může mít i tvar ikonky graficky znázorňující funkci simulačního čipu (například znázornění elektrické analogie při modelování tlaků a průtoků v cirkulačním a respiračním systému (obr. 56). Toho jsme například využívali při tvorbě simulačních čipů využitelných pro modely cirkulace a respirace.

Dobrým implementačním pravidlem je udržovat pokud možno hierarchické uspořádání simulačních čipů (obr. 57), které na jedné straně zpřehledňuje strukturu modelu a napomáhá multidisciplinární komunikaci, na druhé straně podporuje znovupoužitelnost vytvořených simulačních bloků v jiných modelech. Hierarchické uspořádání měla i simulinková implementace našeho modelu Golem (viz obr. 39-41).

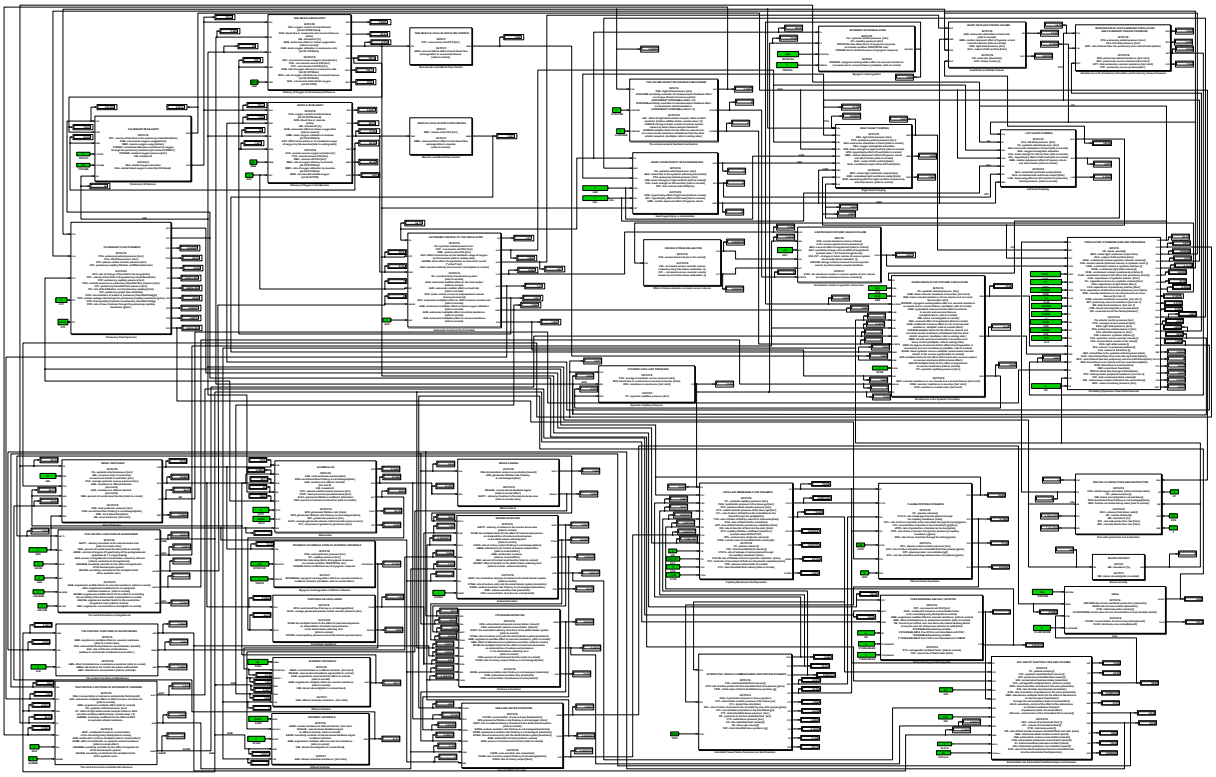
Simulační čipy je možné ukládat do knihoven a uživatel může vytvářet jejich instance pro použití



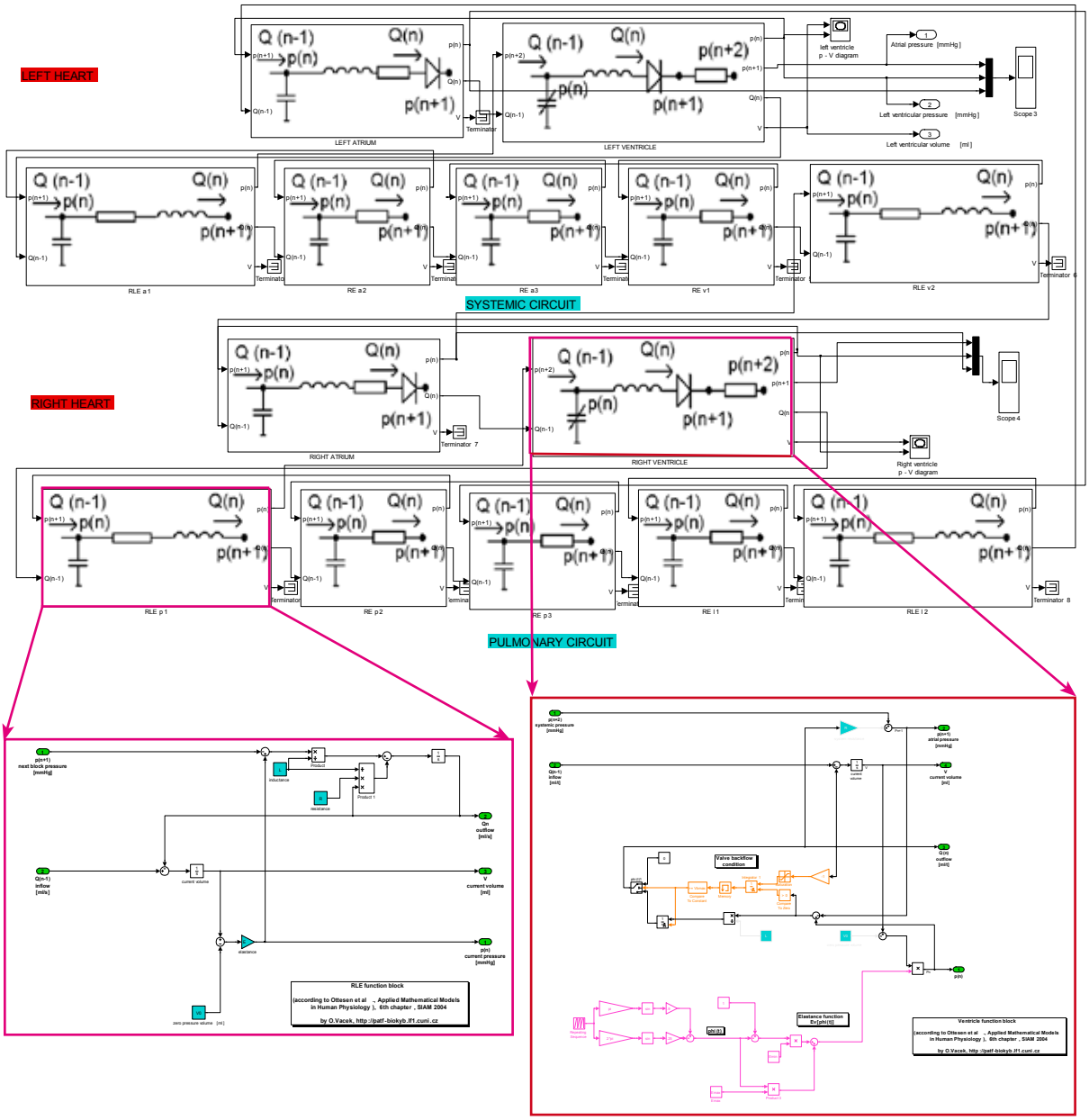
Obr. 53 – Zapojení imulačního čipu „GLOMERULUS“ z předchozího obrázku v jednoduchém modelu ledvin. Propojení jednotlivých simulinkových bloků (simulačních čipů) je srozumitelné i pro experimentálního fyziologa.



Obr. 54 – Model A.C.Guytona a spol. z roku 1972 agregovaný do propojených simulinkových bloků. Má stejnou funkcionalitu jako model z obr. 1 na straně 3, ale pro fyziologa, zajímajícího se o chování modelu a modelované fyziologické závislosti, je srozumitelnější.



Obr. 55 – Verze Guytonova modelu z roku 1986 implementovaná pomocí simulinkových bloků. Ve srovnání s předchozím obrázkem je zřetelné kde a jak se model, čtrnáct let po uveřejnění jeho první verze, rozrostl. Model byl původně implementován ve Fortranu (ze zdrojového textu ale není struktura modelu na první pohled zřetelná; ve zdrojovém textu ve Fortranu se promíchávaly jak příkazy, reprezentující matematické vztahy modelu, tak i problematika sofistikovaného řešení algebrodiferenciálních rovnic pomocí proměnných integračních kroků, různých pro jednotlivé části modelu). Podrobný popis struktury modelu včetně popisu a zdůvodnění všech matematických vztahů je k dispozici na adrese http://physiome.cz/guyton.

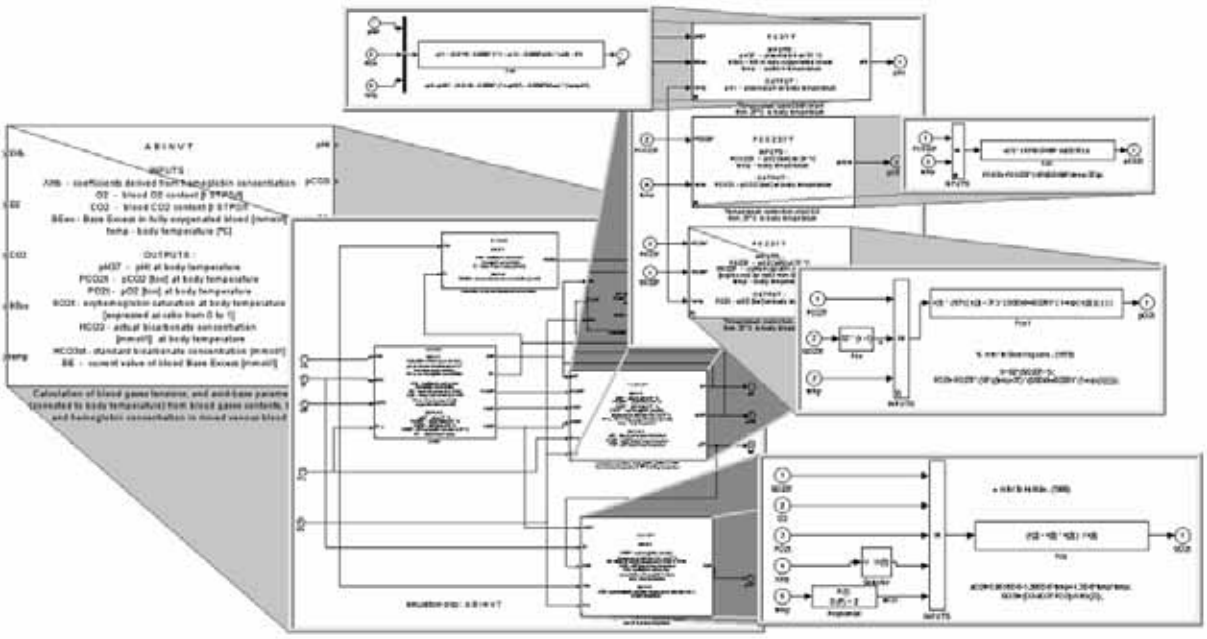


Obr. 56 – Simulační čipy mohou mít v masce i grafické znázornění jejich funkce. Příkladem je model cirkulace, vytvořený ze simulačních čipů knihovny Physiolibrary, v jejichž masce je znázorněna elektrická analogie matematických vztahů příslušných subsystémů modelu. Jednotlivé čipy počítají výstupní hodnoty ze vstupních. Model podle kapitoly 6 z knihy Otessena a spol. (Ottesen, Olufsen & Larsen, 2004) implementoval můj doktorand O. Vacek.

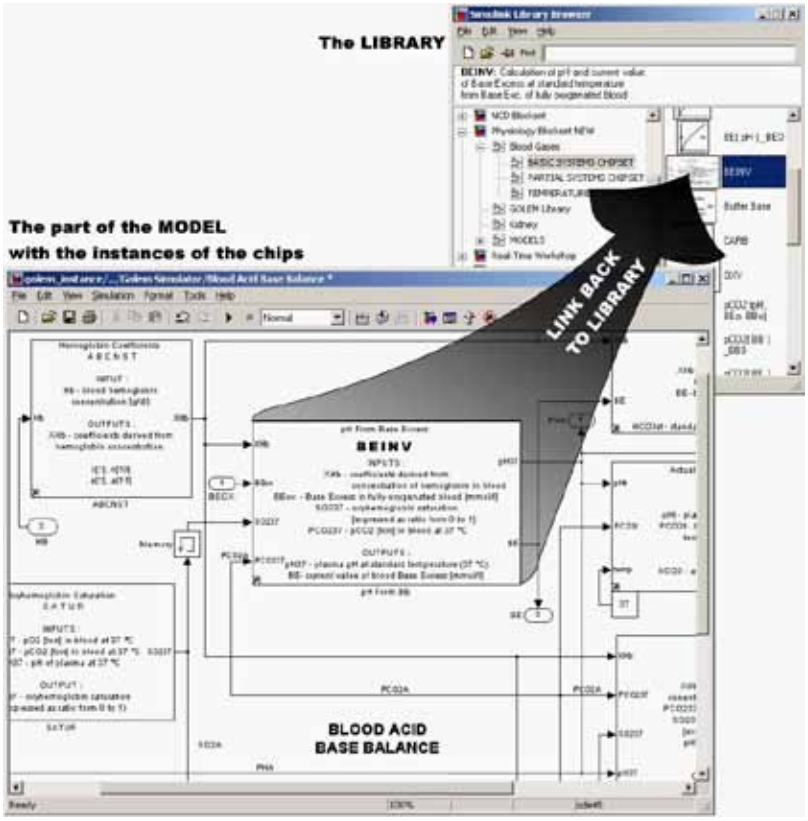
ve svých modelech (obr. 58). Ze simulačních simulinkových čipů jsme vytvořili ***knihovnu Physiolibrary pro modelování fyziologických regulací.*** Knihovna obsahuje i dokumentaci přímo zaintegrovanou do nápovědy k Matlab/Simulink (obr. 59). Knihovnu i příslušný instalátor lze bez omezení stáhnout z adresy http://physiome.cz/simchips.

Hierarchické blokově orientované simulační nástroje proto našly svoje velké uplatnění při popisu složitých regulačních systémů, s nimiž se setkáváme ve fyziologii a uplatňují se při řešení mezinárodních projektů PHYSIOME a EUROPHYSIOME (http://www.physiome.org/, http://www.europhysiome.org/)

V rámci projektu PHYSIOME bylo vytvořeno několik blokově orientovaných simulačních nástrojů, které slouží jako referenční databáze pro formalizovaný popis struktury složitých fyziologických modelů. Patří k nim zejména simulační nástroj i jazyk JSIM (Raymond, Butterworth & Bassingthwaighte, 2003), http://physiome.org/model/doku.php, a také jazyk CEllML (Lloyd, Halstead & Nielsen, 2004),

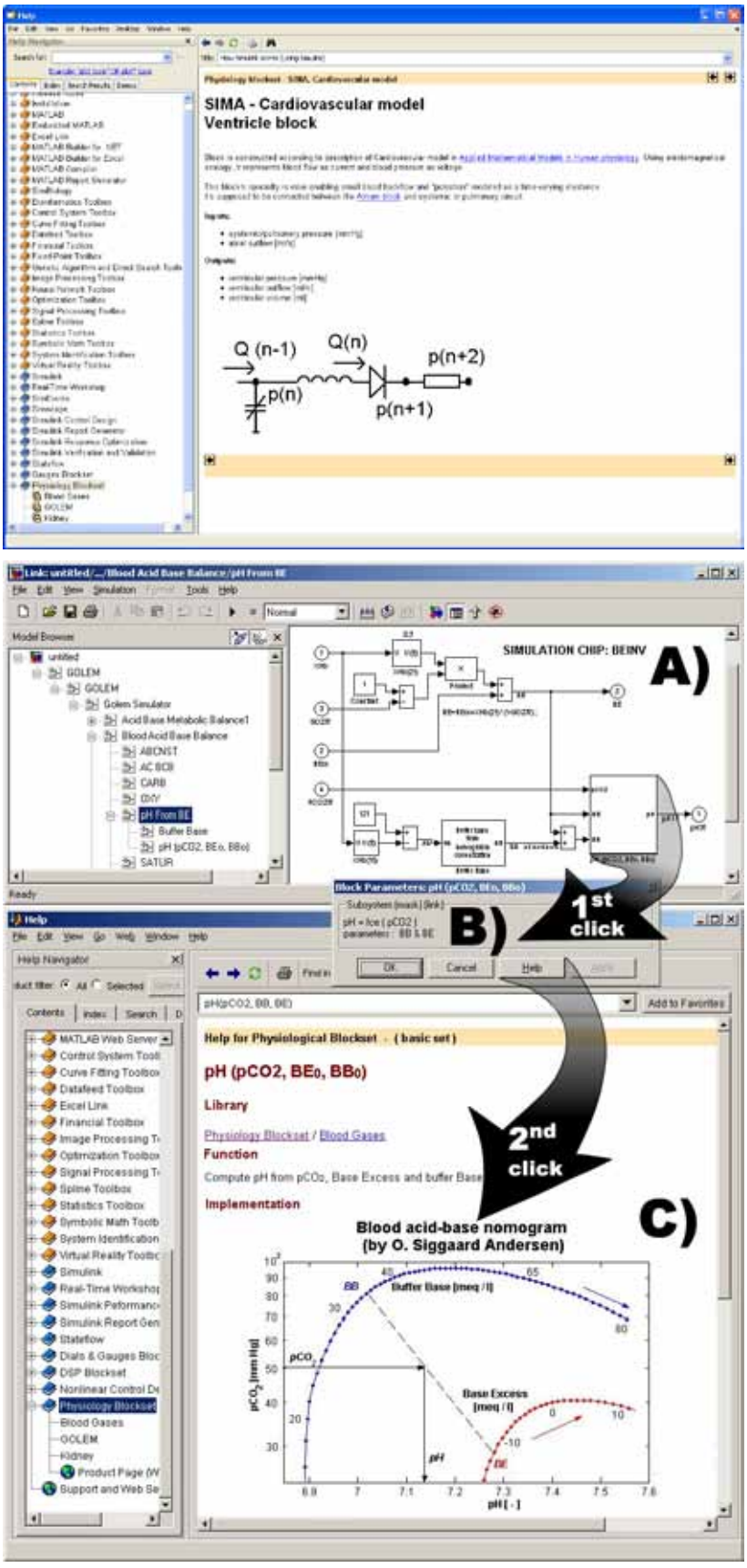


Obr. 57 – Simulační čipy mají hierarchické uspořádání a pokud to jde, skrývají před uživatelem implementační detaily.



Obr. 58 – Simulační čipy mohou být ve vývojovém prostředí Simulink soustřeďovány do hierarchicky uspořádaných knihoven. Z nich pak lze jednotlivé čipy „vytahovat“ pomocí myši (jako z palety nástrojů), umísťovat je do vytvářené aplikace, propojovat je a vytvářet složitější modely.

http://www.cellml.org/. Pro oba jazyky existuje volně stažitelné prostředí pro vývoj i simulačních modelů a především jsou v nich vytvářeny volně dostupné referenční databáze biomedicínských modelů.



Obr. 59 – Naše knihovna Physiolibrary obsahuje i dokumentaci použitých čipů, automaticky zaintegrovatelnou do dokumentace systému Matlab/Simulink. Ze simulinkových bloků je potom mimo jiné dokumentace snadno přístupná na kliknutí myší.

4.2.2 Kauzální a akauzální přístup

Blokově orientované nástroje pracují s hierarchicky propojenými bloky. V propojkách mezi jednotlivými bloky „tečou“ signály, které přenášejí hodnoty jednotlivých proměnných od výstupu jednoho bloku k vstupům dalších bloků. V blocích dochází ke zpracování vstupních informací na výstupní.

Z hierarchicky uspořádaného blokově orientovaného popisu je jasné, jakým způsobem se v modelu počítají hodnoty jednotlivých proměnných – tj. jaký je algoritmus výpočtu.

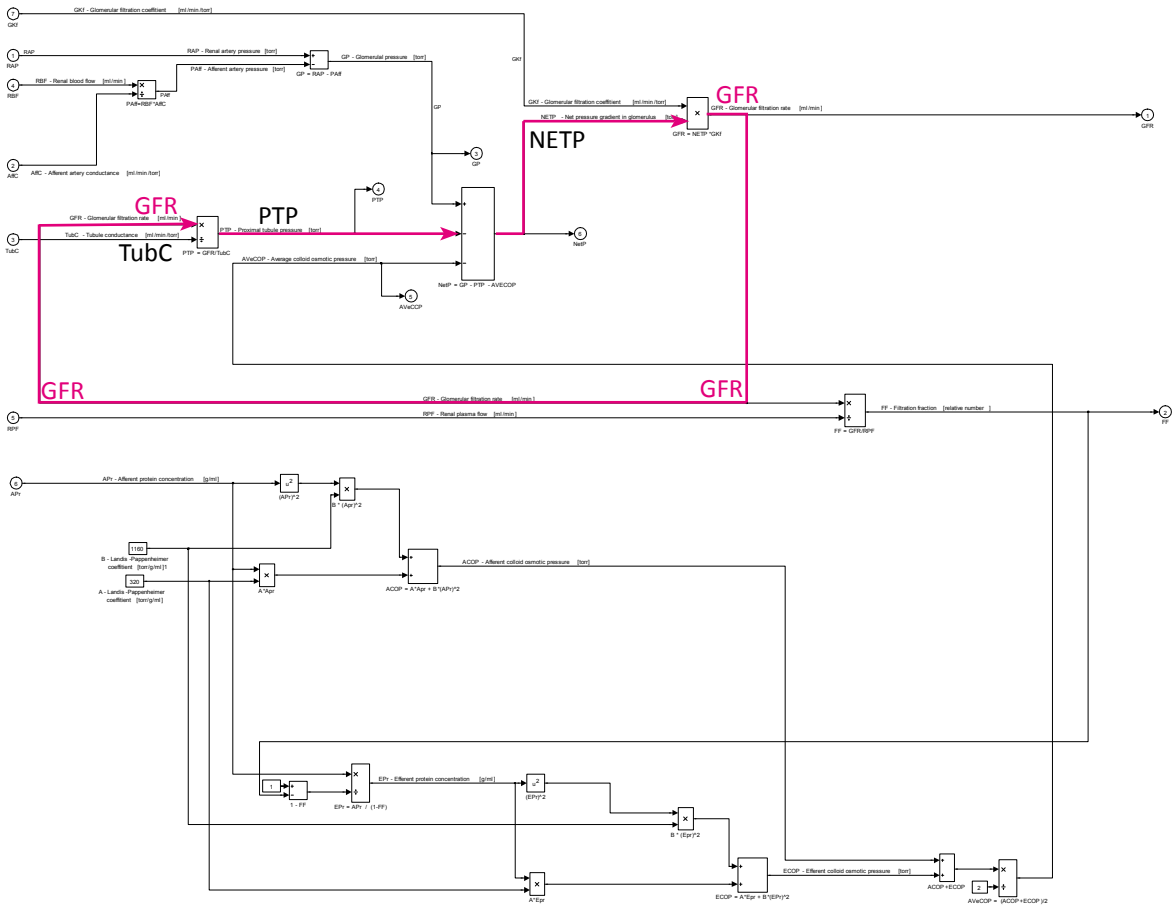
Propojování bloků do sítě vztahů ale bohužel nemůže být zcela libovolné. V propojených prvcích se **nesmějí vytvářet algebraické smyčky** – tj. cyklické struktury, kdy nějaká vstupní hodnota přiváděná jako vstup do výpočetního bloku ve stejném časovém kroku závisí (přes několik prostředníků) na výstupní hodnotě z tohoto bloku

Pro ilustraci uveďme malý příklad algebraické smyčky v blokově orientovaném jazyku Simulink.

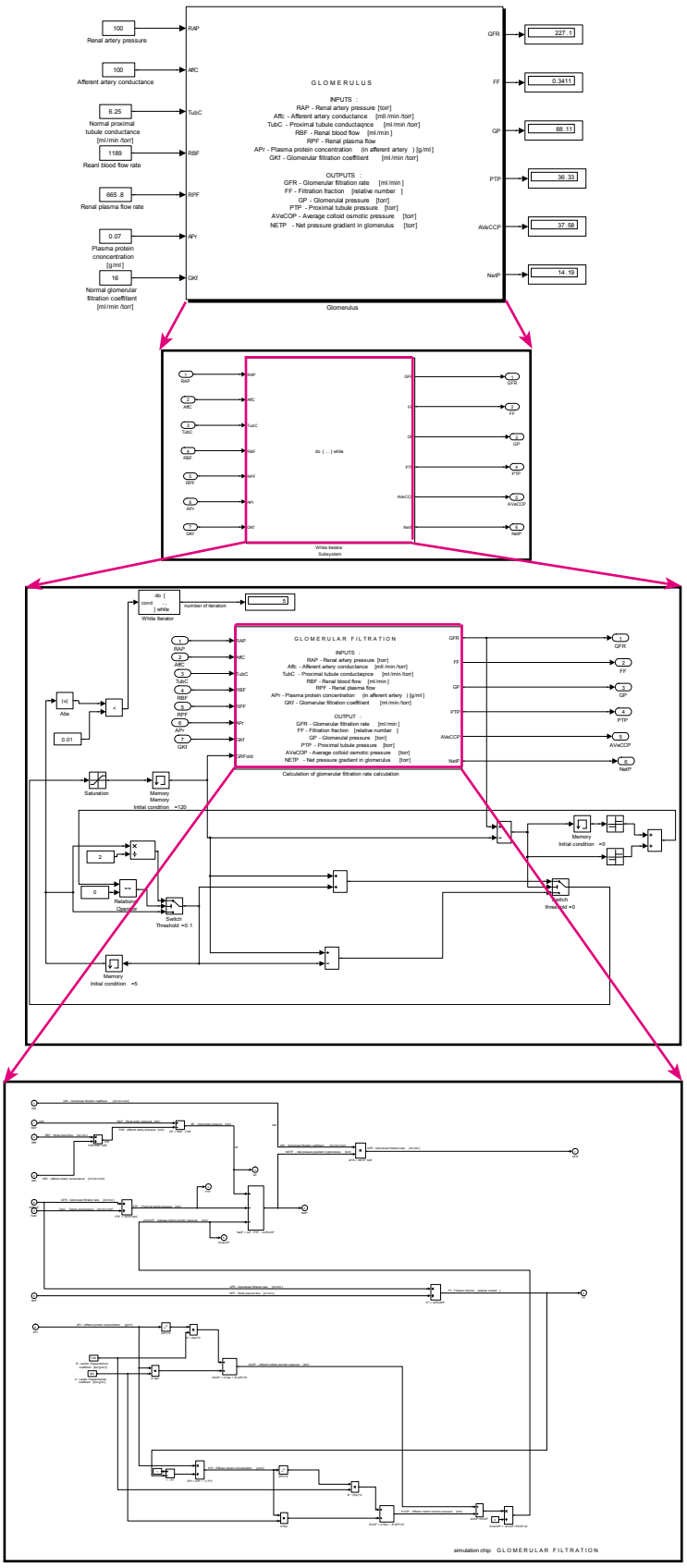
Ve výše uvedeném modelu ledvin (obr. 53) se využívá simulační čip počítající hodnotu glomerulární filtrace. Jednotlivé vstupy a výstupy do tohoto čipu jsou zobrazeny na obr. 52. Vnitřek simulačního čipu je tvořen elementárními bloky provádějícími matematické operace.

Pokud ale bez rozmyslu implementujeme strukturu matematických vztahů modelu do simulinkové sítě počítacích bloků (tj. pomocí násobiček, děliček,sumátorů a funkčního bloku vytvoříme grafickou reprezentaci matematických vztahů), snadno vytvoříme algebraickou smyčku (obr. 60). Hodnota proměnné *GFR*, reprezentující hodnotu glomerulární filtrace je počítána z hodnoty *NETP*, k výpočtu hodnoty *NETP* je zapotřebí znát hodnotu proměnné *PTP*, která je ale počítána jako podíl hodnot proměnných *GFR* a *TUBC*.

V Simulinkovém schématu máme algebraickou smyčku, kterou je nutno přerušit. Řešení je nazna-



Obr. 60 – Algebraická smyčka v simulinkové implementaci simulačního čipu „GLOMERULUS“ počítajícího hodnotu glomerulární filtrace (GFR).



Obr. 61 – Po přerušení algebraické smyčky se vnitřní struktura simulačního čipu „GLOMERULUS“ stala poněkud složitější, nicméně navenek se blok nezměnil a pro jeho uživatele vypadá stále stejně. Vnitřní propojení numerických simulinkových bloků nesmí obsahovat algebraické smyčky, odráží proto spíše postup výpočtu než grafické znázornění matematických vztahů.

čeno na obr. 61 a souvisí s tím, že pomocí nástrojů, které Simulink poskytuje, v každém integračním kroku vlastně počítáme hodnotu ***GFR*** řešením implicitní rovnice.

Simulinková síť proto netvoří grafické zobrazení matematických vztahů v modelu, ale zobrazuje spíše grafické vyjádření řetězce transformací vstupních hodnot na výstupní přes jednotlivé simulinkové elementy, kde cyklení není dovoleno.

Pokud při stavbě modelu v Simulinku budeme myslet spíše na zobrazení struktury matematických vztahů než na algoritmus výpočtů, snadno do modelu algebraické smyčky zaneseme (na což nás ovšem kompilátor upozorní). Existují metody, jak se algebraických smyček zbavit (Dabney & Harman, 2004), vedou však k takovým transformacím, které (jak je ostatně vidět na výše uvedeném příkladě) strukturu modelu dále zesložití a model je méně přehledný. Požadavek pevně zadaného směru spojení od vstupů k výstupům s vyloučením algebraických smyček vede i k náročnější stavbě modelu.

Propojení bloků v Simulinku proto ***odráží spíše postup výpočtu než vlastní strukturu modelované reality.*** Hovoříme o tzv***. kauzálním modelování***.

U složitých systémů se díky tomuto přístupu ***pod strukturou výpočtu pomalu ztrácí fyzikální realita modelovaného systému***.

V poslední době došlo k vývoji nových tzv. **„akauzálních“ nástrojů** pro tvorbu simulačních modelů. Zásadní inovaci, kterou akauzální modelovací nástroje přinášejí je možnost popisovat jednotlivé části modelu přímo ***jako soustavu rovnic a nikoli jako algoritmus řešení těchto rovnic***. Zápis modelů je deklarativní (popisujeme strukturu a matematické vztahy, nikoli algoritmus výpočtu) – zápis je tedy akauzální.

Akauzální modelovací nástroje pracují s propojenými komponentami, které představují instance tříd, v nichž jsou přímo definovány rovnice.

Tyto komponenty (tj. instance tříd s rovnicemi) se mohou propojovat prostřednictvím přesn**ě definovaných rozhraní – konektorů a definovat tak** soustavy rovnic.

Určité možnosti uplatnění akauzálního nástrojů přinášejí od roku 2007 i nové verze Simulinku (od R2007a). Výrobce simulačních nástrojů Matlab/Simulink – firma Mathworks reagovala na nové trendy vytvořením speciální akauzální simulinkové knihovny Simscape (v roce 2007 se objevila verze 1.0 a v roce 2009 je již ve verzi 3.2). Mathworks distribuuje i návazné doménové akauzální knihovny SimElectronics, SimHydraulics, SimMechanics SimDriveline a SimPowerSystems.

Moderním simulačním jazykem, který je přímo postaven na akauzálním zápisu modelů je ***Modelica*** (Fritzon, 2003). Byl původně vyvinut ve Švédsku a nyní je dostupný jak ve verzi open-source (vyvíjené pod záštitou mezinárodní organizace Modelica Association, http://www.modelica.org/), tak i v několika komerčních implementacích. Nejrozšířenější komerční implementace je od firmy Dynasim AB – kterou nyní koupil nadnárodní koncern Dassault Systemes (prodává se pod názvem ***Dymola***, nyní již ve verzi 7.3). Další komerční implementace pochází od firmy MathCore (prodává se pod názvem ***MathModelica***). Modelica od Dynasimu má dobré napojení na simulační nástroje Matlab a Simulink, zatímco MathModelica se umí propojovat s prostředím Mathematica od firmy Wolfram.

Modelica pracuje s propojenými komponentami, které představují instance jednotlivých tříd. Na rozdíl od implementace tříd v jiných objektově orientovaných jazycích (jako je C#, Java apod.), mají třídy v Modelice navíc zvláštní sekci, v níž se definují ***rovnice***.

Rovnice neznamenají přiřazení (tj. uložení výsledku výpočtu přiřazovaného příkazu do dané proměnné), ale definici vztahů mezi proměnnými (tak, jak je v matematice a fyzice zvykem). Tak například následující zápisy vztahu mezi proměnnými vyjadřující odpor (*R*) tok (*F*) a tlakový gradient (*P*) jsou ekvivalentní:

$$F=P/R$$

$$P/R=F$$

$$P=R*F$$

$$R*F=P$$

$$R=P/F$$

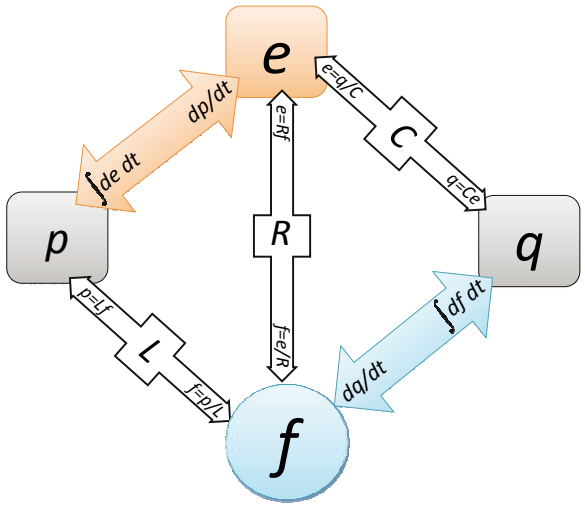
$$P/F=R$$

Komponenty (instance tříd) se mohou v Modelice propojovat prostřednictvím přesně definovaných rozhraní – ***konektorů***.

Důležité je to, že propojením komponent vlastně dochází k ***propojení soustav rovnic v jednotlivých komponentách*** mezi sebou. ***Propojením modelicových komponent tedy nedefinujeme postup výpočtu, ale modelovanou realitu. Způsob řešení rovnic pak „necháváme strojům“.***

4.2.3 Zobecněné systémové vlastnosti

Zobrazení modelu v akauzálním simulačním prostředí tak více připomíná fyzikální realitu modelovaného světa než klasická propojená bloková schémata v kauzálních modelovacích nástrojích. Souvisí to se zobecněnými systémovými vlastnostmi reálného světa (obr. 62), v nichž hrají důležitou roli ***zobecněné úsilí*** (v reálném světě mu odpovídá síla, tlak, napětí apod.) a ***zobecněný tok*** (jemuž v reálném světě mu odpovídá proud, průtok aj.). Integrálem zobecnělého toku je ***zobecněná akumulace*** **či výchylka** (v reálném světě může například představovat elektrický náboj, ale také i objem tekutiny nebo plynu, natažení pružiny, naakumulované teplo apod.). Integrálem zobecnělého úsilí je ***zobecněná hybnost*** (v hydraulice představuje průtočnou hybnost, v elektrických obvodech indukci apod.).



*Obr. 62 – Vztahy mezi zobecněnými systémovými vlastnostmi. „**e**“ znamená zobecněné úsilí (effort) – v mechanice mu odpovídá síla, v elektrických schématech napětí, v hydraulice tlak atd. „**f**“ je zobecněný tok (flow) – v mechanice mu odpovídá rychlost, v elektrických schématech proud, hydraulice průtok a v termodynamice teplotní tok apod. „**q**“ je zobecněná akumulace, či výchylka, reprezentuje integrál zobecnělého toku. V mechanice jí např. odpovídá natažení pružiny, v hydraulice objem tekutiny, v elektrických schématech náboj, v termodynamice aakumulované teplo atd. „**p**“ je zobecněná hybnost (inertance) – integrál zobecnělého úsilí, reprezentující kinetickou energii, v hydraulice představuje změnu rychlosti proudu úměrnou rozdílu tlaků (průtočnou hybnost), v elektrických obvodech potenciál nutný ke změně elektrického proudu (indukce) apod. „**R**“, „**C**“ a „**L**“ představují konstanty úměrnosti mezi jednotlivými zobecněnými systémovými vlastnostmi. Odpovídá jim např. odpor, kapacitance či hmotnost.*

Se zobecněnými systémovými vlastnostmi souvisí i to, že se nezřídka při popisu modelů biologických a fyziologických dějů se pro názornost využívá elektrická či hydraulická analogie. Využití zobecněných systémových vlastností si a rozdíl mezi modelováním v blokově orientovaných simulačních nástrojích a v Modelice si ilustrujme na příkladu modelování fyziologické reality – na modelu jednoduché mechaniky plicní ventilace.

Uvažujme ***jednoduchý model mechaniky plic***, který schematicky zobrazuje obr. 63. Plíce si ve velkém zjednodušení představíme jako tři vaky propojené dvěma trubicemi. Plíce jsou připojeny k ventilátoru umělé plicní ventilace, který periodicky vhání tlakem *PAO* vzduch do plic. *P0* je tlak okolní atmosféry. Proud vzduchu *Q* proudí skrze horní cesty dýchací, jejichž odpor je *RC*. Z horních cest dýchacích se vzduch prodírá dolními dýchacími cestami do alveolů. Odpor dolních dýchacích cest je *RP*, tlak v centrálních partiích dýchacích cest (na rozhraní horních a dolních dýchacích cest) je *PAW*, tlak v alveolech je *PA*.

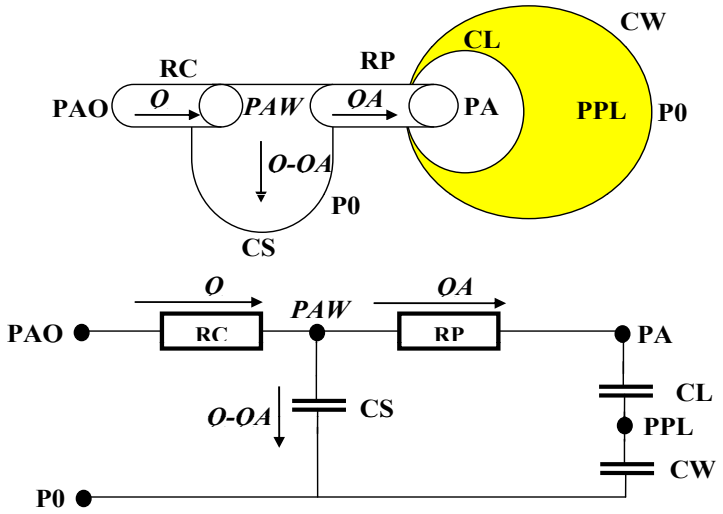
Vzduch roztahuje plicní alveoly, jejichž poddajnost je *CL* (jako celková poddajnost plic). Mezi plícemi a hrudním košem je interpleurální dutina. Tlak v ní je *PPL*. Při umělé plicní ventilaci, kdy se

pod tlakem vhání vzduch do plic, se ještě musí roztáhnout hrudník – poddajnost hrudníku je *CW*. Malá část vzduchu, která se nedostane až do alveolů, pouze roztahuje dýchací cesty – jejich poddajnost je *CS*.

Nyní si můžeme sestavit rovnice. Podle Ohmova zákona musí platit:

$$\begin{aligned} PAW - PA &= RP\,QA \\ PAO - PAW &= RC\,Q \end{aligned} \tag{1}$$

Vztah mezi poddajností, tlakovým gradientem a objemem (vyjádřeným jako integrál průtoku) popisují rovnice:



Obr. 63 – Jednoduchý model plicní mechaniky (hydraulická a elektrická analogie).

$$\begin{aligned} PA - PPL &= \frac{1}{CL}\int QA\,dt \\ PPL - P0 &= \frac{1}{CW}\int QA\,dt \\ PAW - P0 &= \frac{1}{CS}\int (Q - QA)\,dt \end{aligned} \tag{2}$$

Podle zobecněného Kirchhoffova zákona musí být součet všech tlaků (napětí) podél uzavřené smyčky rovný nule, tj. ve smyčce podél uzlu *PAW* a podél uzlu *PA0* musí platit:

$$\begin{aligned} &(PAW - PA) + (PA - PPL) + (PPL - P0) + (P0 - PAW) = 0 \\ &(PAO - PAW) + (PAW - P0) + (P0 - PAO) = 0 \end{aligned} \tag{3}$$

Po dosazení z rovnic Ohmova zákona a poddajností dostaneme:

$$\begin{cases} RP\,QA + \left(\frac{1}{CL} + \frac{1}{CW}\right)\int QA\,dt - \frac{1}{CS}\int (Q - QA)\,dt = 0 \\ Q\,RC + \frac{1}{CS}\int (Q - QA)\,dt + (P0 - PAO) = 0 \end{cases} \tag{4}$$

4.2.4 Kauzální přístup – implementace modelu mechaniky plicní ventilace v Simulinku

Stavíme-li model v Simulinku, musíme přesně určit ***postup výpočtu*** ze vstupních proměnných na výstupní. Chceme-li počítat reakci toku vzduchu do/z plic (*Q*) na vstup – tj. na změny tlaku na začátku dýchacích cest (*PAO*) způsobované aparátem umělé plicní ventilace – bude simulinkový model vypadat jako na obr. 64.

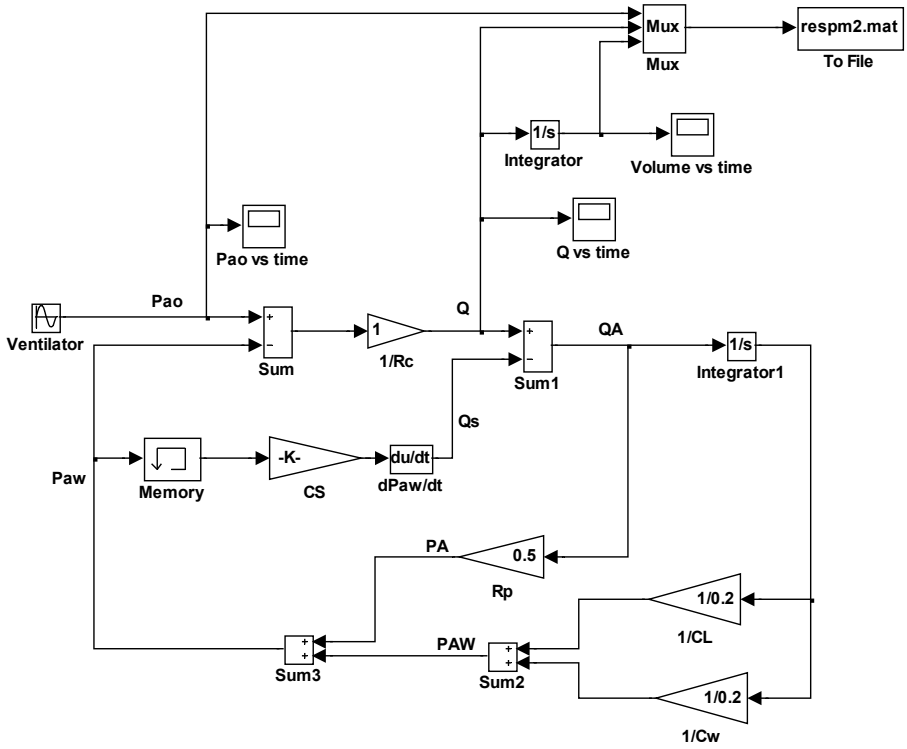
Simulinkový model můžeme vyjádřit i jednodušeji. Nejprve z rovnic (4) odvodíme diferenciální rovnici (vstupní proměnná *PAO*, výstupní *Q*):

$$\frac{d^2PAO}{dt^2}+\frac{1}{RP\ CT}\frac{dPAO}{dt}=RC\frac{d^2Q}{dt^2}+\left(\frac{1}{CS}+\frac{RC}{RP\,CT}\right)\frac{dQ}{dt}+\frac{1}{RP\,CS}\left(\frac{1}{CL}+\frac{1}{CW}\right)Q \tag{5}$$

Při zadání číselných hodnot parametrů odporů (v jednotkách cm H_2O/L/sec) a poddajností (v jednotkách L/cmH_2O) převzatých z (Kho, 2000):

$$RC=1;\quad RP=0,5;\quad CL=0,2;\quad CW=0,2;\quad CS=0,005 \tag{6}$$

se rovnice (5) zjednoduší:



Obr. 64 – Implementace modelu v Simulinku podle rovnic (4).

$$\frac{d^2PAO}{dt^2}+420\frac{dPAO}{dt}=\frac{d^2Q}{dt^2}+620\frac{dQ}{dt}+4000\,Q \tag{7}$$

V Laplaceově transformaci rovnice (7) dostaneme:

$$\frac{Q(s)}{PAO(s)}=\frac{s^2+420s}{s^2+620s+4000} \tag{8}$$

To umožní simulinkový model zjednodušit (obr. 65):

Při změnách hodnot parametrů se ale musí transformační funkce (7) přepočítat a simulinkový model se změní.

Nyní model mírně zesložitíme tak, že budeme uvažovat ***inerci vzduchu*** v horních dýchacích cestách (obr. 66).

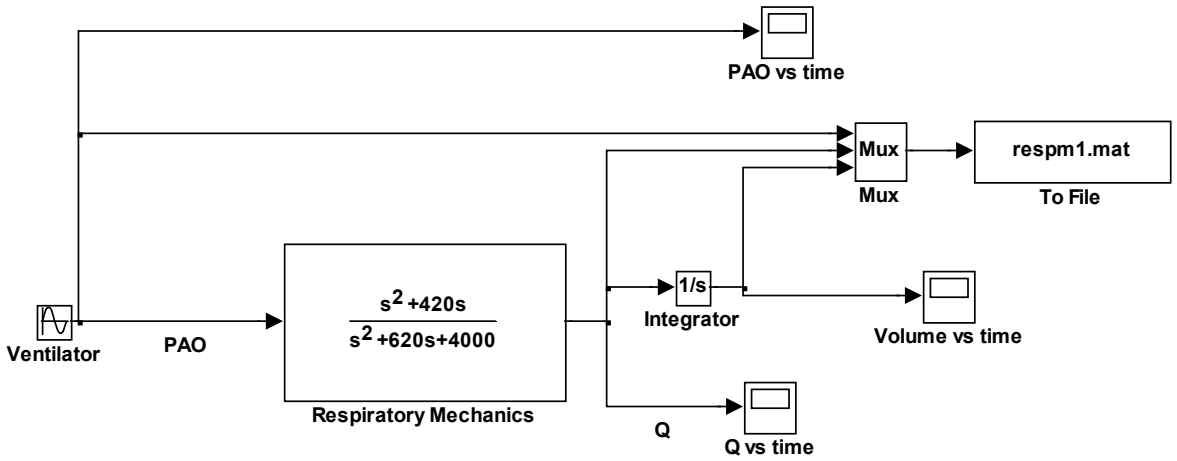
Nyní navíc uvažujeme inerční element *LC=0,01 cm H_2O s^2 L^{-1}*:

$$LC = \frac{\Delta P}{\frac{dQ}{dt}} \tag{9}$$

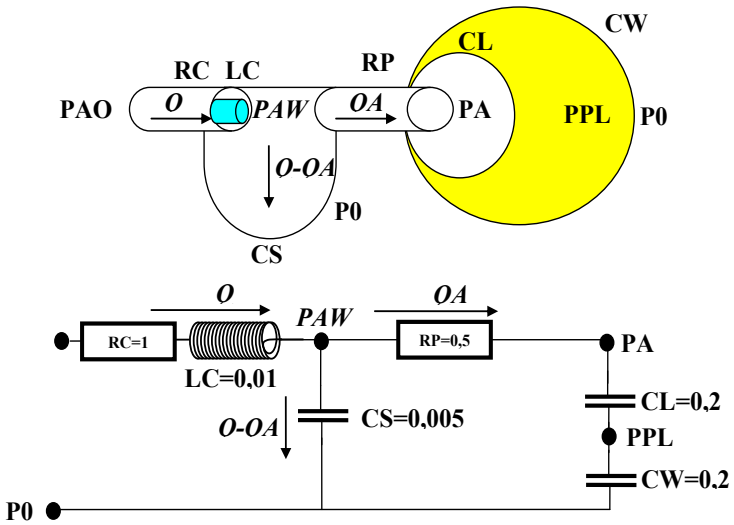
kde ***ΔP*** je tlakový gradient a **dQ/dt** je zrychlení průtoku, neboli:

$$\Delta P = LC\frac{dQ}{dt} \tag{10}$$

Potom místo soustavy rovnic (4) dostaneme:



Obr. 65 – Implementace modelu v Simulinku s využitím Laplaceovy transformace podle rovnice (7)



Obr. 66 – Jednoduchý model plicní mechaniky s uvažováním inerce (hydraulická a elektrická analogie).

$$\begin{cases} RP\dfrac{dQA}{dt}+\left(\dfrac{1}{CL}+\dfrac{1}{CW}\right)QA-\dfrac{1}{CS}(Q-QA)=0 \\ RC\dfrac{dQ}{dt}+LC\dfrac{d^2Q}{dt^2}+\dfrac{1}{CS}(Q-QA)+\dfrac{dP0}{dt}-\dfrac{dPAO}{dt}=0 \end{cases} \tag{11}$$

Místo rovnice (7) dostaneme:

$$\frac{d^2PAO}{dt^2}+420\frac{dPAO}{dt}=0{,}01\frac{d^3Q}{dt^3}+5{,}2\frac{d^2Q}{dt^2}+620\frac{dQ}{dt}+4000\,Q \tag{12}$$

a v Laplaceově transformaci dostaneme:

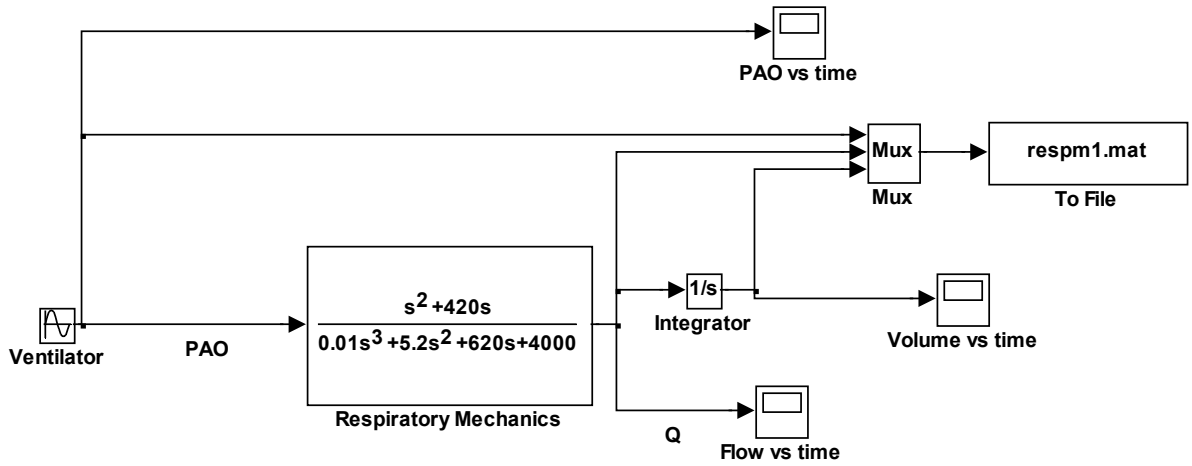
$$\frac{Q(s)}{PAO(s)}=\frac{s^2+420s}{0{,}01s^3+5{,}2s^2+620s+4000} \tag{13}$$

Simulinkový model se změní (obr. 67):

Díky tomu, že v Simulinku musíme vždy uvažovat ***směr výpočtu***, je vlastní simulinkové schéma dosti vzdáleno skutečné fyzikální realitě popisovaného systému. I malá změna v modelu, jako je přidání inerčního elementu, nutí k pečlivém propočtu a změně struktury modelu. K zásadní změně modelu dojde i tehdy, pokud bychom místo umělé plicní ventilace uvažovali spontánní dýchání. Vstupem modelu pak nebude tlak ***PA0*** vytvářený respirátorem umělé plicní ventilace, ale například poddajnost hrudní stěny ***CW*** (cyklickou změnou poddajnosti se dá modelovat funkce dechových svalů).

4.2.5 Akauzální přístup – implementace modelu mechaniky plicní ventilace v Modelice

Porovnáme-li strukturu modelu na obr. 63 a obr. 66, vyjádřenou pomocí zobecněných stavových proměnných s implementací modelu v Simulinku (obr. 64, obr. 65 a obr. 67), vidíme, že propojené bloky



Obr. 67 – Implementace modelu v Simulinku s využitím Laplaceovy transformace podle rovnice (13)

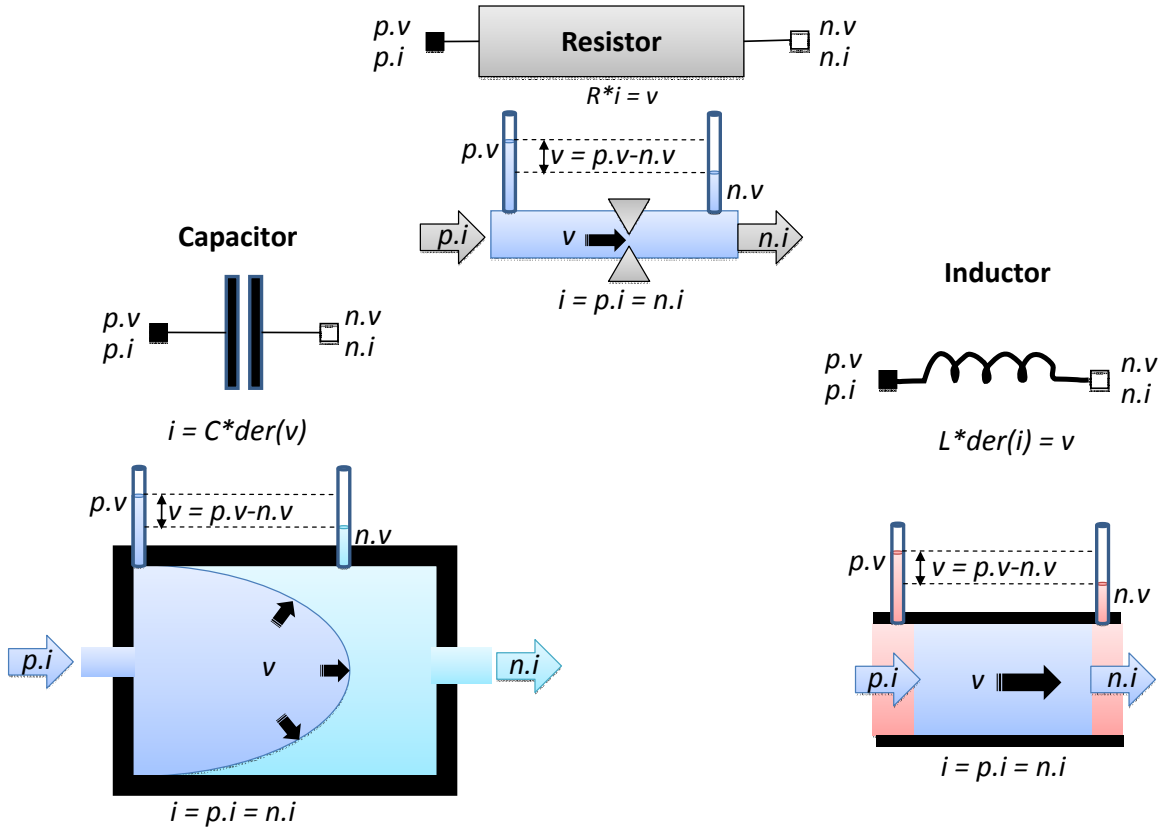
v Simulinku vyjadřují spíše strukturu postupu výpočtu než strukturu modelované reality.

V Modelice je to ale jinak. Akauzální modelovací nástroje, jejichž typickým představitelem je Modelica, pracují s propojenými komponentami, které jsou instancemi speciálních tříd, v nichž jsou definovány ***rovnice***. Prvním úkolem při modelování v Modelice je formálně vyjádřit modelovanou realitu pomocí rovnic.

V jednoduchém modelu plicní mechaniky popisujeme odpory dechových cest, roztažlivými pružnými vaky (viz obr. 63) a eventuálně také uvažujeme inerci proudu vzduchu (viz obr. 66). Popis proudění vzduchu v plicích patří do oblasti pneumatické domény. Pokud ale zanedbáme stlačitelnost plynů, můžeme popsat model pomocí hydraulické domény. Stejné formální vyjádření vystihuje i elektrická analogie.

Je zajímavé, že jednotlivé elementární prvky mají v různých doménách (elektrické, hydraulické či pneumatické) stejné formální vyjádření (obr. 68). Je to způsobeno obecnými systémovými vlastnostmi reálného světa, kde zobecnělému úsilí v daném případě odpovídá elektrické napětí nebo tlak, a zobecnělému toku elektrický proud nebo látkový tok.

V Modelice pro sestavení modelu plicní mechaniky budeme potřebovat definovat rovnice tří základních tříd, jejichž instance budeme v modelu využívat. Pro vyjádření odporů dechových cest využijeme instance třídy Resistor. Pružné dechové cesty, alveoly i hrudník popíšeme jako pružné vaky pomocí instancí třídy Capacitor a inerci proudu vzduchu vyjádříme pomocí instance třídy Inductor.



Obr. 68 – Hydraulické a elektrické prvky pocházejí z různých domén, mají však stejný formální popis. Analogií elektrického napětí (v) je v hydraulické doméně tlak, analogií elektrického proudu (i) je v hydraulické doméně proud kapaliny (a v pneumatické doméně proud plynu). Pro hydraulický odpor (R), stejně jako pro elektrický odpor platí Ohmův zákon (pouze místo rozdílu elektrického napětí je tlakový gradient a místo elektrického proudu je průtok). Hydraulickou analogií elektrického kondenzátoru je pružný vak, roztahovaný rozdílem tlaků uvnitř a vně vaku. Analogií elektrické kapacity kondenzátoru (C) je poddajnost stěny pružného vaku. Uvažujeme-li v hydraulickém systému ještě inerci, pak síla, která urychluje proudění kapaliny je tlakový gradient. Zrychlení proudu, tj. první derivace toku der(i), je proto podle Newtonova zákona úměrná tlakovému gradientu (v) a nepřímo úměrná hmotnosti zvoleného sloupce kapaliny tzv. inertanci (L). V elektrické doméně inertance odpovídá indukčnosti cívky. Každý prvek hydraulické či elektrické domény má dva propojovací konektory, kterými přitéká a odtéká elektrický nebo látkový proud (p.i, n,i), přičemž platí, že se protékající tok (i) nikde v prvku neztrácí (tj. i=p.i=n.i). Zároveň je na konektorech propojením do sítě přivedeno elektrické napětí či tlak (p.v, n.v), a v prvku se vytváří gradient elektrického napětí nebo gradient tlaků (v).

Fragment zápisu rovnic v třídě „Resistor“, popisující v Modelice vztah mezi proměnnými vyjadřujícími odpor (*R*), tlakový spád (*v*) a tok (*i*) je dle Ohmova zákona jednoduchý:

```
equation
  R*i = v;
end Resistor;
```

Třídou „Capacitor“ popisujeme pružný vak roztahovaný proudem vzduchu na jeho vstupu. Poddajnost (*C*) charakterizuje míru „roztažlivosti“ stěny vaku vlivem rozdílu tlaků (*v*) mezi tlakem vzduchu vhánějícím do vaku vzduch a tlakem vně pružného vaku. Pak rychlost toku vzduchu proudícího do vaku (*i*) popisuje v jazyku Modelica rovnice (kde *„der“* znamená derivaci):

```
equation
  i = C*der(v);
end Capacitor;
```

Inerční element v modelu realizujeme pomocí třídy „Inductor“. Síla, která urychluje proudění vzduchu je tlakový gradient. Zrychlení proudu, tj. první derivace toku der(i), je proto podle Newtonova zákona úměrná tlakovému gradientu (*v*) a nepřímo úměrná hmotnosti zvoleného sloupce plynu tzv. inertanci (*L*). V třídě „Inductor“ proto popisujeme vztah mezi změnou rychlosti toku (i) a tlakovým spádem (*v*) v závislosti na intertanci (*L*) pomocí jednoduché rovnice:

```
equation
  L*der(i) = v;
end Inductor;
```

Instance výše uvedených elementárních prvků se propojují do sítí pomocí konektorů – každý z těchto prvků má definovány dva propojovací konektory, označené jako „*p*“ a „*n*“. Na každý z nich je při zapojení přivedeno napětí, či v případě hydraulické či pneumatické domény – tlak (*p.v*, *n.v*), a zároveň těmito konektory může protékat elektrický proud, či látkový tok (*p.i*, *n.i*).

Konektory jsou instance speciálních konektorových tříd, kde se definují proměnné, používané pro propojení. Propojovat se mohou prvky pomocí konektorů, které jsou instancemi stejných konektorových tříd („propojovací zásuvky“ musí být stejného typu). V daném případě jsou konektory „*p*“ a „*n*“ instancemi konektorové třídy „*Pin*“, která umí s okolím propojovat napětí či tlaky (*p.v*, *n.v*) a toky (*p.i*, *n.i*). Hodnoty z těchto konektorů jsou propojeny s hodnotami proměnných (*i*) a (*v*) uvnitř jednotlivých elementárních prvků. Přitom platí, že ve všech výše uvedených elementárních prvcích se tok nikde neztrácí – co do prvku přitéká to z něj také odtéká (*i=p.i=n.i*), a z rozdílů napětí či tlaků se počítá příslušný gradient (*v=p.v-n.v*).

Implementace tohoto požadavku je jednoduchá – protože Modelica je objektový jazyk, všechny tři výše uvedené třídy elementárních prvků budou mít společného předka (OnePort), od kterého zdědí konektory „p“ a „n“ a zároveň i následující rovnice:

```
equation
  v= p.v-n.v;
 0= p.i-n.i;
 i= p.i;
end OnePort;
```

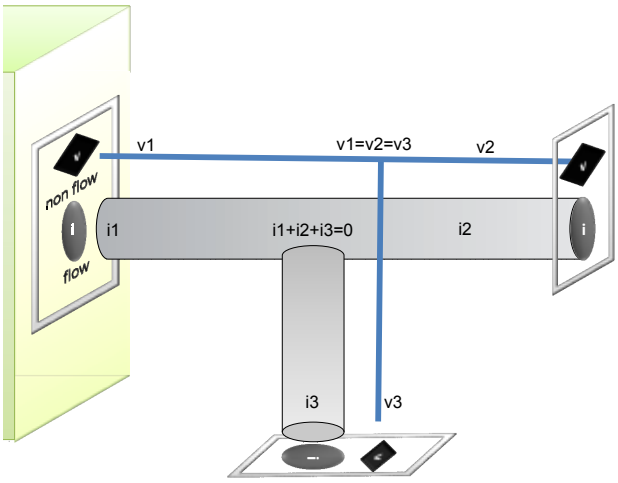
Rovnice tak spojí hodnoty tlaků či napětí přivedené z okolí na konektory „*p*“ a „*n*“ (*p.v*, *n.v*) s tlakovým či napěťovým gradientem (v) a vyjádří stejnou hodnotu (elektrického nebo hydraulického) toku na obou konektorech (*p.i*, *n.i*) i uvnitř prvku (*i*).

Konektorové třídy definují způsob komunikace jednotlivých modelicových komponent mezi sebou. Obrazně řečeno, definicí konektorových tříd definujeme typy „zásuvek“. V konektorech se definují jednotlivé proměnné, kterými bude konektor propojovat příslušnou komponentu s okolím.

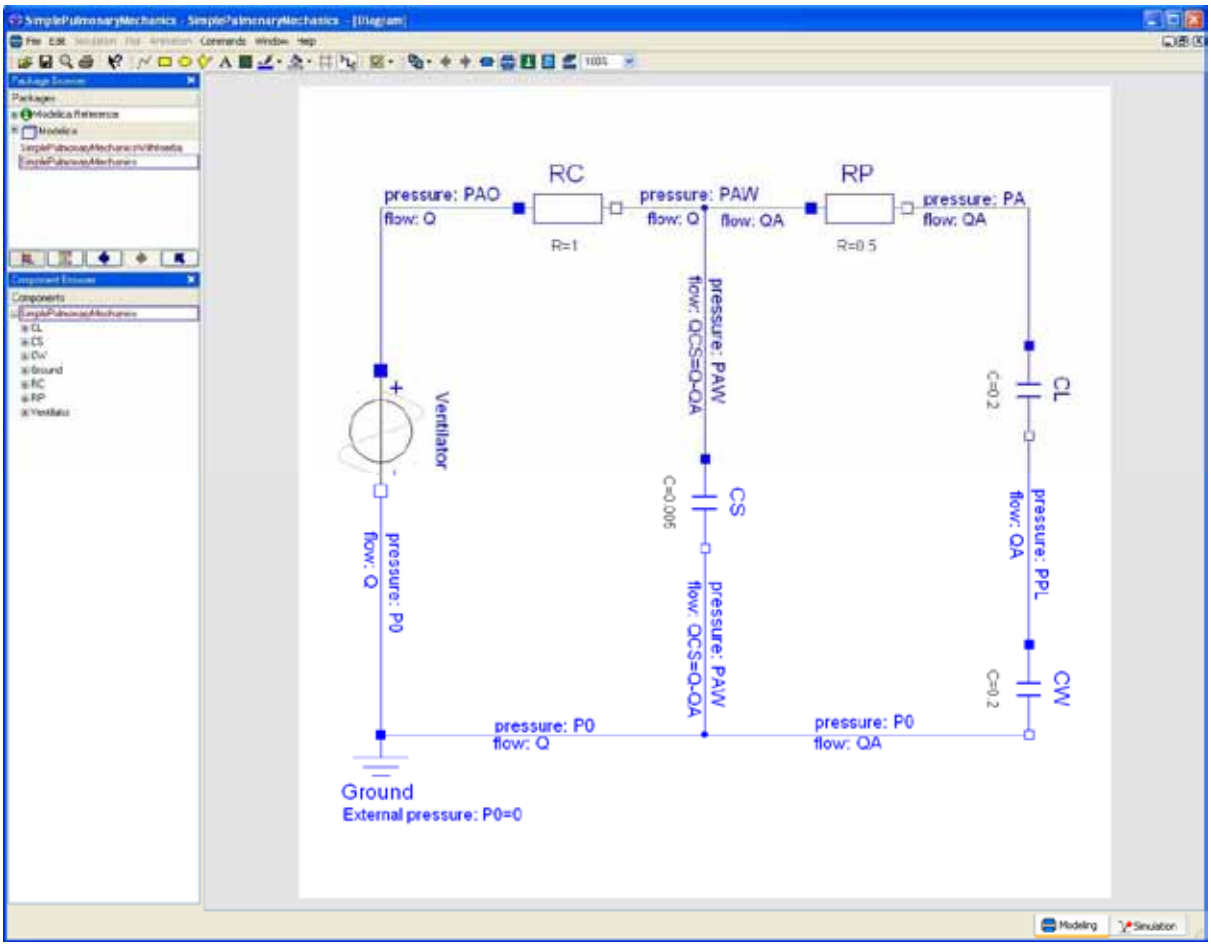
Pro každou proměnnou v konektoru se definuje, zda představuje nějaký tok – (proměnná je pak označena atributem „flow“), či nikoli (tzv. „non-flow“ proměnné). Toto rozlišení je důležité pro správnou

interpretaci propojení jednotlivých komponent (instancí tříd prvků) přes příslušné konektory (obr. 69). U tokových proměnných je zřejmé, že musí být zajištěno, aby se nikde v propojení příslušná entita (jejíž tok příslušná proměnná charakterizuje) neztrácela nebo neakumulovala. Proto součet hodnot všech propojených veličin s atributem „flow“ musí být nulový (jako podle Kirchhoffova zákona v elektrické doméně). U netokových proměnných propojení definuje, že jejich hodnoty u všech propojených konektorů musí být stejné (podle prvního Kirchoffovu zákonu). Propojením instancí jednotlivých elementárních prvků prostřednictvím konektorů se vyjádří požadavek nulovosti algebraického součtu hodnot propojených tokových proměnných a požadavek rovnosti hodnot propojených netokových proměnných.

Každá modelicová třída může mít svoji grafickou reprezentaci – ta je důležitá zejména pro zobrazení propojení instancí, kdy propojováním komponent se vytváří názorná grafická struktu-



Obr. 69 – Propojení modelicových komponent prostřednictvím akauzálních konektorů. Hodnoty konektorových proměnných typu flow (v daném případě proměnné i) budou nastaveny tak, aby algebraický součet hodnot všech propojených toků byl nulový. Hodnoty ostatních (non-flow) proměnných (v daném příkladě hodnoty v) budou na všech propojených konektorech nastaveny na stejnou hodnotu.



Obr. 70 – Implementace modelu plicní mechaniky (podle obr. 59) v Modelice mnohem více připomíná strukturu modelované reality než implementace v Simulinku.

ra modelu. Proto pro každou třídu v Modelice můžeme definovat i příslušnou ikonu. Tato ikona může být i animovaná.

Model pak v Modelice můžeme vytvářet graficky propojováním instancí jednotlivých prvků, které pomocí myši vybíráme z knihovny a v dialogu pak nastavíme hodnoty příslušných parametrů.

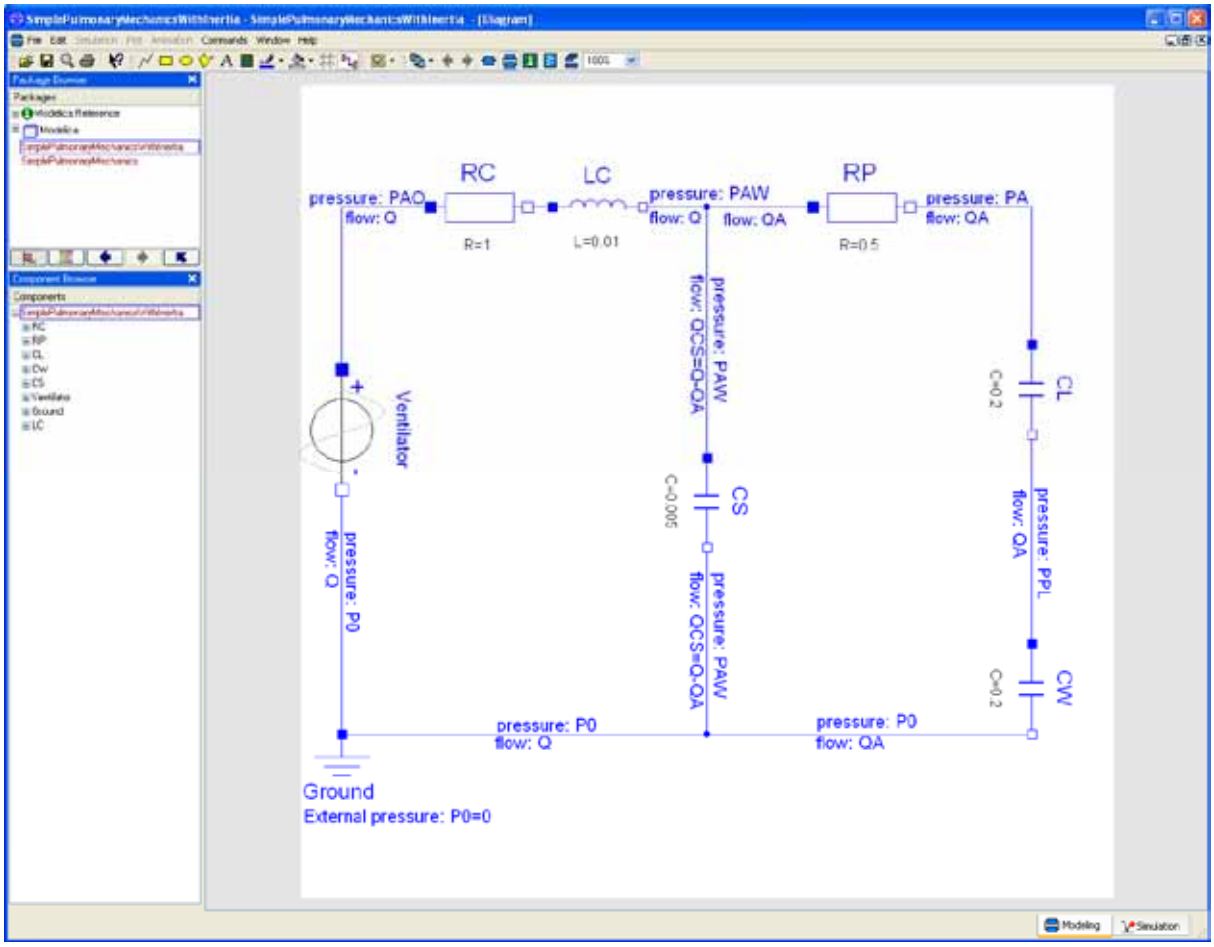
V případě implementace modelu plicní ventilace potřebujeme propojit instance prvků „Resistor“, „Capacitor“ a „Inductor“.

Základní prvky, které potřebujeme, ale nemusíme programovat od samého počátku – Modelica obsahuje bohaté knihovny z různých fyzikálních domén (elektrické, hydraulické, mechanické aj.), kde tyto prvky lze již najít.

V našem konkrétním případě můžeme pro rychlé řešení využít např. vizuální komponenty elektrických obvodů – vytvoříme jednotlivé instance (*RC*, *RP*, *CL*, *CW* a *CS*), zadáme příslušné hodnoty parametrů (*C* a *R*) a pomocí konektoru komponenty propojíme.

Výsledek uvádí obr. 70. Porovnáme-li strukturu modelu implementovanou v Modelice s původním schematickým obrázkem zobrazujícím strukturu modelu (obr. 63), vidíme, že v Modelice je řešení přímočaré a (na rozdíl od implementace v Simulinku – obr. 64 a obr. 64) struktura modelu odpovídá struktuře modelované reality.

Zesložitění modelu přidáním inerčního elementu nepřináší žádné zvláštní těžkosti – stačí pouze z knihovny pomocí myši vytáhnout příslušnou inerční komponentu (*LC*), nastavit hodnotu jejího parametru (*L*) a propojit do modelu. Struktura implementovaného modelu v Modelice, uvedená na obr. 71, odpovídá struktuře modelované reality (obr. 66), zatímco struktura Simulinkové implementace (obr. 67) odpovídá spíše způsobu řešení rovnic modelu.



Obr. 71 – Pro implementaci modelu plicní mechaniky (podle obr. 62) s uvažováním inerčního členu v Modelice stačí přidat inerční komponentu LC.

Elementární prvky simulované reality mohou mít velmi triviální zápis vztahů mezi danými veličinami. Odpor, kondenzátor či cívka z elektrické fyzikální domény či jejich hydraulické analogie jsou toho názorným příkladem.

Složitý systém pro výpočet vznikne, když tyto elementární prvky začneme propojovat do sítí – při jejich vzájemném propojování vznikají soustavy rovnic. Jejich numerické řešením v kauzálních simulačních nástrojích zdaleka nemusí být triviální – tak např. složitější R-C-L modely cirkulace či respirace implementované v Simulinku jsou velmi složité (např. modely cirkulace v naší simulinkové knihovně Physiolibrary – http://www.physiome.cz/simchips)

V Modelice se o způsob řešení rovnic nemusíme starat. Pozornost je nutno věnovat spíše definici rovnic v jednotlivých prvcích a propojování jejich instancí (jednotlivých komponent).

V Modelice se o algoritmus řešení vzniklé soustavy rovnic se postará samotný akauzální nástroj, a po spuštění simulace můžeme na různých místech simulovaného obvodu sledovat příslušné toky a tlaky.

4.2.6 Kauzální a akauzální konektory

Akauzální konektorové spojení komponent se realizuje pomocí dvou typů veličin: jedné, která představuje tok – pro něj platí, že součet hodnot toků je na všech připojených uzlech nulový (protože v oblasti rozvětvení do připojených uzlů se žádná látka neakumuluje), a druhé, jejíž hodnota zůstává na všech připojených uzlech stejná. Je vhodné, aby každá proměnná s atributem flow byla v konektorovém propojení provázena non-flow proměnnou reprezentující zobecnělé úsilí ve vztahu k flow proměnné.

Na rozdíl od Simulinkových komponent (které mají definovány vstupy do komponenty a výstupy z komponenty) se v akauzálním propojení se nedefinuje co je vstup a co je výstup. Akauzální modelicová komponenta nepočítá ze vstupních hodnot výstupní hodnoty. Propojení modelicových komponent akauzálními konektory propojuje rovnice v jednotlivých komponentách do soustav rovnic.

Modelicové třídy mohou, kromě akauzálních vazebních konektorů, obsahovat i kauzální vstupní konektory, kterými se z okolí přivádějí skutečné vstupní veličiny, stejně jako kauzální výstupní konektory, kterými se do okolí vysílají výstupní veličiny.

Kromě rovnic, modelicové třídy mohou také obsahovat přesně definovaný algoritmus výpočtu výstupních hodnot ze vstupních (typickým příkladem je modelování funkčních závislostí).

Modelicové komponenty jsou tedy propojovány jak akauzálními vazbami, tak i kauzálními směrovanými vstupy a výstupy.

Kauzální konektory obvykle rozvádějí signály – například v modelu krevního oběhu mohou signálové kauzální vstupy obsahovat signály, kterými se nastavují hodnoty odporů v komponentách reprezentujících rezistenci cirkulačního řečiště.

Model v Modelice se tak obvykle reprezentuje grafickou soustavou komponent, propojených jak akauzálními, tak i kauzálními vazbami. Komponenty jsou instance modelicových tříd, jejichž struktura může být také reprezentován sítí propojených instancí.

4.2.7 Příklad definice a využití akauzálního prvku – elastický kompartment

Ukažme si jednoduchý příklad definice a využití modelicové třídy.

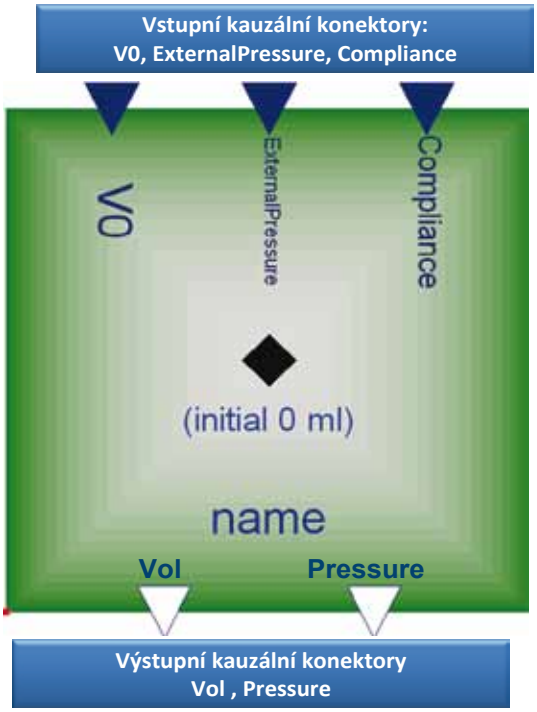
V modelování fyziologických systémů a ve farmakokinetice se často používá tzv. kompartmentový přístup, kdy se studuje dynamika pohybu látek mezi jednotlivými „kompartmenty“, kterými se označují částečně nebo úplně oddělené součásti celku které mají určité specifické vlastnosti (Sheppard 1948).

Při modelování dynamiky cév často potřebujeme elastický (nafukovací) kompartment.

Nadefinujeme si proto třídu *VascularElacticBloodCompartment,* jejíž instance budou elastické akauzálně propojitelné kompartmenty, které bude možno přes akauzální konektor připojit na „rozvod“ tekutiny – tekutina může do/z kompartmentu proudit určitou rychlostí a pod určitým tlakem. V programovacím prostředí můžeme každé třídě, reprezentující model nebo konektor přiřadit grafickou ikonu.

I pro náš vytvářený elastický kompartment můžeme vytvořit ikonu (obr. 72).

Není to jen čistě školní případ – tento kompartment využíváme v naší Modelicové implementaci rozsáhlého modelu fyziologických funkcí „Quantitative Human Physiology“ (Abram, Hodnett, Summers, Coleman & Hester, 2007; Coleman, Hester & Summers, 2009). Na obr. 77 je zobrazen příklad využití instancí elastického kompartmentu v naší implementaci tohoto rozsáhlého modelu.



Obr. 72 – Modelica umožňuje vytvořit pro každou vytvářenou třídu, reprezentující model nebo konektor, vytvořit ikonu, která poslouží k tomu, aby instance třídy bylo možno pomocí grafických nástrojů propojovat s jinými instancemi. Výsledkem je struktura modelu z propojených instancí, velmi blízká modelované realitě. V daném případě jsme pro vytvoření ikony elastického kompartmentu vytvořili ikonu s jedním akauzálním konektorem (černý kosočtverec), třemi konektory pro signálové vstupy a dvěma konektory pro signálové výstupy. Každá instance elastického kompartmentu bude mít tuto ikonu s tím, že se v ikoně bude místo řetězce „initialVol“ se bude zobrazovat skutečná hodnota počátečního objemu (zadaná jako parametr) a místo řetězce „name“ se bude zobrazovat název instance.

Elastický cévní kompartment (obr. 73) si můžeme představit jako nafukovací vak s jedním ***akauzálním propojovacím konektorem*** (nazveme jej třeba *„ReferencePoint“*), přes který se budeme propojovat na okolí. Tento konektor nám zprostředkuje dvě veličiny:

- tok *„ReferencePoint.q“;*
- tlak *„ReferencePoint.pressure“*.

Pokud bude konektor zapojen přes konektor do okolí, pak hodnota tlaku bude na všech ke kompartmentu připojených uzlech skutečně stejná, a tok se bude rozdělovat do všech připojených uzlů tak, že jeho algebraický součet bude nulový (v oblasti rozvětvení se nebude vůbec nic akumulovat) – příklad zapojení komponenty na obr. 74.

Do kompartmentu budou vstupovat z vnějšku tři signálové (kauzální) vstupy:

- základní náplň *„V0“* – hodnota objemu, po jehož dosažení bude v elastickém kompartmentu stoupat tlak; pokud bude objem menší než nula, bude tlak v kompartmentu nulový;
- vnější, externí tlak *„ExternalPressure“* – tlak vnějšího okolí na elastický kompartment;
- poddajnost elastického kompartmentu *„Compliance“* – té bude nepřímo úměrný tlak v kompartmentu pokud objem kompartmentu překročí základní náplň.

Z kompartmentu budou navenek vystupovat dva (kauzální) signálové výstupy:

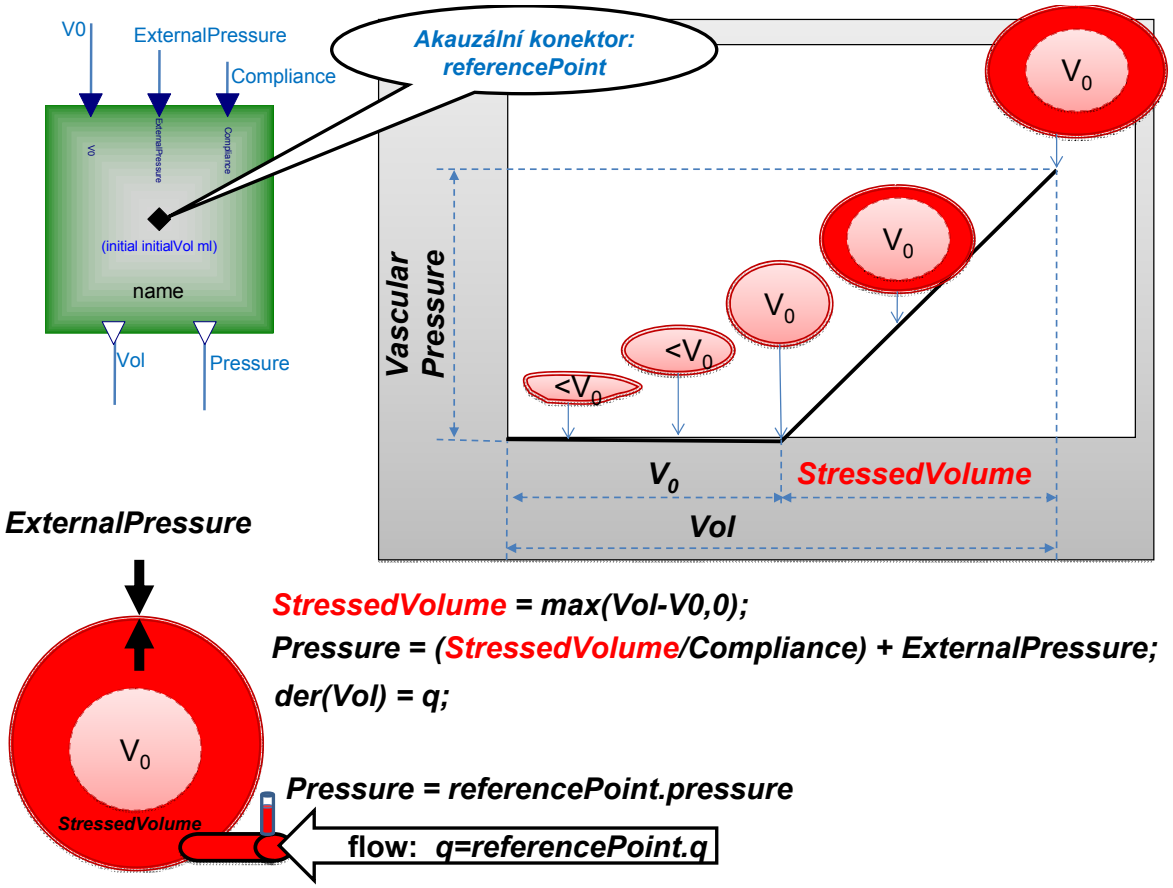
- informace o momentálním objemu kompartmentu *„Vol“*;
- informace o hodnotě tlaku uvnitř kompartmentu *„Pressure“*.

Pro kompartment je výhodné ještě navrhnout parametr (jehož hodnota se načte před začátkem simulace), kterým by se určila jeho počáteční náplň:

- počáteční objem kompartmentu *„initialVol“.*

V programovém prostředí můžeme navrhnout i ikonu pro zobrazení elastického kompartmentu.

Vlastní fragment programu popisující chování elastického kompartmentu vypadá v Modelice následovně:

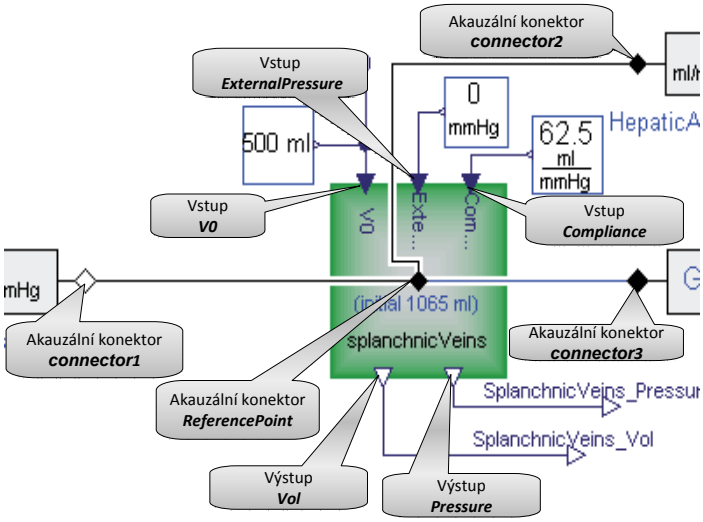


Obr. 73 – Koncepce eleastického cévního kompartmentu je založena na představě, že plní-li se céva krví, do dosažení určitého reziduálního objemu (V0) je tlak v cévě určován pouze vnějším tlakem na cévu, poté se začnou napínat elastická a svalová vlákna v cévě a budou komprimovat objem krve v cévě tlakem VascularPressure. Označíme-li objem tekutiny v cévě jako Vol, pak objem krve napínající cévu (StressedVolume) bude určovat tlak uvnitř cévy (Pressure) v závislosti na její poddajnosti (Compliance) a na vnějším tlaku na cévu (ExternalPressure). Cévní kompartment je zapojen do systému přes konektor ReferencePoint, kterým může do kompartmentu proudit krev (rychlostí referencePoint.q) pod tlakem (referencePoint.pressure).

```
model VascularElacticBloodCompartment extends QHP.Library.Interfaces.BaseModel;

Real StressedVolume ( final quantity="Volume", final unit="ml");
parameter Real initialVol(  final quantity="Volume", final unit="ml")
                                  „initial compartment blood volume";
...
initial equation
        Vol = initialVol;
equation
        der(Vol) =  referencePoint.q;
        StressedVolume = max(Vol-V0,0);
        Pressure = (StressedVolume/Compliance) + ExternalPressure;
        referencePoint.pressure = Pressure;
end VascularElacticBloodCompartment;
```

První řádek deklaruje třídu modelu, dále se deklaruje reálná proměnná *„StressedVolume"*, u níž budou kontrolovány i fyzikální jednotky. Není to jen otázka přehlednosti a čitelnosti programu. Kontro-



Obr. 74 – Instance „splanchnicVeins“ elastického kompartmentu „VascularElacticBloodCompartment“. Akauzální spojení s příslušnými konektory na regulovatelných odporech (zde označených jako „connector1“, „connector2“ a „connector3“) propojí rovnice v instanci elastického kompartmentu „splanchnicVeins“ do soustavy rovnic všech propojených elementů. Hodnota tlaku bude stejná na všech propojených konektorech: splanchnicVeins.ReferencePoint.pressure = connector1.pressure = connector2.pressure = connector3.pressure. Algebraický součet všech toků na propojených konektorech musí být nulový: splanchnicVeins.ReferencePoint.q + connector1.q + connector2.q + connector3.q = 0.

la kompatibility jednotek nám umožní se vyhnout velmi špatně hledatelné chybě, kdy omylem v propojeních prohodíme konektory (pokud se zjistí, že jednotky jsou inkompatibilní, kontrola nám vytvoření špatného propojení vůbec nedovolí).

Dále se deklaruje parametr *„InitialVol“* u něhož také budou kontrolovány fyzikální jednotky. A pak následuje sekce rovnic. Nejprve se deklaruje inicializace počátečního objemu kompartmentu, tj. proměnné *„Vol“*. Na dalších řádcích se v sekci equation deklarují čtyři rovnice. První je diferenciální rovnice – derivace objemu *„der(Vol)“* se rovná přítoku *„q“* z konektoru *„referencePoint“.*

Další rovnice deklaruje, že hodnota elasticky napínaného objemu *„StressedVolume“* bude počítána jako rozdíl mezi objemem kompartmentu *„Vol“* a hodnotou jeho základní náplně *„V0“* (která je vstupem), a dále rovnice říká, že hodnota objemu kompartmentu nikdy nemůže poklesnout k záporným hodnotám.

Třetí rovnice deklaruje vztah mezi tlakem v kompartmentu *„Pressure“*, hodnotou napínaného objemu *„StressedVolume“,* poddajností *„Compliance“* a externím tlakem *„ExternalPressure“*. Ještě jednou připomínáme, že se zde jedná o rovnice a nikoli o přiřazení. Rovnici by bylo možno v Modelice napsat i takto:

Pressure - ExternalPressure= (StressedVolume/Compliance);

Poslední rovnice propojuje hodnotu tlaku v kompartmentu *„Pressure“* s hodnotou tlaku propojovanou *„referencePoint.pressure“* akauzálním konektorem s okolím.

Hodnota „*Pressure*“, je zároveň signálovým výstupem z kompartmentu – jako signál ji můžeme přivádět k dalším blokům – je to ale kauzální výstupní (signálová) proměnná a její hodnota nemůže být ovlivněna tím, k čemu ji připojíme. U propojení z akauzálního konektoru je to ale jiné. Když instanci elastického kompartmentu propojíme akauzálním konektorem s dalšími prvky, pak čtveřice rovnic v kompartmentu se stane součástí soustavy rovnic daných příslušným propojením a hodnoty proměnných v instanci elastického kompartmentu budou záviset na řešení takto vzniklé soustavy rovnic.

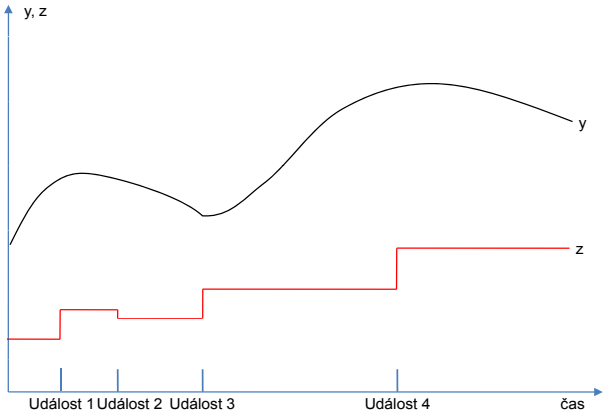
4.2.8 Hybridní modely

Pro matematický popis modelů reálného světa často vystačíme se spojitou dynamikou vyjádřenou soustavou algebro-diferenciálních rovnic. Přesto však nezřídka potřebujeme vyjádřit nespojité, diskrétní chování (které je často aproximací rychlých spojitých dějů fyzikálních systémů) a samotné spojité dynamické systémy (continuous dynamic systems) nám pro popis dějů reálného světa nestačí. Příkladem je třeba otevírání a uzavírání ventilů v hydraulické a pneumatické doméně, chování diod v elektrické doméně nebo zapínání a vypínání genů, vznik a přenos nervových vzruchů či otevírání či uzavírání iontových kanálků v biologické doméně. Diskrétní systémy řízené událostmi (discrete event dynamic systems) jsou časté při popisu technických aplikací. Velmi mocným nástrojem pro formalizovaný popis procesů a jejich interakcí jsou diskrétní hierarchické stavové automaty (Harel, 1987; Harel, Kugler & Pnueli, 2005; Sadot, Fischer, Admanit, Stern, Hubbard & Harel, 2008).

Často je při modelování rozsáhlejších systémů vhodné v menší či větší míře kombinovat diskrétní i spojitý popis. Tyto, tzv. hybridní modely mohou kombinovat proměnné diskrétního i spojitého času, mohou generovat a reagovat na nejrůznější události (obr. 75).

Hybridní modely podporují moderní vývojové simulační prostředí. Tak např. model spojitého dynamického systému v Simulinku je možné kombinovat s hierarchickými stavovými automaty vytvořenými ve speciálním modelovacím nástroji Stateflow – hodnoty proměnných v Simulinku mohou měnit stavy automatů v Stateflow, a Stateflow pomocí generovaných událostí může přepínat výpočetní bloky v Simulinku a tím měnit postup výpočtu.

Akauzální vývojové nástroje však mohou na základě generovaných událostí měnit přímo použité rovnice (nikoli jen způsob jejich řešení). Jako malou ilustraci si můžeme uvést příklad modelování průměrného objemu krve v srdeční komoře (obr. 76).



Obr. 75 – Příklad chování reálných proměnných v hybridních systémech. Proměnná spojitého času „y“ se mění v čase (její hodnota nemusí být spojitá – jako reakce na určitou událost se např. může nespojitě, třeba skokově změnit). Reálná proměnná diskrétního času „z“ mění své hodnoty pouze v okamžicích určité události.

Srdeční komora je modelována jako kontinuální čerpadlo s proměnlivým vnitřním objemem.

Akauzálními konektory *„q_in“* a *„q_out“* je model srdeční komory zapojen do oběhu. Těmito konektory je propojen tok krve do (*„q_in.q“*) a z (*„q_out.q“*) komory. Změna objemu krve v komoře bude určena algebraickou sumou toků v obou akauzálních konektorech. V Modelice to zapíšeme takto:

der(Volume)=q_in.q+q_out.q

V modelu jsou dva kauzální vstupy – prvním je momentální průtok komorou (*„BloodFlow“*) a druhým je požadovaný objem krve v komoře v ustáleném stavu (*„VentricleSteadyStateVolume“*). Pokud je tento objem vyšší, než je momentální objem komory (*„Volume“*), pak přítok do komory se nastaví na větší hodnotu než odtok, a to úměrně rozdílu mezi požadovanou hodnotou a skutečnou hodnotou:

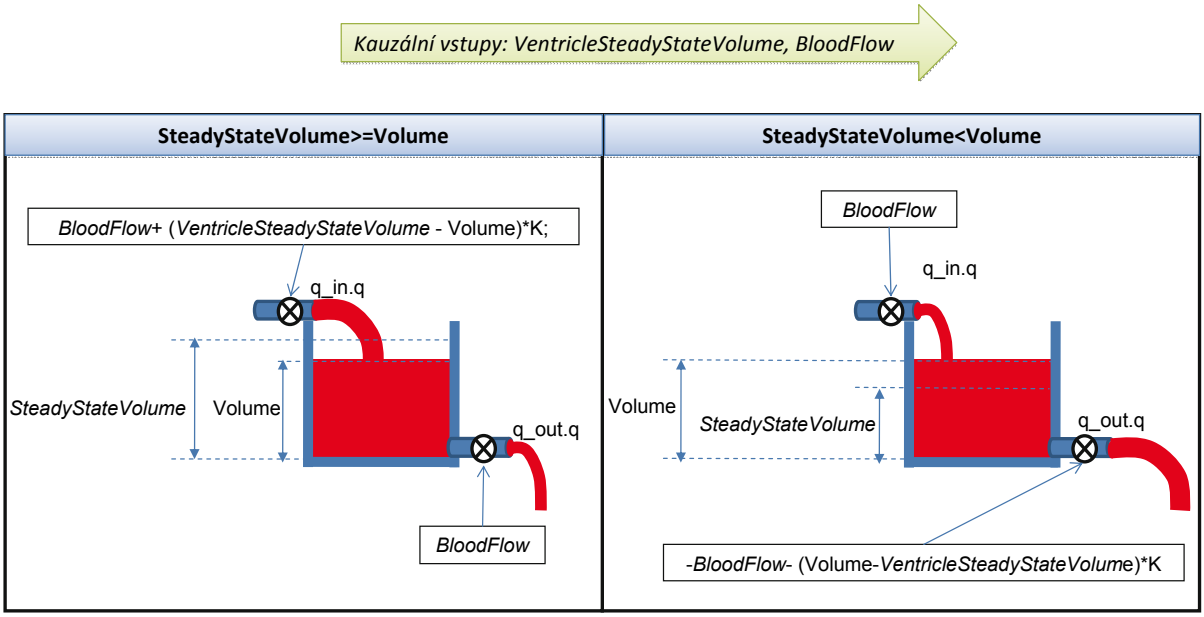
*q_in.q = BloodFlow + (VentricleSteadyStateVolume – Volume)*K;*

Odtok z komory (*„q_out.q“*) bude nastaven na hodnotu „*BloodFlow*“ se záporným znaménkem, protože se jedná o odtok z kompartmentu:

q_out.q=-BloodFlow;

V opačném případě, kdy požadovaná hodnota objemu krve v komoře (*„VentrickeStaedyStateVolume*“) bude nižší než skutečná hodnota (*„Volume“*), bude přítok krve nastaven na hodnotu *„BloodFlow“* a odtok krve se nastaví na větší hodnotu než přítok a to úměrně rozdílu mezi skutečnou hodnotou a požadovanou hodnotou. Fragment zápisu rovnic v Modelice pak vypadá následovně:

model *VentricleVolumeAndPumping;*
….
equation
der(Volume)=q_in.q+q_out.q
    if *(SteadyStateVolume >=Volume) then*




```
der(Volume)=q_in.q+q_out.q
if (SteadyStateVolume >=Volume) then
    q_in.q=BloodFlow+ (VentricleSteadyStateVolume - Volume)*K;
    q_out.q=-BloodFlow;
else
    q_in.q=BloodFlow;
    q_out.q=-(BloodFlow+ (Volume-VentricleSteadyStateVolume)*K);
end if;
```

Obr. 76 – Akauzální modelovací nástroj Modelica umožňuje dynamicky měnit používanou sadu rovnic. Na obrázku je hydraulická analogie struktury modelu objemu srdeční komory vyjádřeným jako kontinuální čerpadlo s proměnlivým vnitřním objemem. Dva akauzální konektory (q_in) a (q_out) propojují komponentu s okolím, komponenta dostává jako kauzální vstupy hodnotu průtoku krve (BloodFlow) a požadovanou hodnotu objemu krve v komoře (VentricleStaedyStateVolume). Rovnice počítají objem krve v komoře (Volume) a hodnoty přítoku a odtoku krve (q_in.q, q_out.q). Použité rovnice se liší v závislosti na tom, zda je či není požadovaný objem krve větší než momentální objem krve v komoře (tj. zda SteadyStateVolume >=Volume)..

```
        q_in.q=BloodFlow+ (VentricleSteadyStateVolume - Volume)*K;
        q_out.q=-BloodFlow;
    else
        q_in.q=BloodFlow;
        q_out.q=-(BloodFlow+ (Volume-VentricleSteadyStateVolume)*K);
    end if;
end VentricleVolumeAndPumping;
```

V závislosti na hodnotě proměnných „*Volume*“ a „*VentricleSteadyStateVolume*“ se pak v soustavě rovnic modelu přepínají dvě rovnice. Na první pohled zápis vypadá, že jde o přiřazení (jako v běžných programovacích jazycích), jde však o rovnice. Ekvivalentní zápis může totiž vypadat i takto:

```
model VentricleVolumeAndPumping;
....
equation
    delta = (VentricleSteadyStateVolume - Volume)*K;
```

```
        der(Volume) = delta;
        q_in.q + q_out.q = delta;
        if (delta<0) then
            q_in.q=BloodFlow;
        else
            q_in.q=BloodFlow+delta;
        end if;
end VentricleVolumeAndPumping;
```

Protože jde o rovnice, nezáleží na jejich pořadí, ani na tom zda hodnota proměnné *„delta“* v třetí rovnici je vpravo či vlevo.

Skutečný zápis použitých rovnic v Modelice ke ještě úspornější:

```
model VentricleVolumeAndPumping;
....
equation
        delta = (VentricleSteadyStateVolume - Volume)*K;
        der(Volume) = delta;
        q_in.q + q_out.q = delta;
        q_in.q = if (delta<0) then BloodFlow else BloodFlow+delta;
end VentricleVolumeAndPumping;
```

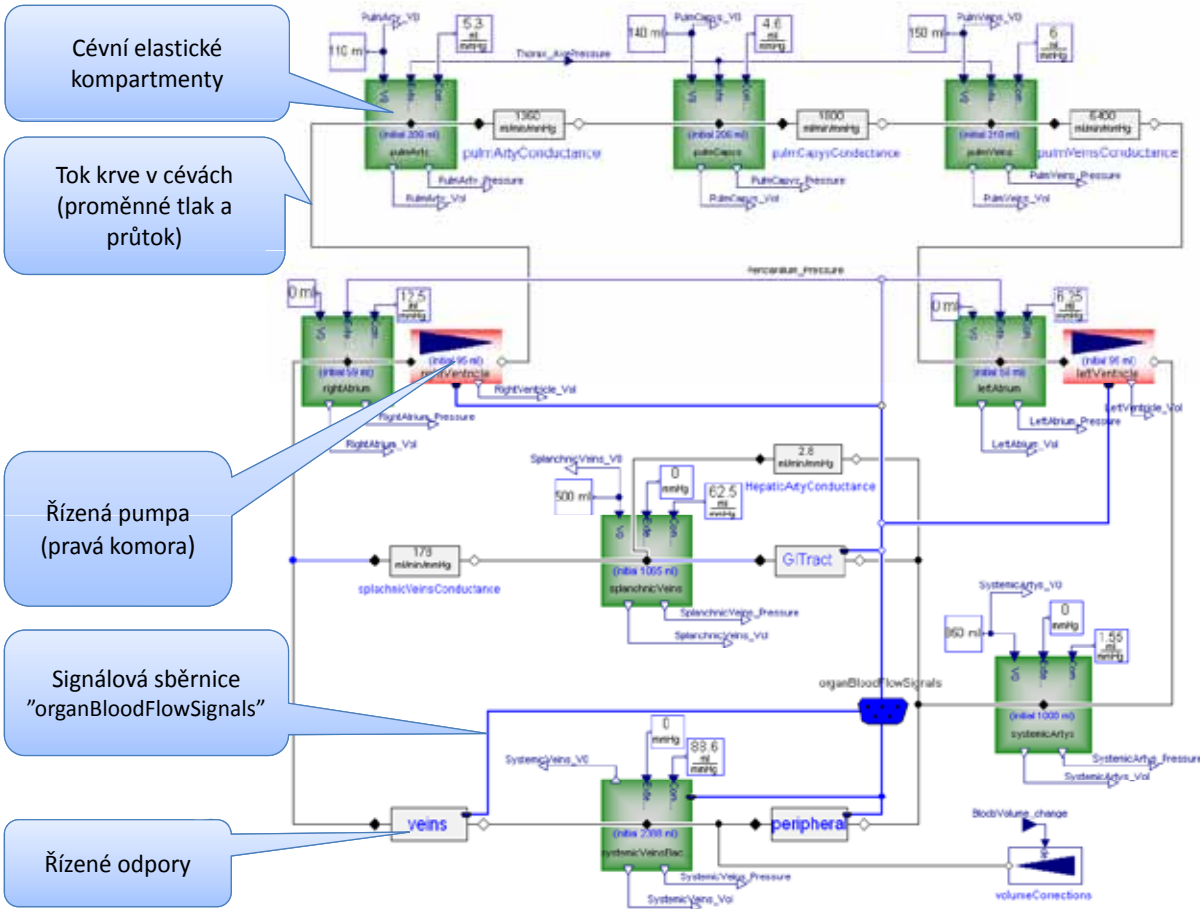
Modelica umožňuje akauzálně popisovat diskrétní i spojité systémy a zároveň dává i velké možnosti pro kombinaci modelů s diskrétní a spojitou částí. Podrobnosti je možno najít v (Fritzon, 2003).

4.2.9 Kombinace akauzálních a kauzálních (signálových) vazeb v hierarchicky uspořádaných modelech

Modelica usnadňuje a zpřehledňuje modelování rozsáhlých systémů a podporuje jejich hierarchickou dekompozici.

Objektová architektura Modeliky podporuje členění modelu na vhodné části, které mají ucelený význam, aby je bylo možné zkoumat samostatně v určitých podmínkách nebo je znovu použít (ať již na jiném místě stejného modelu nebo v jiném modelu) a tím podstatně zvyšuje přehlednost vytvářených modelů. Proto v Modelice vytváříme rozsáhlé znovupoužitelné knihovny modelicových „simulačních čipů“ a každý model je obvykle provázen rozsáhlou hierarchicky uspořádanou knihovnou jednotlivých prvků. Hierarchické komponenty se dají „rozkliknout“ a odkrýt tak jejich vnitřní uspořádání.

Příkladem hierarchické struktury modelicového programu je třída *„VascularCompartments“* (obr. 77) která implementuje část subsystému krevního oběhu a využívá instance výše uvedené třídy *„VascularElacticBloodCompartment“*. Krev proudí akauzálními konektory mezi instancemi elastických kompartmentů, odpory jednotlivých částí cévního řečiště a dvěma čerpadly modelujícími činnost pravé a levé komory srdce. V komponentě se využívají i akuzální signálové vazby. Celá sada signálových vazeb (přicházejících z vnějšku komponenty) je např. rozváděna sběrnicí „OrganBloodFlowSignals“. Vstupní signálové vazby řídí hodnotu periferních odporů a čerpací funkce pravé a levé komory. Struktura modelu mnohem lépe a přehledněji vyjadřuje strukturu modelované reality než modely vytvořené v kauzálních modelovacích prostředích. Stačí porovnat model z obr. 77 v Modelice s centrální částí klasického Guytonova modelu z roku 1972 implementovaného v Simulinku (obr. 6 na straně 8). Oba modely vyjadřují zhruba totéž – průtok pružným cévním řečištěm a srdeční pumpu (modelicový model je ovšem podrobnější). Simulinkový model daleko více vyjadřuje postup



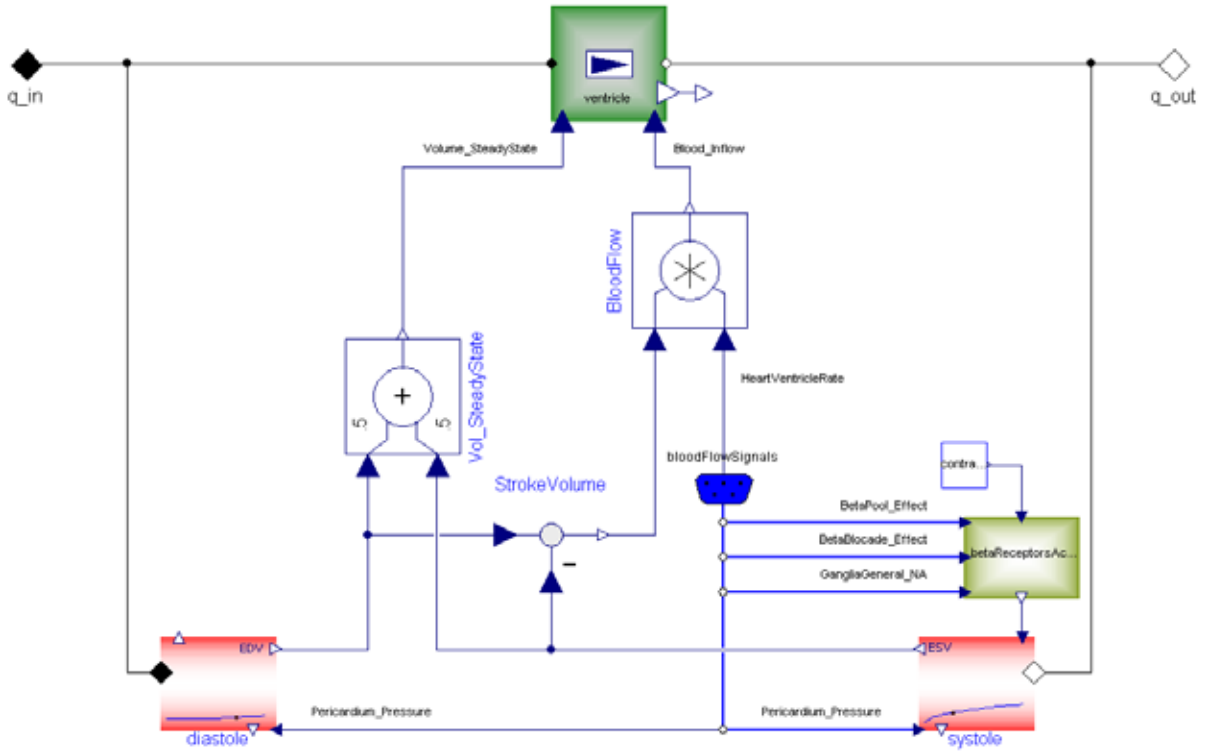
Obr. 77 – Ukázka části modelu subsystému krevního oběhu – instance třídy „VascularCompartments" v Modelice (část Modelicové implementace rozsáhlého modelu Quantitative Human Physiology). V modelu se kombinují akauzální a kauzální (řídící, signálové) vazby. Propojení akauzálními vazbami v daném případě modeluje rozvod průtoků a tlaků krve mezi jednotlivými propojenými komponentammi. Model je hierarchicky organizován jednotlivé bloky se dají „rozkliknout", představují instance tříd, v nichž jsou uvedeny rovnice. Modelicová síť tak mnohem lépe vyjadřuje strukturu modelovaného systému, než sítě v kauzálních modelovacích nástrojích, které zobrazují spíše postup výpočtu.

výpočtu, než strukturu modelovaného systému.

Rozdíl je vidět i na příkladu využití komponent RLC prvků z naší knihovny *Physiolibrary* pro modelování krevního oběhu na obr. 56. Tyto komponenty obsahují po zapojení do modelu průběžně počítající tlaky, průtoky a objemy v jednotlivých částech krevního řečiště včetně modelování čerpací funkce komor. Každá komponenta má ovšem jasně definovaný tlak a průtok na vstupu a tlak a průtok na výstupu. Propojení prvků pak definuje postup výpočtu, nikoli strukturu matematických vztahů, které jsou na pozadí těchto komponent. V Modelice propojujeme rovnice v blokově orientovaném Simulinku jsme propojovali bloky, které ze vstupních dat počítají podle zadaného algoritmu výstupní data. Důležitost akauzálního přístupu si ovšem uvědomila i firma Mathworks, která nabízí speciální simulinkovou knihovnu Simscape, v níž se dají propojovat akauzální komponety (Mathwoorks, 2009).

Výhoda akauzálních modelovacích nástrojů je zvláště patrná u složitějších modelů, kde je klíčem k úspěchu možnost hierarchické dekompozice modelu, kdy je důležité, aby propojení komponent vždy agregovaně vyjadřovalo nejpodstatnější vztahy na dané hierarchické úrovni a pro podrobnosti je zapotřebí se pohroužit do struktury jednotlivých komponent, které odkryjí opět agregovanou strukturu modelované reality na nižší hierarchické úrovni.

Tak např. komponenta zobrazující čerpadlo pravé komory je dvěma kauzálními konektory (rozvádějícími průtok a tlak krve) propojena s elastickým kompartmentem pravé síně a elastickým kom-



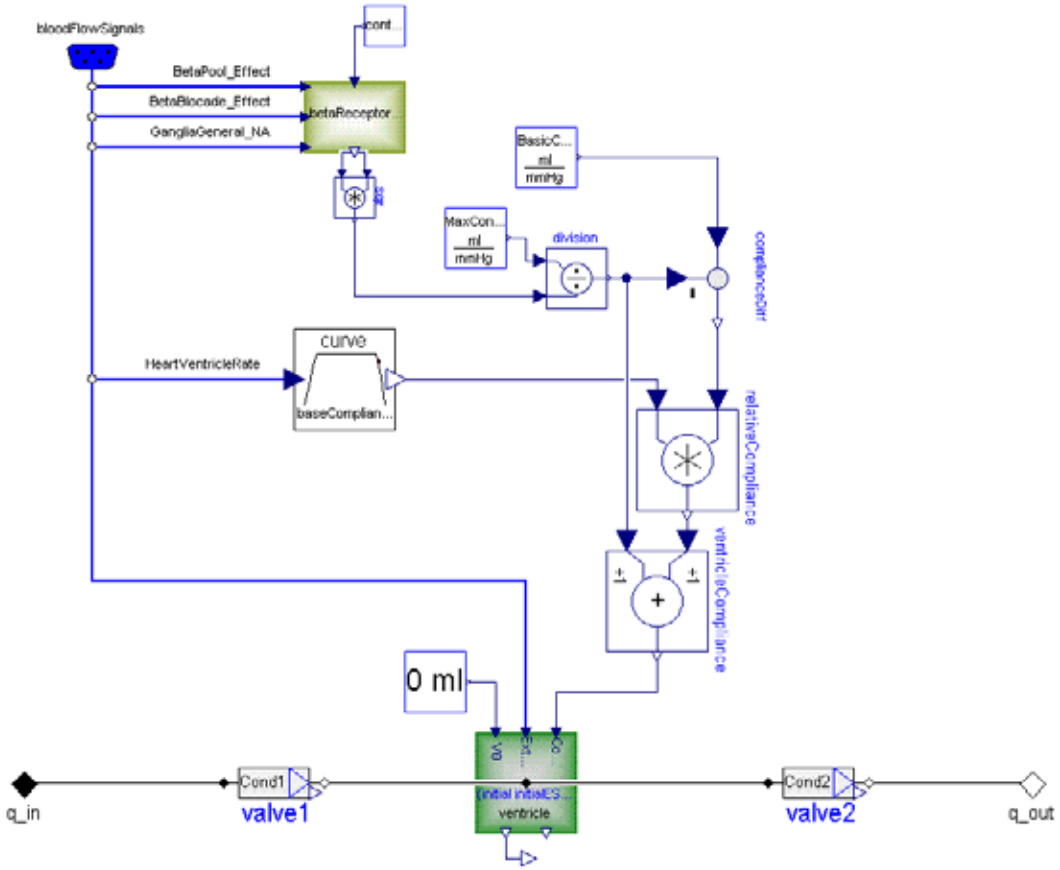
Obr. 78 – „Vnitřek“ instance pumpy pravého komory (komponenta „rightVentricle“) z předchozího obr. 77. Komora je modelována jako kontinuální čerpadlo s proměnným vnitřním objemem

partmentem plicních arterií. Ze sběrnice „*organBloodFlowSignals*“ jsou k ní připojeny kauzální signálové řídící vstupy. „Vnitřek“ komponenty je zobrazen na obr. 78.

Srdce je pulzační pumpa, která do srdečních komor nejprve v tzv. „diastole“ nasaje ze srdečních síní krev – na konci diastoly se bude objem krve v komoře rovnat tzv. koncovému diastolickému objemu (*EDV*). Po ukončení diastoly se uzavřou chlopně mezi síní a komorou, a komora se začne v tzv. „systole“ smršťovat. Otevřou se příslušné chlopně a pravá komora začne vypuzovat krev do plicní artérie (levá komora do aorty). Na konci systoly se v pravé komoře uzavřou chlopně mezi komorou a plicní artérií (a v levé komoře mezi komorou a aortou) – objem krve v komoře na konci systoly se nazývá koncový systolický objem (*ESV*). Svalovina komory ochabne, tlakový gradient mezi síní a komorou otevře síňo-komorové chlopně a opět začne diastola.

V modelu srdeční komory se v komponentě „diastole“ počítá koncový diastolický objem (*EDV*) a v komponentě „systole“ koncový systolický objem (*ESV*). Modelica umožňuje nejen navrhnout grafický tvar ikonek, zobrazujících jednotlivé komponenty, ale také i ikonky (pro zvýšení názornosti) animovat. V daném příkladě jsou v obou komponentách během simulace animovány křivky, které vyjadřují vztah mezi tlakem v komoře a hodnotami *ESV* a *EDV*. Tečkou na křivkách je znázorněna momentální hodnota *EDV* resp. *ESV*. Tlak krve v komoře při diastole je odvozen od hodnoty plnícího tlaku v síni (ten se do komponenty dostává akauzálním konektorem z „*q_in*“) a hodnoty vnějšího tlaku v perikardu – ten se do komory dostává ze signálové sběrnice „*BloodFlowSignals*“ kauzálním konektorem „*Pericardium_Pressure*“. V komponentě „*systole*“ je tlak krve na konci systoly v pravé síní odvozen od hodnoty protitlaku v plicní arterii (nebo tlaku v aortě v levé komoře) – přes akauzální konektor „*q_out*“, a hodnotu vnějšího tlaku v perikardu (přes kauzální konektor „*Pericardium_Pressure*“.

Při systole ovlivňuje závislost hodnoty *ESV* na koncovém systolickém tlaku ještě stimulace (nebo blokování) beta receptorů, která se projevuje změnami stažlivé síly srdečního svalu. Podrobný popis rovnic, které tuto závislost popisují, je popsán v komponentě „*BetaReceptorsActivityFactor*“, jejímž výstupem je kauzální vstup do komponenty „systole“.



Obr. 79 – Model srdeční komory s chlopněmi, který generuje pulzační tok krve tep po tepu. Protože má pro zapojení do modelu vyšší hierarchické úrovně stejné vnější rozhraní, jako model pumpy.

Model srdeční komory na obr. 79 je vyjádřen nikoli jako pulzní, ale jako kontinuální čerpadlo s proměnlivým vnitřním objemem. Nemodelujeme tedy zde čerpání „tep po tepu“, ale průměrný minutový průtok srdeční (cardiac output).

Nejprve je (v komponentě „*StrokeVolume*“) vypočítán systolický objem, jako rozdíl mezi koncovým diastolickým (*EDV*) a koncovým systolickým (*ESV*) objemem. Hodnota minutového průtoku krve je počítána (v násobičce „*BloodFlow*“) ze systolického objemu vynásobeného frekvencí. Hodnota srdeční frekvence („*HeartVentricleRate*“) přichází zvnějšku ze sběrnice „*bloodFlowSignals*“.

Průměrný objem krve v komoře je odhadnut aritmetickým průměrem („*Vol_SteadyState*“) mezi objemem maximální náplně srdce v diastole (objem *EDV*) a objemem srdce na konci systoly (ESV).

Srdeční komora je vyjádřena (komponentou „*ventricle*“) jako kontinuální čerpadlo, které má proměnlivý vnitřní objem (komponenta je instancí modelu z obr.76). Do krevního oběhu je čerpadlo zapojeno přes akauzální konektory („q_*in*“ a „q_*out*“). Dvěma kauzálními konektory dostává vypočítanou hodnotu minutového objemu srdečního (*Blood_inflow*) a požadovanou průměrnou hodnotu vnitřního objemu čerpadla „*Volume_StaedyState*“).

Model srdce aproximovaného jako kontinuální čerpadlo je postačující (a dostatečně rychlý) pro řadu aplikací v lékařských simulátorech. Pokud ale např. chceme modelovat nejrůznější chlopňové vady, musíme použít podrobnější model, popisující chování komory tep po tepu.

Výměna jednodušší komponenty za složitější nemusí vždy znamenat předělat celý model. Zápis modelu v Modelice totiž umožňuje velmi elegantně zaměňovat komponenty jako různé varianty tříd se stejným rozhraním.

Například je možné uvnitř modelu subsystému krevního oběhu (obr. 77) vyměnit instance modelu levé a pravé komory (komponenty „*rightVentricle*“ a „*leftVentricle*“): místo modelu komor jako kontinuálního čerpadla (obr. 78) vložíme do schématu instance složitějšího modelu generujícího

proud krve tep po tepu. Stačí jen přetypovat instance levé a pravé komory.

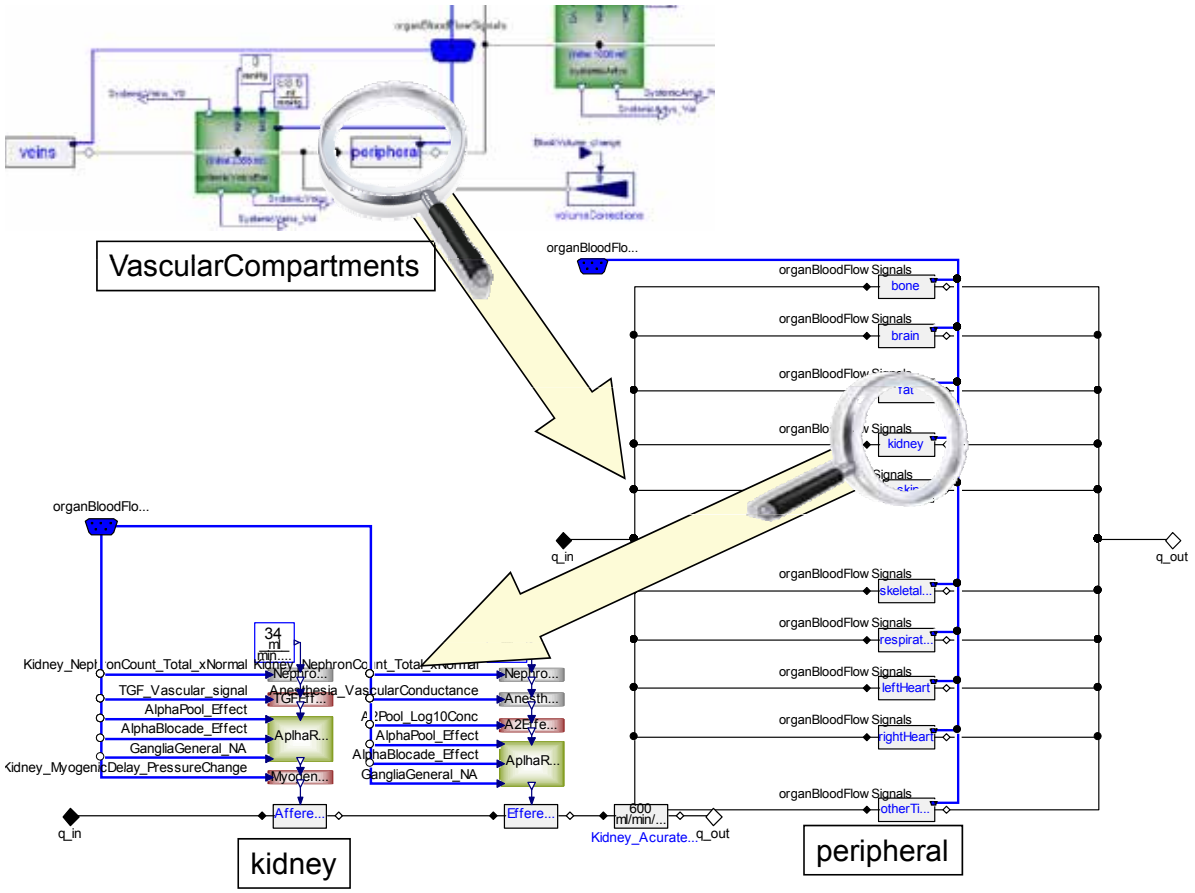
Základem modelu srdeční komory s chlopněmi, který generuje pulzační tok krve tep po tepu (obr. 79) je elastický komparment („*ventricle*“), který má (na rozdíl od využití elastického kompartmentu v cévách) generovanou oscilující hodnotu poddajnosti. Frekvence těchto oscilací je určená počtem tepů za minutu. Tvar jedné periody změny poddajnosti (komponenta „*curve*“) vyjadřuje vlastnosti srdečního svalu. Amplituda je ovlivněná stimulací a blokováním beta receptorů. Směr a rychlost proudu krve komorou se nakonec automaticky odvodí z vlastností komponent chlopní (komponenty „*valve1*“ a „*valve2*“) a z tlakových gradientů.

Jednoduchý model chlopně můžeme vyjádřit jako analogii sériového zapojení ideální diody s odporem. Alternativní (složitější) model chlopní umožní modelovat různé chopňové vady.

Záměnou komponent různé složitosti se stejným rozhraním můžeme podle potřeby vytvářet instance modelu různé složitosti podle účelu jejich aplikačního využití.

Možnost záměny jednotlivých komponent podporuje Modelica tím, že umožňuje definovat rozhraní s proměnným počtem řídících (vstupních, kauzálních) signálů. Podle počtu řídících signálů mohou být komponenty více komplexní, nebo naopak zjednodušené, a jejich funkci můžeme testovat v rámci zapojení do modelu vyšší hierarchické úrovně. Tato velká výhoda se dá efektivně využít nejen při ladění složitých modelů, ale i při identifikaci modelu podle experimentálních dat.

V Modelice je velmi důležité využívání hierarchičnosti a komponentové výstavby modelu (obr. 80). Pro architekturu stavby modelů je vhodné dodržovat pravidlo, aby se struktura komponenty vždy vešla na jednu obrazovku. Složitá spleť propojek nesvědčí o dobrém návrhu a vede ke zmatkům.



Obr. 80 – Hierarchické uspořádání modelů v Modelice. V komponentě „VascularCompartments“ (z obr. 30) je jeden periferní řízený odpor s názvem „peripheral“. Rozkliknutím se zobrazí řada paralelně zapojených řízených odporů. Rozkliknutím jednoho z nich – s názvem „kidney“ se zobrazí složitě řízené odpory v ledvinách. Kombinace akauzálních a (kauzálních) signálových vazeb a bohatost grafických možností pro zobrazení modelovaných vztahů umožňuje vytvářet hierarchicky členěné a „samodokumentující se“ modely.

Účelem této kapitoly nebylo popisovat fyziologii krevního oběhu. Na příkladě poněkud podrobnějšího popisu struktury některých komponent jsme chtěli pouze ilustrovat, jak akauzální modelovací nástroje umožňují vytvářet bohatě hierarchicky členěné, snadno modifikovatelné „samodokumentující se“ modely. Při modelování rozsáhlých systémů, jakými jsou např. modely propojených fyziologických regulací jako podklad lékařských simulátorů, je akauzální modelovací prostředí jazyka Modelica velkým pomocníkem.

4.3 Od Simulinku k Modelice

Rychlost, s jakou si nový simulační jazyk Modelica osvojila nejrůznější komerční vývojová prostředí je ohromující. Jestliže ještě před nedávnem existovaly pouze dvě komerční implementace tohoto jazyka (D*ymola* od Dynasimu a *MathModelica* od Mathcore), dnes (v prosinci 2009) již jazyk Modelica využívají také simulační prostředí *LMS Imagine.Lab AmeSim* od firmy LMS (http://www.lmsintl.com), *MapleSim* od Maplesoftu (http://www.maplesoft.com/), *Mosilab* od firmy Frauenhofer (http://www.fraunhofer.de) a *SimulationX* od společnosti ITI (http://www.iti.de).

Modelica proto v průmyslových aplikacích nachází stále větší uplatnění. Tento moderní simulační jazyk dnes využívají velké průmyslové korporace, jako Siemens, ABB a EDF. Firmy působící v automobilovém průmyslu, jako (AUDI, MBW, Daimler, Ford, Toyota, VW) používají Modelicu pro návrh energeticky úsporných automobilů a pro návrh klimatizačních jednotek. Rozvoj vývojových protředí a technologií využívajících jazyk Modelica i vývoj příslušných aplikačních knihoven je součástí celoevropských výzkumných projektů EUROSYSLIB, MODELISAR a OPENPROD financovaných celkovou částkou 54 milionů Euro (viz http://www.modelica.org/).

V biomedicínských aplikacích se však Modelica prozatím příliš neuplatnila.

Drtivá většina biomedicínských simulačních aplikací je dosud realizována v kauzálních blokově orientovaných prostředích. Patří k nim například vývojové prostředí referenčních databází biomedicínských modelů (v jazycích JSIM http://physiome.org/model/doku.php nebo CEllML http://www.cellml.org/).

Zhusta využívaným prostředím v biologii a medicíně je Matlab/Simulink - monografie věnované biomedicínským modelům bývají často doprovázeny přídatným softwarem pro toto vývojové prostředí, zatím ale bez využití nových akauzálních simulinkových knihoven (např. Wallish, Lusignan, Benayoun, Baker, Dickey, & Hatsopoulos, 2008; Logan & Wolesensky, 2009; Oomnes, Breklemans, & Baaijens, 2009).

Nicméně již v roce 2006 Cellier a Nebot ukázali výhody, které Modelica přináší pro přehlednou implementaci popisu fyziologických systémů. V Modelice implementovali klasický model McLeodův cirkulačního systému PHYSBE (PHYSiological Simulation Benchmark Experiment) (McLeod, 1966; McLeod, 1967; McLeod, 1970). Tyto rozdíly zvláště vyniknou, porovnáme-li si Cellierovu implementaci modelu (Cellier & Nebot, 2006) s volně stažitelnou verzí implementace modelu PHYSBE v Simulinku http://www.mathworks.com/products/demos/simulink/physbe/.

Haas a Burnhan ve své nedávno vydané monografii upozorňují na velké možnosti jazyka Modelica pro modelování medicínských adaptivních regulačních systémů (Haas & Burnham, 2008). Nejnověji Brudgárd a spol. (2009) referují o práci na implementaci knihovny značkovacího jazyka SBLM (System Biology Markup Language), používaného jako jeden ze standardů pro popisování modelů biologických systémů (http://sbml.org/), do jazyka Modelica. To by v budoucnu umožnilo jednoduchým způsobem přímo spouštět modely, jejichž struktura je popsaná v jazyce SBLM, na vývojových platformách, založených na jazyce Modelica (Brudgárd, a další, 2009).

Nicméně akauzální modely je dnes možné vytvářet i v Simulinku s využitím nových akauzálních knihoven (Simscape a dalších).

V Matlabu a Simulinku jsme po léta vyvíjeli modely fyziologických systémů a rozvíjeli příslušnou aplikační simulinkovou knihovnu Physiolibrary (http://physiome.cz/simchips). Vyvinuli jsme také příslušné softwarové nástroje usnadňující převod modelů implementovaných v Simulinku do vývojových prostředí (Control Web a Microsoft .NET), v nichž vytváříme vlastní výukové simulátory. Náš vývojový tým dlouholetou praxí poměrně slušné zkušenosti v práci s vývojovým prostředím Matlab/Simulink od

renomované firmy MathWorks. Na druhé straně nás lákaly nové možnosti vývojových prostředí využívající jazyk Modelica.

Stáli jsme proto před rozhodnutím, zda nadále pokračovat ve vývoji modelů fyziologických systémů v prostředí Simulink (s využitím nových akauzálních knihoven), nebo zda učinit radikálnější rozhodnutí a přejít na novou platformu jazyka Modelica.

Naše rozhodování ovlivnila snaha o implementaci rozsáhlého modelu Guytonových spolupracovníků a žáků – ***Quantitative Human Physiology (QHP)***, zmíněném v kapitole 2. Model dnes zřejmě patří k nejrozsáhlejším modelům fyziologických systémů (obsahuje více než 4 000 proměnných). Jeho struktura je na webových stránkách (http://physiology.umc.edu/themodelingworkshop) do všech podrobností zveřejňována jako open source. Popis modelu je však rozčleněn do více než dvou tisícovek XML souborů roztroušených do stovek adresářů, z nichž speciální solver vytváří a spouští simulátor (viz Obr .17 na straně 20).

Celková struktura modelu a jednotlivé návaznosti jsou proto velmi nepřehledné.

Pro přehledné zobrazení matematických vztahů z XML notace QHP jsme proto nejprve vytvořili ***speciální softwarový nástroj QHPView*** (obr. 19 na straně 21), avšak i s jeho pomocí se celková struktura modelu rozplétala poměrně obtížně.

Model jsme se nejprve pokusili implementovat v prostředí Simulink.

V modelu se vyskytuje celá řada vztahů, které vedou na řešení implicitních rovnic. Z tohoto důvodu je blokově orientovaná implementace modelu (kde výstupy jednoho bloku jsou využity jako vstupy do dalších bloků) velmi složitá a v průběhu implementace se stoupající složitostí modelu rapidně klesala jeho přehlednost. Využití nových akauzálních knihoven v takto složitém modelu se ukázalo problematickým a přehlednost modelu se příliš nezvýšila.

Proto jsme simulinkovou implementaci přerušili a model začali implementovat pomocí jazyka Modelica (s využitím vývojového prostředí Dymola (http://www.3ds.com/products/catia/portfolio/dymola).

Velmi rychle se ukázalo, že ***implementace rozsáhlého modelu v Modelice je mnohem efektivnější než pouhé využívání akauzálních knihoven v Simulinku***. Při porovnání simulinkové a modelicové implementace se projevil podstatný rozdíl, spočívající zřejmě v tom, že nové akauzální knihovny jsou pouhou akauzální nadstavbou Simulinku a nikoli objektově orientovaným na rovnicích postaveným modelovacím jazykem, jakým je Modelica.

Porovnáme-li tedy vývojová prostředí, založená na simulačním jazyce Modelica s vývojovým prostředím Matlab/Simulink od firmy Mathworks můžeme konstatovat že:

- na rozdíl od Simulinku model implementovaný v Modelice mnohem lépe vystihuje podstatu modelované reality a ***simulační modely jsou mnohem čitelnější i méně náchylné k chybám***;
- ***objektová architektura*** Modeliky umožňuje ***postupně stavět a ladit modely s hierarchickým uspořádáním s využitím knihoven znovupoužitelných prvků***;
- na rozdíl od Simulinku (který je průmyslovým standardem firmy Mathworks) je Modelica ***normalizovaný programovací jazyk,*** proto také mohou ***existovat různá komerční (i nekomerční) vzájemně si konkurující vývojová prostředí*** a pro řešení specifických problémů z různých aplikačních oblastí se v tomto jazyce vytvářejí ***(komerční i nekomerční) specializované knihovny***;
- v Modelice je možné ***nenásilně kombinovat kauzální (většinou signálové) a akauzální vazby***; na rozdíl od Simulinku v propojení kauzálních bloků je možné bez větších problémů vytvářet algebraické smyčky – součástí kompilátoru Modeliky jsou symbolické manipulace na pozadí a proto ***rozpojení algebraických smyček je starostí vývojového prostředí a nikoli programátora***.

Výše uvedené důvody nás vedly k tomu, že jako ***hlavní implementační prostředek pro tvorbu modelů*** jsme zvolili ***Modelicu*** a postupně upouštíme od vývoje modelů v prostředí Matlab/Simulink (Kofránek, Mateják, & Privitzer, 2009b).

Do tvorby aplikačních knihoven a nástrojů vývojových prostředí pro jazyk Modelica jsme se zapojili i v rámci mezinárodní spolupráce.

Společným úsilím 12 firem a 9 univerzit sdružených v Open Modelica Source Consortium je společně vyvíjeno prostředí Open Modelica, šiřitelné jako open source (Open Modelica Source Consortium - viz http://www.ida.liu.se/labs/pelab/modelica/OpenSourceModelicaConsortium.html). Tohoto vývoje se účastní i náš vývojový tým v rámci firmy Creative Connections s.r.o., která je členem tohoto konsorcia (viz http://www.creativeconnections.cz/). Pro toto konsorcium vyvíjíme nástroj, umožňující z modelu vyvinutého a odladěného v Modelice vygenerovat zdrojový text modelu v jazyce C#.

4.3.1 QHP v modelicovém kabátě

Výhody tvorby modelů v Modelice je možné názorně demonstrovat na příkladu implementace modelu QHP.

Porovnáme-li spletitou strukturu modelu QHP vizualizovanou pomocí QHPView (obr. 19 na straně 21) s ukázkami jeho implementace v simulačním jazyku Modelica na předchozích obrázcích 74-80, vidíme, že akauzální implementace vede k přehledné struktuře modelu a ke snadnějším následným modifikacím a úpravám modelu (Kofránek, Mateják & Privitzer, 2008).

Model QHP implementovaný v prostředí jazyka Modelica ***modifikujeme*** a dále ***rozšiřujeme***.

Úpravy a rozšíření QHP jsou z části převzaty z našeho původního modelu Golem (Kofránek, Anh Vu, Snášelová, Kerekeš & Velan, 2001) a dále modifikovány podle novějších poznatků.

Naše úpravy spočívají zejména v rozšíření, které zlepšuje použitelnost modelu pro modelování složitých poruch acidobazické, iontové, objemové a osmotické homeostázy vnitřního prostředí, což má, zejména v medicíně akutních stavů značný význam.

Naše modifikace modelu QHP spočívá zejména v ***přeprogramování subsystému acidobazické rovnováhy***, který je v původním modelu QHP založen na tzv. Stewartově pojetí acidobazické rovnováhy. Zjednodušeně řečeno, tzv. „moderní přístup“ Stewarta (Stewart, 1983) a jeho následovníků (např. Sirker, Rhodes & Grounds, 2001; Fencl, Jabor, Kazda & Figge, 2000) k vysvětlení poruch acidobazické rovnováhy vychází z matematických vztahů, počítajících koncentraci vodíkových iontů [H^+] z parciálního tlaku CO2 v plazmě (pCO2), celkové koncentrace ($[Buf_{tot}]$) slabých (neúplně disociovaných) kyselin ($[HBuf]$) a jejich bazí ($[Buf^-]$, kde $[Buf_{tot}]=[Buf^-]+[HBuf]$) a z rozdílu mezi koncentrací plně disociovaných kationtů a plně disociovaných aniontů – tzv. *SID* (strong ion difference):

$[H^+]=Funkce\ (pCO2, SID, Buf_{tot})$

Problémem tohoto pojetí je, že v modelu pak přesnost acidobazických výpočtů závisí na přesnosti výpočtu SID, tj. rozdílu mezi koncentrací plně disociovaných kationtů (tj. především sodíku a draslíku) a plně disociovaných aniontů (především chloridů). Nepřesnosti, které vznikají při modelování příjmu a vylučování sodíku, draslíku a chloridů se pak odrážejí v nepřesnostech modelování acidobazického stavu.

Přestože Coleman a spol. (2009) ve svém modelu QHP podstatně zlepšili přesnost modelování příjmu a vylučování sodíku, draslíku a chloridů v ledvinách, modelujeme-li dlouhodobý stav (kdy se s virtuálním pacientem nic neděje) má virtuální pacient v současné verzi modelu po měsíci simulovaného času tendenci upadat do lehké ustálené metabolické acidózy.

Náš bilanční přístup k modelování a hodnocení poruch acidobazické rovnováhy (Kofránek, 1980; Kofránek, Matoušek & Andrlík, 2007; Kofránek, 2009) vychází z modelování bilancí dvou toků – tvorby a vylučování CO_2 a tvorby a vylučování silných kyselin, propojených přes pufrační systémy jednotlivých oddílů tělních tekutin. Toto pojetí dle našeho názoru lépe vyskytuje fyziologickou kauzalitu acidobazických regulací, než nepřímé modelování acidobazických poruch přes bilanci doprovázejících elektrolytů. Krom toho se podstatně zlepšuje věrnost modelování zejména smíšených (acidobazických a elektrolytových) poruch vnitřního prostředí.

Další závažnou modifikací QHP je ***rozšíření modelu o závislost toku draslíku do buněk na vtoku glukózy způsobeném vlivem inzulínu***, což umožní mimo jiné modelovat vliv infúzí draselných roztoků s inzulínem a glukózou, které se v akutní medicíně podávají pro léčení deplece draslíku.

Bilanční přístup k modelování acidobazické rovnováhy jsme využívali již v našem starém simulátoru *„Golem“* – viz příklad simulace vzniku a neadekvátního léčení metabolické acidózy s následkem ohrožení života pacienta navozenou hypokalémií, která byla nakonec zvládnuta infúzí draselného roztoku s inzulínem a glukózou na obr. 21-32 na str. 26-31.

Rozšířený model QHP bude podkladem výukového trenažéru ***„eGolem“,*** určeného pro lékařskou výuku v oblasti klinické fyziologie akutních stavů, vyvíjeného v rámci výzkumného projektu MŠMT č. 2C067031, na webových stránkách tohoto projektu je možno i najít aktuální strukturu naší modelicové implementace modelu QHP (http://patf-biokyb.lf1.cuni.cz/wiki/projekty/e-golem).

4.4 Od modelu k simulátoru

Simulační model, implementovaný v tom nejrafinovanějším vývojovém prostředí není sám o sobě použitelný jako výuková pomůcka. Je jen implementací formalizovaného popisu modelované reality umožňující testovat chování matematického modelu při nejrůznějších hodnotách vstupů a hledat takové rovnice a parametry modelu, které by ve zvolených mezích přesnosti nakonec zajistily dostatečně dobrou shodu chování modelu s chováním modelovaného systému (identifikace modelu)

I po dosažení tohoto cíle je od identifikovaného modelu k výukovému simulátoru ještě poměrně dlouhá cesta (viz obr. 51 na str. 53). Díky pokroku v softwarových technologiích se však objevily nové nástroje nejen pro efektivnější vytváření simulačních modelů, ale také i prostředky, usnadňující tvorbu vlastních simulátorů s působivým vizuálním uživatelským rozhraním.

4.4.1 Kostra výukové simulační aplikace – scénář

Simulační technologie se zlepšily a zefektivnily, ceny zařízení i softwaru padají. Rozvoj internetu, virtuální 3D prostředí, jakým je např. již výše zmíněné prostředí Second Life (viz kapitola 2.8), nebo stále širší nabídka lékařských trenažérů využívajících robotizovanou figurínu pacienta přinášejí nové možnosti pro lékařskou výuku. Výuka pomocí simulátorů se v poslední době velmi rozšiřuje (zejména v USA a v Izraeli).

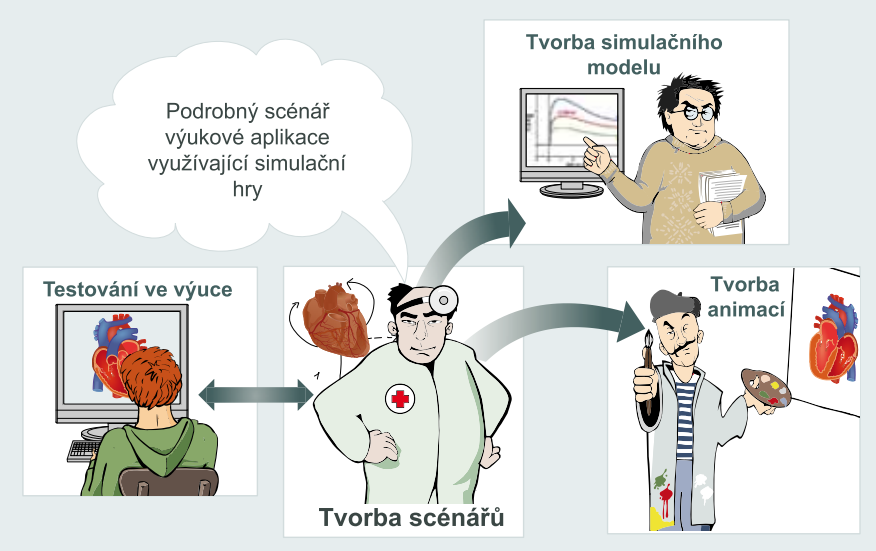
Výuka se simulátorem je z pedagogického hlediska vysoce účinná, klade však mnohem větší nároky na vyučujícího, než klasická výuka**.** Sebesložitější simulátor s tím nejatraktivnějším uživatelským rozhraním nemusí proto nutně být účinnou výukovou pomůckou. Pedagogická efektivita využití simulátoru především ***závisí na pedagogovi,*** který musí mít jasné představy, ***jakým způsobem a kde je nejvhodnější využít ve výuce simulační model***.

Čím složitější je výukový simulátor, tím důležitější je mít předem jasnou představu o scénáři výukových simulačních her, které s ním budeme provádět. To potvrzují i naše praktické zkušenosti s využitím složitých simulátorů ve výuce (např. se simulátorem Golem nebo s QHP). Ukázalo se, že uživatelské rozhraní, které při mnoha větvících se nabídkách a možnostech prohlížení hodnot stovek proměnných, studenty zbytečně rozptyluje. Bez jasného pedagogického vedení, na co a kdy se při dané simulační hře se složitým simulátorem podívat a jak interpretovat výsledky, má jejich využití mizivé výsledky.

O tom, ***jak*** může být simulátor pedagogicky využíván, je ovšem třeba přemýšlet především ***před*** tím, než se pustíme do jeho vytváření. To platí zejména, chceme-li vytvářet multimediální interaktivní výukové programy, dosažitelné po internetu, využívající simulační hry pro lepší pochopení a procvičení vykládané látky. Klíčem k úspěchu je dobrý ***scénář***.

První, na kom závisí úspěch vytvářené aplikace, je tedy ***zkušený pedagog***, který musí mít jasno v tom, co a jakými prostředky chce svým studentům pomocí multimediální výukové aplikace vysvětlit, kde a jak využít pro ozřejmění vykládané látky simulační model.

Kostrou výukové aplikace je její ***scénář***. Jeho základem je obvykle nějaký výukový text – skripta, kapitola v učebnici apod. Při tvorbě scénáře pro multimediální výukovou aplikaci však musíme myslet i na to, jak se bude výukový program jevit na obrazovce, jaká bude posloupnost jednotlivých obrazovek, jaké bude jejich výtvarné ztvárnění, kde budou umístěny interaktivní elementy, kde se bude zapojovat zvuk, jak budou vypadat jednotlivé animace, kde se vloží simulační model a jak bude ovládán, kde se vloží test znalostí, jak bude vypadat, jak se bude vyhodnocovat a jak se bude reagovat na jeho výsledek



Obr. 81 – Úloha pedagoga – tvůrce scénáře ve vývoji výukových simulátorů je klíčová. Autor scénáře musí ve spolupráci s tvůrcem modelu přesně definovat, jakým způsobem se bude model ve výukové aplikaci využívat a jak bude vypadat uživatelské rozhraní modelu. Pro výtvarníka musí připravit dostatečně podrobné zadání požadavků na výtvarné ztvárnění všech potřebných grafických prvků a animací. Důležité je testování vytvořeného simulátoru ve výuce, které obvykle přinese cenné podněty pro modifikaci výukové aplikace.

apod.

Finální podobu grafických prvků tvoří profesionální výtvarník. Proto je nesmírně důležitá ***dobrá komunikace pedagoga - tvůrce scénáře s výtvarníkem***. Pedagog nemusí umět dostatečně dobře kreslit, musí mít ale jasnou a promyšlenou představu o tom, co od výtvarníka potřebuje. Kamenem úrazu se zpočátku stávalo neustálé předělávání již nakreslených animací, způsobených obvykle původně ne zcela jasnou představou pedagoga o výtvarném ztvárnění multimediálního prvku ve scénáři. Věnovat pozornost pečlivé přípravě scénáře před počátkem vlastní realizace se proto vyplatí (obr. 81).

Při tvorbě scénáře výukové aplikace se nám osvědčilo využívat postup, který je znám u kresleného filmu – nakreslit (nejlépe ve spolupráci s výtvarníkem) obrazový scénář, tzv. „Story Board“ – hrubou posloupnost jednotlivých obrazovek, a ke každé obrazovce pak v klasickém textovém editoru napsat komentář (případně odkaz na příslušnou část textu).

Interaktivní multimediální program však nejsou do počítačové podoby jednoduše přepsaná skripta. Není to ani lineární posloupnost textů, zvuků a pohyblivých obrázků jako kreslený film. Výrazným rysem výukového programu je jeho ***interaktivita*** – a s tím spojená možnost větvení a vzájemného propojení jednotlivých částí. Přetvořit lineární textový a obrazový scénář do scénáře větveného, hypertextovými odkazy provázaného interaktivního programu ovšem není jednoduché.

Jedním z metodologických problémů, který bylo nutno při tvorbě scénářů našich výukových aplikací vyřešit, byl způsob ***jak ve scénáři zobrazit vlastní strukturu výukového programu,*** zahrnující výklad, interakce s uživatelem, větvení programu apod.

Nejjednodušší je v textovém či obrazovém editoru pomocí klasických blokových diagramů či struktogramů popsat příslušná větvení, rozhodovací bloky apod. s příslušnými odkazy na stránky textu a příslušné obrázky uložené v dalších souborech.

Při psaní scénáře se osvědčilo využít schopností moderních textových editorů a vytvářet příslušné hypertextové odkazy – tím již vlastní scénář dostává jisté rysy budoucí interaktivity.

Pro zápis scénáře výukové aplikace jsme také vyzkoušeli nástroj ***Adobe Captivate*** (http://www.adobe.com/cz/products/captivate/), který umožňuje rychle vytvářet profesionálně vypadající e-learningový obsah s pokročilou interaktivitou, bez velkých nároků na znalosti programování. Ukázalo se ale, že pro cíl – zapsat scénář ve formě interaktivně se větvícího storyboardu, je tento nástroj až příliš zbytečně komplikovaný. Naopak, ukázalo se, že pro sdílení postupně vznikajících scénářů mezi členy vývojového týmu stačí jednoduché nástroje. Pro specifikaci zadání vytváření grafických prvků výtvarníkům i sledování dosažených výsledků jejich práce apod., velmi postačoval nástroj ***Microsoft OneNote***.

Moderní interaktivní výukový program není ani do počítačové podoby převedený výukový animovaný film – nejvyspělejším rysem interaktivity, kterou počítač poskytuje, je právě možnost ***využití simulátoru***, který formou simulační hry umožní ve virtuální realitě ozřejmit vysvětlovaný problém. A scénář

výukového programu s tím musí počítat. Autor si musí položit otázky – jaké experimenty se simulačním modelem je vhodné nabídnout, jaké má být uživatelské rozhraní simulační hry, a konečně, jaké jsou požadavky na simulační model na pozadí.

Proto je důležité, aby „scénárista“ komunikoval i s tvůrcem modelu, aby znal strukturu modelu a mohl navrhnout její případnou modifikaci, či formuloval požadavky, které by model měl splňovat.

Klíčová je i pedagogická zkušenost. Nezřídka se ukazuje, že to, co se při návrhu scénáře výukové aplikace jevilo jako jednoduché a srozumitelné, může být v pedagogické praxi komplikované a obtížně pochopitelné. Kromě toho, právě při využívání simulační hry ve výuce se vyjeví nutnost modifikace ať již uživatelského rozhraní, nebo i simulačního modelu na pozadí aplikace.

Proto je důležité průběžně vytvářet návrhy scénářů v těsném kontaktu s ***pedagogickou praxí***. Proto se nám také osvědčilo si simulační aplikaci „nejdříve vyzkoušet na studentech“ při výuce, a poučeni z této zkušenosti pak vytvářet nebo dotvářet vysvětlující text, modifikovat simulátor a navrhovat konečnou formu uživatelského rozhraní v produkční verzi internetové e-learningové aplikace.

4.4.2 Svaly výukové simulační aplikace – interaktivní multimediální komponenty

Pro vytváření uživatelského rozhraní výukového simulátoru je velmi působivé simulátor navenek reprezentovat jako pohyblivé obrázky řízené simulačním modelem. Řízené animace mohou graficky reprezentovat význam číselných hodnot – např. schematický obrázek cévy se může roztahovat nebo komprimovat, srdce může rychle či pomaleji tepat, plíce mohou hlouběji či mělčeji „dýchat“, ručička měřicího přístroje se může pohybovat a průběžně zobrazovat hodnotu nějaké výstupní proměnné modelu čtené z běžícího simulačního modelu na pozadí apod. Na druhé straně můžeme přes vizuální prvky (přes nejrůznější tlačítka, knoflíky, táhla apod.) do simulačního modelu zadávat nejrůznější vstupy.

Grafický vzhled v nezanedbatelné míře rozhoduje o tom, jak bude výuková aplikace přijímána potenciálními uživateli.

Pro profesionální výsledný vzhled výukového simulátoru je proto nezbytné, aby ***vlastní animace vytvářel výtvarník*** – výsledky jsou neporovnatelně lepší, než když animace vytváří graficky nadaný programátor. Předpokladem ale je, že výtvarník musí dobře zvládat práci s nástroji pro tvorbu interaktivní grafiky. Takto vzdělaných výtvarníků je ale na trhu práce kritický nedostatek a ti, kteří tyto nástroje ovládají, jsou žádanými (dobře placenými) členy profesionálních týmů vytvářejících počítačové hry, webové portály, reklamní multimediální aplikace apod.

Znamenalo to ovšem ***věnovat nemalé úsilí výuce výtvarníků***, které jsme tyto dovednosti museli naučit. Proto jsme již před lety začali úzce spolupracovat se ***Střední uměleckou školou Václava Hollara*** a na této škole jsme otevřeli „laboratoř interaktivní grafiky“ jako detašované pracoviště Univerzity Karlovy.

Věnovali jsme značnou část našeho pracovního času tomu, abychom profesionální výtvarníky naučili pracovat s vývojovými nástroji pro tvorbu interaktivních animací (jako je Adobe Flash, Microsoft Expression Blend aj.), s nástroji pro tvorbu 3D grafiky (např. Cinema 3D), s nástroji pro zpracování videa (např. Adobe Premiere), poskládat vše do internetem dostupné aplikace (např. pomocí Adobe Flex) i naučit výtvarníky základům programového ovládání interaktivních animací a tvorby interaktivních webových stránek.

Naše úsilí mělo úspěch. Výtvarníci ztratili ostych před počítačem a záhy pochopili, že „digitálním štětec“ představuje jen další nástroj pro kreativní výtvarné vyjadřování, jehož ovládnutí navíc přináší další velké možnosti jejich pracovního uplatnění.

Iniciovali jsme také založení ***Vyšší odborné školy***, která vyučuje v tříletém studiu předmět „interaktivní grafika“. Pracovníci našeho Oddělení biokybernetiky a počítačové podpory výuky se v této Vyšší odborné škole mimo jiné podílejí i na výuce (http://www.hollarka.cz/).

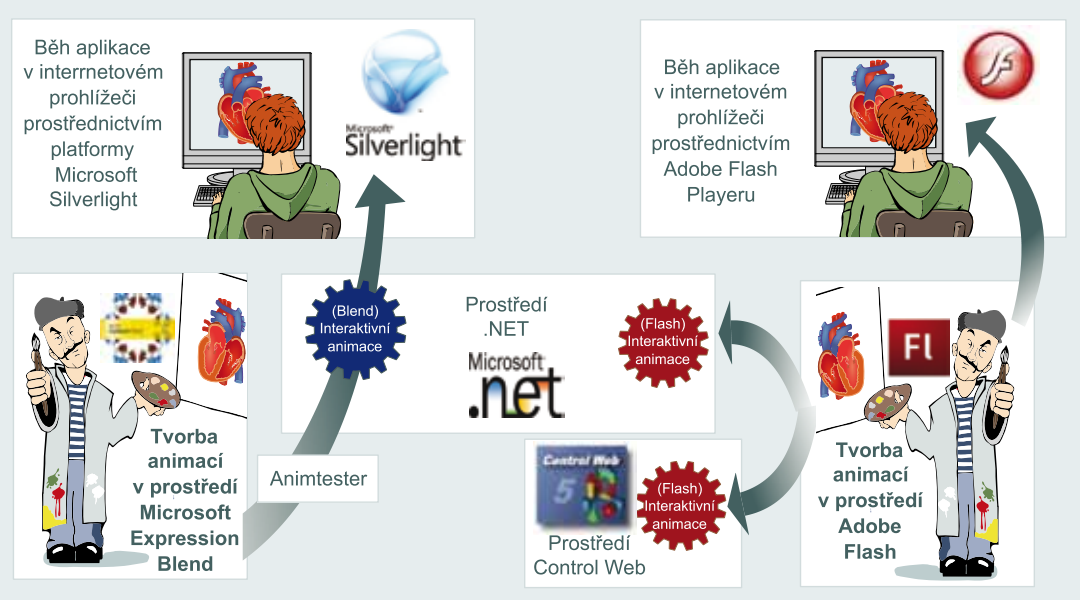
Výtvarník musí na jedné straně ***spolupracovat s pedagogem***, vytvářejícím scénář výukové aplikace, a na druhé straně s ***programátorem***, který zajistí adekvátní chování interaktivních animací (např. interaktivní animace propojí se simulačním modelem). Proto i výtvarník musí mít i některé základní programátorské znalosti, aby komunikace s programátorem byla efektivní.

V minulosti jsme pro vizuální rozhraní simulátorů (např. simulátoru Golem) využívali softwarové prostředí ***Control Web***, původně určené pro vizualizaci a řízení průmyslových procesů, které nabízí řadu předpřipravených prvků – virtuálních přístrojů, z nichž šlo velmi pohodlně a rychle sestavit uživatelské rozhraní. Za rychlost a snadnost sestavení se ale platilo tím, že námi vytvořený simulátor měl spíše tvar řídícího pultu technologického procesu než interaktivní obrázek z lékařské učebnice (viz např. obr. 47 na str.45).

S příchodem profesionálních výtvarníků do našeho vývojového týmu se možnosti vizualizace značně posunuly a ve výukových programech jsme již nebyli omezováni nabídkou předpřipravených prvků. Mohli jsme vytvářet libovolné tvary animovaných komponent, které mohly být propojeny se simulačním modelem a jako loutky řízeny hodnotami na výstupech či vstupech simulačního modelu.

Pro vytváření interaktivních multimediálních komponent, propojitelných se simulačním modelem na pozadí, využíváme dva nástroje (obr. 82):

- Prvním z nich je ***Adobe Flash***, v němž můžeme vytvářet animované interaktivní komponenty, jejichž chování se dá programovat (a v našich aplikacích propojit se simulačním modelem). Vytvořené komponenty se dají na platformách různých operačních systémů (s využitím volně stažitelného doplňku internetového prohlížeče - Adobe Flash Playeru) snadno přehrávat v prohlížeči internetových stránek. Ve Flashi se tak dají vytvářet i numericky méně náročné simulátory spustitelné přímo v prohlížeči webových stránek. Flashové komponenty jsme také využívali jako vizuální rozhraní komunikující s jádrem simulátoru (prostřednictvím ActiveX komponent) v simulátorech vytvořených v prostředí Control Web a v prostředí .NET.
- Druhým nástrojem, který jsme začali využívat pro tvorbu grafických komponent simulátorů teprve v nedávné době, je vývojové prostředí ***Microsoft Expression Blend***. Toto prostředí umožňuje (s využitím vývojového prostředí Microsoft .NET) vytvářet aplikace přímo spustitelné v internetovém prohlížeči prostřednictvím volně stažitelného doplňku ***Microsoft Silverlight***. Platforma Microsoft Silverlight umožňuje běh rozsáhlých numericky náročných



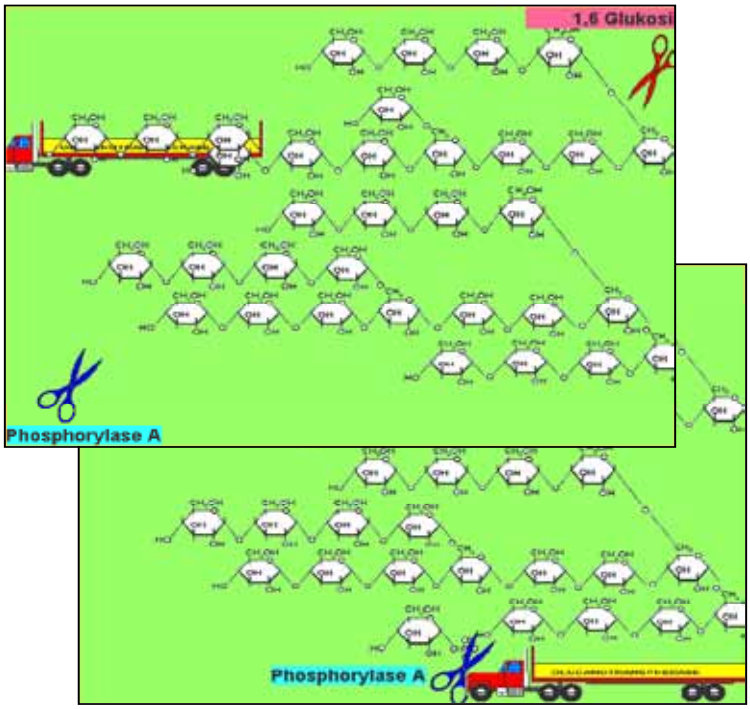
Obr. 82 – Na bedrech výtvarníků spočívá odpovědnost za tvorbu multimediálních komponent, zejména interaktivních animací, propojených se simulačním jádrem výukových simulátorů. Animační komponenty vytvářené v prostředí Adobe Flash využíváme v numericky méně náročných simulátorech spustitelných v internetovém prohlížeči nebo jako prvky uživatelského rozhraní simulátorů vytvářených ve vývojových prostředích Control Web nebo Microsoft .NET. V poslední době využíváme též nástroj Microsoft Expression Blend pro tvorbu grafických komponent pro multimediální simulátory spustitelné prostřednictvím platformy Silverlight přímo v internetovém prohlížeči. Pro usnadnění tvorby animací, které budou řízeny simulačním modelem, využíváme námi vytvořený nástroj Animtester.

aplikací s interaktivním multimediálním rozhraním. Tato nová platforma Microsoftu umožňuje pomocí internetu distribuovat numericky náročné multimediální simulátory spustitelné přímo v internetovém prohlížeči.

4.4.3 Animační štětec pro výtvarníky - Adobe Flash

Flash prošel poměrně dlouhým vývojem. Původně to byl především program od firmy Macromedia určený pro pouhé vytváření animovaných obrázků. V té době jsme pro vytváření výukových animací používali jiný produkt Macomedie – Director (obr. 83), který umožňoval animace řídit pomocí skriptů (Kofránek & Svačina, 2001).

Postupně se rozšiřovala možnost vytvářené animace řídit pomocí skriptů i ve Flashi, jejichž syntaxe se postupně vyvíjela a obohacovala. Od verze 7 (prodávané pod názvem Macromedia Flash MX 2004) byl již do prostředí Flash zabudován již skutečně objektový řídící jazyk (ActionScript), syntaxí velmi podobný jazyku Java, který umožňuje poměrně pohodlné a sofistikované řízení chování vizuálních interaktivních elementů.



Obr. 83 – Ukázka z multimediálního výukového programu s interaktivními grafickými animacemi, vytvořeného v programu Macromedia Director v roce 2000. Director v té době měl silnější podporu pro programování interaktivity než Flash. Flash ale záhy tento hendikep vyrovnal a později poskytl bohatší možnosti řízení interaktivity než Director. Kromě toho měl více intuitivní uživatelské rozhraní, připomínající nástroje pro tvorbu animovaných filmů a mezi výtvarníky si rychle získal oblibu. Proto jsme po roce 2000 Director opustili a začali vytvářet interaktivní animace ve Flashi.

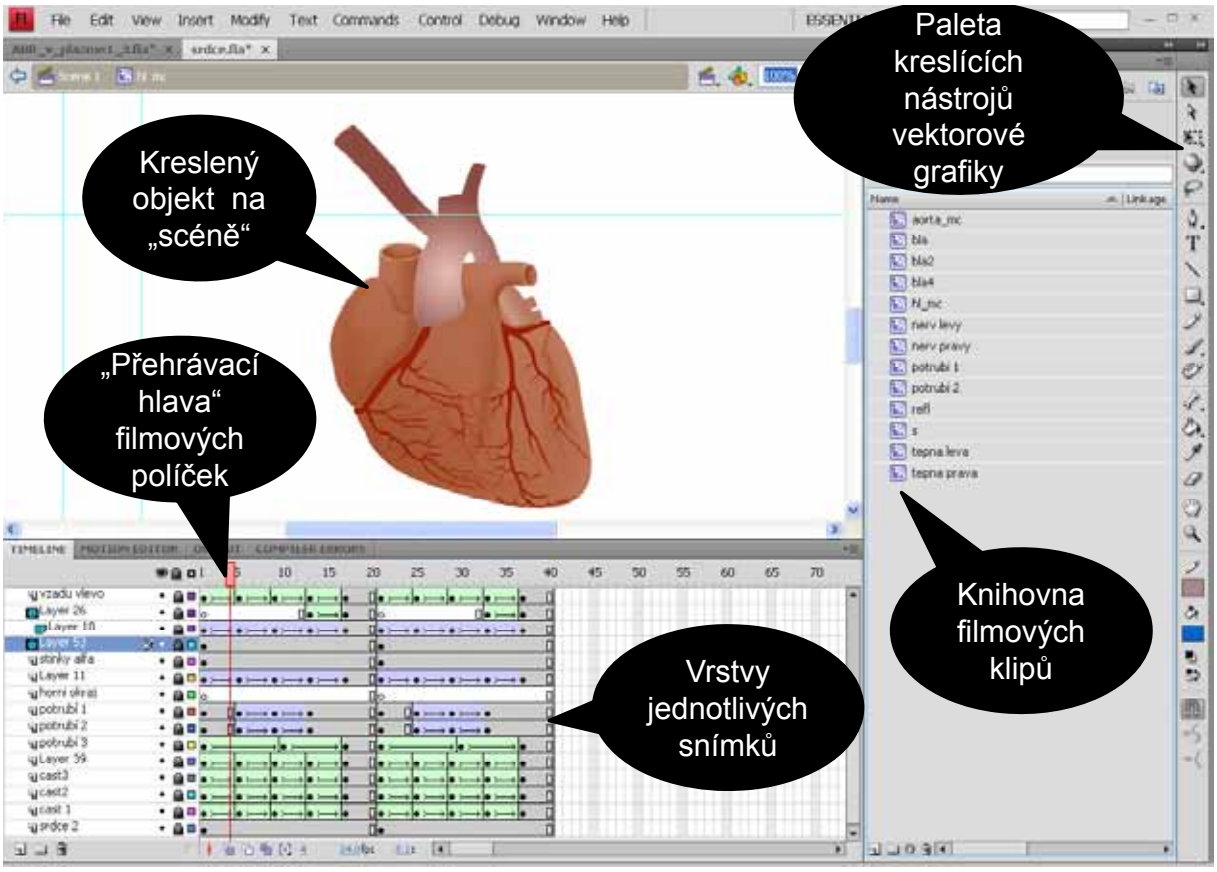
Velký úspěch vývojového prostředí Macromedia Flash je mimo jiné založen na tom, že tvůrcům se poměrně úspěšně podařilo definovat ***rozhraní pro výtvarníky*** (vytvářející základní animační prvky) a ***programátory***, kteří těmto komponentám mohou pomocí výše zmíněného objektového jazyka vdechnout interaktivnost.

Základní komponentou Flashových aplikací je film (movie). Film je možné dělit na jednotlivé scény, které je možno postupně či programově (na přeskáčku) přehrávat. Scény jsou tvořené sekvencí jednotlivých snímků (frames). Filmy (movies) je možno libovolně řetězit – z každého filmu (movie) je možné programově zavolat zavedení dalšího filmu, a pak spustit jeho přehrávání. To má výhodu zejména v internetových aplikacích, kdy po zavedení a spuštění první části animace se na pozadí z příslušného serveru stahuje její další část.

Při vytváření počítačových animací se vychází z klasického způsobu tvorby kresleného filmu, kde se pro každé políčko kreslí jednotlivé součásti na průhledných průsvitkách vrstvených na sebe. Některé průsvitky se nemusí pro další políčko překreslovat, nebo se jen o málo posunou (např. pozadí), zatímco jiné (např. postavička) se na každém políčku překreslují, buď celé, nebo jen zčásti, aby postupně vznikl animovaný pohyb.

Ve Flashi (obr. 84) je každá scéna tvořena několika vrstvami, které fungují obdobně jako průsvitky při ruční tvorbě kresleného filmu.

Každé políčko filmu, resp. scény (frame) má tedy několik vrstev, do nichž se ukládají jednotlivé obrazové prvky. Tyto vizuální prvky se mohou v každé vrstvě samostatně kreslit – Flash obsahuje poměrně výkonný nástroj pro vytváření vektorových obrázků (vektorové obrázky je samozřejmě možné i impor-



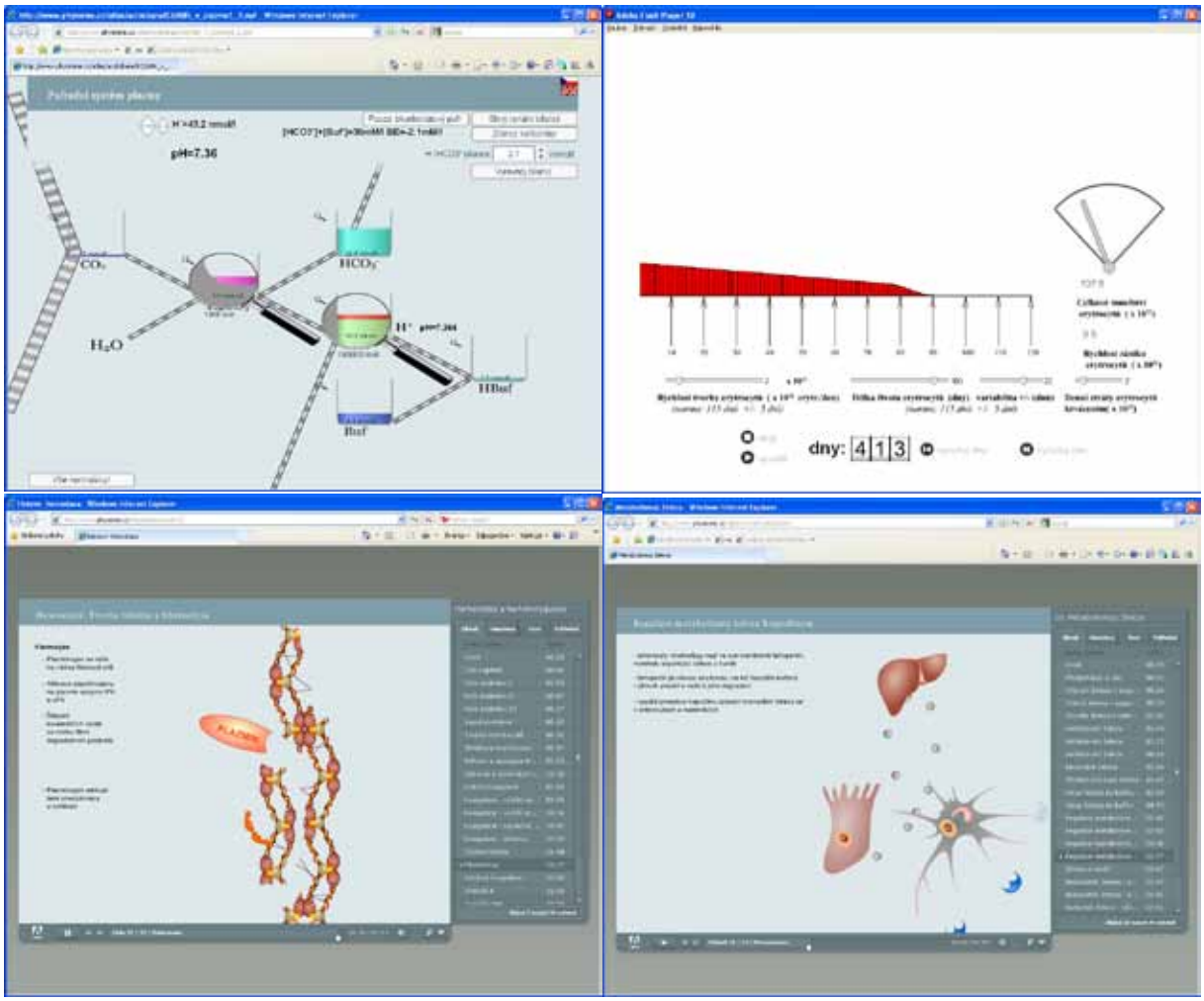
Obr. 84 – . Prostředí Adobe Flash poskytuje výtvarníkům nástroje pro kreslení vektorových obrázků. Do jednotlivých vrstev filmových políček je však (jako v příkladě na obrázku) také možno vložit instanci filmového klipu (MovieClipu) z knihovny. Chování jednotlivých vizuálních (i nevizuálních) komponent je možno programovat (ve speciálním okně pro programování).

tovat z mnoha jiných externích kreslících programů). Jinou možností je vybrat si z knihovny příslušný obrázek a vytvořit jeho instanci. Instancí však nemusí být jenom statický obrázek. Instancí může být i tzv. filmový klip (MovieClip), který je vlastně instancovanou třídou již dříve vytvořeného filmu. Tak například při vytváření obrázku letadla můžeme jako jeden jeho element do jedné z vrstev vložit z knihovny instanci filmového klipu s točící se vrtulí. Speciálním druhem filmových klipů jsou tlačítka (Buttons), u nichž je možno definovat jak jejich vizuální vzhled (při přejetí kurzoru nad tlačítkem a při jeho stisku), tak i chování obsloužením příslušné vyvolané události. Vlastní MovieClip může mít i poměrně složitou hierarchickou strukturu – film, který ho vytváří může obsahovat instance dalších filmových klipů. Tak např. MovieClip auta může v sobě obsahovat MovieClipy jeho točících se kol. Každá instance MovieClipu má své vlastnosti (souřadnice umístění na scéně, velikost, přebarvení, průhlednost apod.), které je možné dynamicky měnit z programu. Kromě toho, třída MovieClip má celou řadu metod, které můžeme využívat (např. metodu pro detekci kolize z jinou instancí MovieClipu na scéně apod.).

Při tvorbě MovieClipu můžeme také naprogramovat specifické metody, které pak můžeme volat u všech jeho instancí. Tak je možné naprogramovat složité chování vizuálních komponent. Poměrně jednoduše je možno vytvářet speciální MovieClipy jako skutečné komponenty, a jejich vlastnosti pak nastavovat ve speciálním editoru komponent a za běhu volat jejich metody. To otevřelo možnost vytváření (a následné distribuci či prodeji) nejrůznějších vizuálních (ale i nevizuálních) komponent od řady tvůrců a přispělo k velkému rozšíření Flashe mezi výtvarnou komunitou.

Ve vývojovém prostředí je možné jednotlivé filmy vytvořit (jak po grafické, tak i po programátorské stránce), otestovat a přeložit do mezijazyka (ve formě .swf souboru), který je možno interpretovat pomocí volně stažitelného interpretu (tzv. Flash Playeru) a přehrávat buď jako samostatně spustitelnou animaci nebo ji prohlížet přímo v internetovém prohlížeči (viz obr. 82).

Kromě toho je možné vytvořený .swf soubor interpretovat pomocí speciální ActiveX komponenty,



Obr. 85 – Prostředí Adobe Flash je možné využít pro implementaci simulátorů spustitelných přímo v internetovém prohlížeči - např. model acidobazické rovnováhy plazmy nebo model anémie. Flash byl doposud i hlavním implementačním prostředím pro ozvučené výkladové kapitoly našeho internetového Atlasu fyziologie a patofyziologie. Ve výkladových kapitolách jsou animace plně synchronizovány se zvukem. Pomocí jezdce umožňují výklad libovolně zastavovat, posouvat či navracet a přitom zůstává plná synchronizace animaci.

kterou je možno zabudovat do jiného programu – např. do aplikace vytvářené ve Control Webu nebo v Microsoft Visual Studiu. Důležité je, že tato komponenta si s aplikací může vyměňovat zprávy a tak můžeme jinou aplikací pohodlně řídit chování interaktivní animace. Aplikace zároveň může od interaktivní animace přijímat zprávy, referující o zásazích uživatele.

Velký úspěch Flashe vedl k tomu, že firma Macromedia byla zakoupena firmou Adobe a Flash se stal jednou z integrálních částí portfolia nástrojů počítačové grafiky tohoto výrobce.

Flashové komponenty se nyní dají využít v tzv. RIA (Rich Internet Application) – nové generaci multiplatformních webových aplikací s propracovaným designem komplexního uživatelského rozhraní, vytvářených pomocí nástroje ***Adobe Flex*** či jako desktopové aplikace vytvářené pomocí nástroje ***Adobe Air***.

Rychlost interpretu „.swf“ souborů (flash playeru) se zvýšila a v jazyku ActionScript je možné vytvářet i vlastní simulační jádro výukových simulátorů. Výhodou čistě flashových výukových aplikací (které mohou být i složité RIA aplikace, zkomponované v prostředí Adobe Flex***)*** je to, že je možno je spouštět přímo z internetového prohlížeče (který má příslušný plugin) a na všech platformách vypadají stejně.

V tomto prostředí jsme vytvořili některé výukové simulátory a výukové multimediální interaktivní aplikace se simulačními hrami (viz obr. 85).

Ve Flashi jsme v roce 2004 mimo jiné vytvořili interaktivní multimediální výukovou aplikaci: „Hud-

ba pohledem fyziky a fyziologie. Teorie ladění“ (http://patf-biokyb.lf1.cuni.cz/~obdrzalek/ladeni.htm) za níž jsme obdrželi na festivalu TechFilm cenu „Laureát festivalu TECHFILM“(Obdržálek & Kofránek, 2004). Ve Flashi jsme dokonce vytvořili i dvacetiminutový krátký film „Dějinné setkání“ (Kofránek, 2006), věnovaný historii česko-švédských vztahů v době třicetileté války.

Flash je také současnou platformou, v níž je implementován náš Atlas fyziologie a patofyziologie (Kofránek, a další, 2007).

Prostředí Flash playeru je nicméně stále ještě prostředím, založeným na interpretaci flashových .swf souborů. Pro numericky náročnější výpočty u složitějších simulátorů zde narážíme na určitou ***bariéru výkonu***. Pro komplikovanější simulátory prostředí Adobe Flash samo o sobě (zatím) nestačí.

Složitější simulátory jsou vytvářeny v ***prostředí .NET*** (a v minulosti i v prostředí ***Control Web***). Flashové komponenty do takto vytvářených simulátorů začleňujeme prostřednictvím komponenty Active X. Překlenování dvou nesourodých světů Adobe Flash a .NET ovšem klade další nároky na (ruční) programátorskou práci.

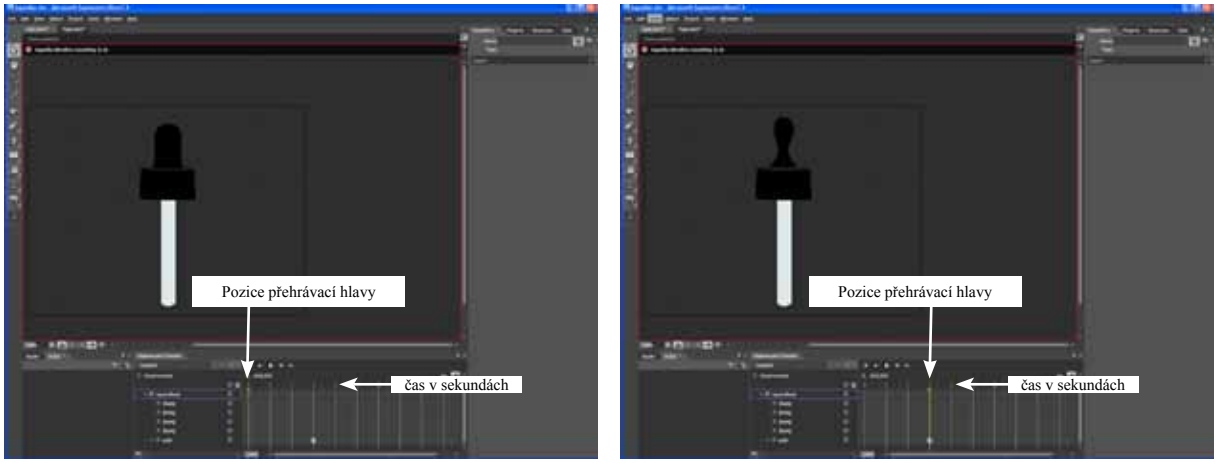
4.4.4 Microsoft Expression Blend - nástroj pro tvorbu „grafických loutek“ pro simulátory

Pro grafické aplikace je ale možné využít platformu Microsoft .NET přímo (a obejít se bez propojování s komponentami Adobe Flash). Díky technologii ***WPF - Windows Presentation Foundation*** (Sells & Griffins, 2007) je dnes možno přímo v platformě .NET vytvářet složité grafické komponenty obsahující animace, vektorovou grafiku, 3D prvky apod. Grafické prvky je možné vytvářet obdobně jako v prostředí Adobe Flash, ale s potenciálně většími možnostmi ovládání jejich chování než v prostředí Adobe Flash.

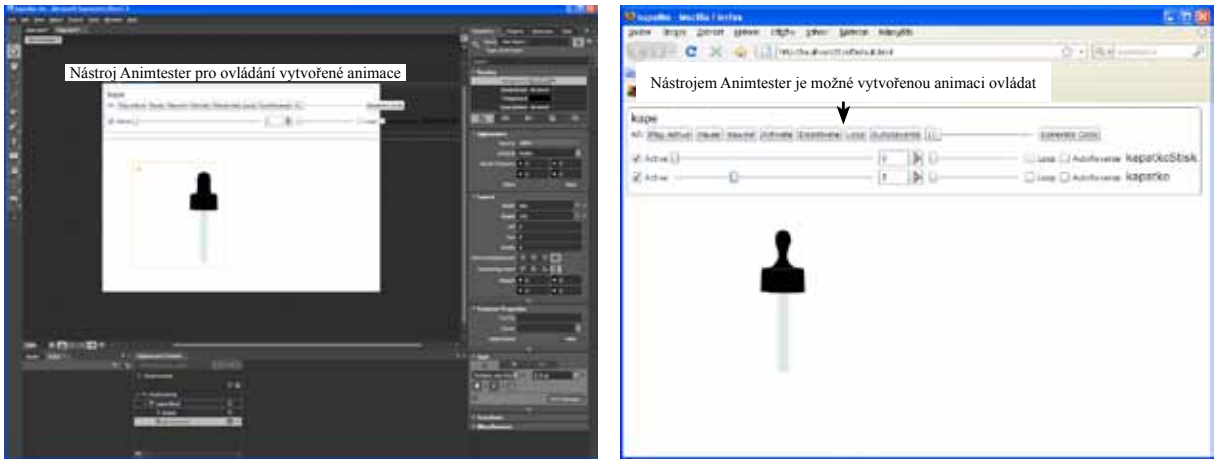
Krom toho, Microsoft reagoval na hojně rozšířený doplněk internetových simulátorů Adobe Flash vytvořením vlastní platformy ***Silverlight***, umožňující, obdobně jako Flash Player, spouštět složité aplikace kombinující text, vektorovou i bitmapovou grafiku, animace a video v internetovém prohlížeči. Aplikace primárně běží v internetovém prohlížeči, aniž by ji bylo nutno zvlášť instalovat (jediná potřebná instalace je samotný Silverlight plugin). Pomocí malé stažitelné komponenty tedy Silverlight umožňuje interaktivní ovládání aplikací ve většině současných webových prohlížečů (Internet Explorer, Firefox, Safari) na různých hardwarových a softwarových platformách. Přímo jsou nyní podporovány operační systémy Windows a Mac pro nejpoužívanější prohlížeče. Pro Linux je vyvíjená plně kompatibilní open source implementace Moonlight. Aplikace vytvořené pro tuto platformu využívají podstatnou část .NET frameworku, který je součástí pluginu (a tudíž mohou provádět i poměrně složité výpočty).

Silverlight je tedy platformou umožňující přes internet distribuovat ***simulátory, které mohou běžet přímo v internetovém prohlížeči*** (a to ***i na počítačích s různými operačními systémy*** – stačí aby prohlížeč měl instalován příslušný doplněk).

Důležitou vlastností ***Silverlightu*** je, že v sobě ***zahrnuje nativní podporu animací*** (Little, Beres,



Obr. 86 – Klíčové snímky animací, vytvářených v prostředí Microsoft Expression Blend jsou umísťovány přímo do časové osy. To mimo jiné následně zjednoduší synchronizaci animací se zvukovou stopou.

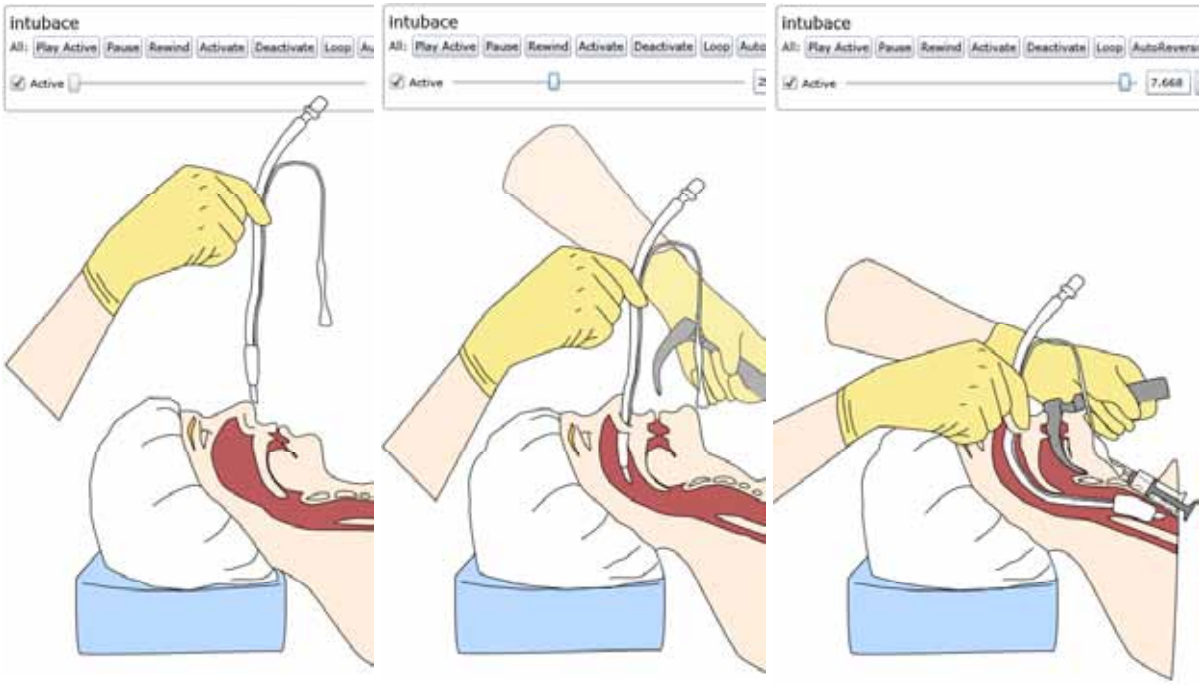


Obr. 87 – V prostředí Microsoft Expression Blend se vytvořené animace stávají vlastností vytvářených grafických prvků. Nastavením hodnoty, odvozené z hodnoty na časové ose je možno ovládat tvar zobrazeného animovaného grafického prvku. Nástroj Animtester umožní propojit tyto vlastnosti s vizuálně zobrazitelným ovládacím prvkem, kterým je možné animaci ve spuštěné aplikaci ovládat a testovat tak chování vytvořeného animovatelného grafického prvku. Nástroj slouží k mezioborové komunikaci výtvarníka, pedagoga - tvůrce návrhu výukové aplikace a programátora, který ve vytvářené výukové aplikaci grafické prvky propojuje se simulačním modelem.

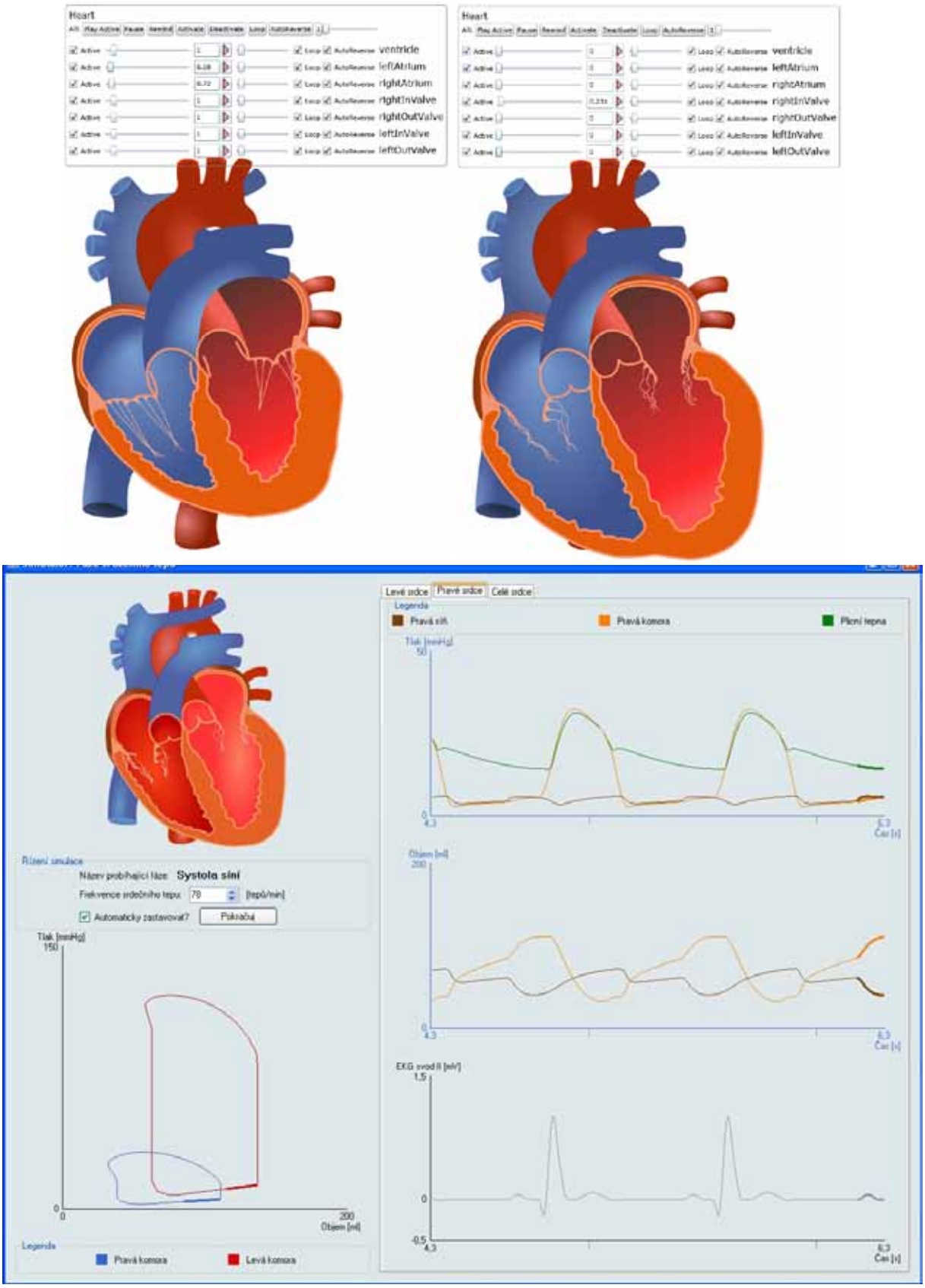
Hinkson, Rader & Croney, 2009). Animace jsou tedy přímo součástí aplikací a není potřeba pro grafickou vrstvu používat další dodatečnou platformu (např. Adobe Flash).

Pokročilejší je i způsob animace. V prostředí Adobe Flash se animace řídí přehráváním jednotlivých snímků zadanou rychlostí přehrávání (viz obr. 84 na str. 92 a obr. 95 na str. 104). Když se při přehrávání animace na klientském počítači nestačí včas všechny snímky vykreslit, přehrávání některých snímků se přeskočí a animace je „trhaná“. V Silverlightu se ale animace řídí přímo časovou osou. To mimo jiné vede k plynulejšímu přehrávání animací, protože snímková rychlost přehrávání se plynule nastaví podle zdrojů na klientském počítači, kde je animace přehrávána.

Microsoft vytvořil i nástroj pro pohodlnou tvorbu grafických prvků a tvorbu animací – **Microsoft Expression Blend**. V tomto nástroji můžeme vytvářet grafické rozhraní pro ***Silverlight***.



Obr. 88 – Intubace pacienta. Příklad komplexní animace vytvořené v prostředí Expression Blend s využitím Animtesteru.



Obr. 89 – Animace tlukoucího srdce. Výstupy modelu ovlivňují fáze tlukotu srdce, otevírání a zavírání chlopní atd. Nad vlastní animací jsou pomocné ovládací prvky Animtesteru umožňující grafikovi ladit jednotlivé subanimace. Grafik je plně odstíněn od programování. Ve výsledném simulátoru pak za příslušná táhlíčka obrazně řečeno „tahá" simulační model naprogramovaný na pozadí.

Microsoft Expression Blend má ***rozhraní pro výtvarníka*** i pro ***programátora.*** Microsoft Expression Blend zároveň pracuje přímo nad projektem vytvářené aplikace ve Visual Studiu .NET (Williams, 2008). Tím eliminuje potřebu převádět návrhy od designéra do projektu aplikace, což ***podstatně zjednodušuje spolupráci programátora a výtvarníka***.

V prostředí ***Microsoft Expression Blend*** jsou klíčové snímky animací vytvářeny nikoli po jednotlivých políčkách filmu (jako u Adobe Flash), ale podle časové osy (obr. 86). Nakreslené animace se pak dají vyjádřit jako vlastnosti objektů a tvar animovaného objektu se dá nastavit hodnotou odvozenou z časové hodnoty animace (obr. 87). Tím můžeme zadáním hodnoty vytvořené vlastnosti ovládat tvar animovaného grafického prvku.

Vytvořený ovladatelný animovaný grafický objekt můžeme použít i jako komponentu pro další, složitější animovaný objekt a jeho tvar ovládat nastavováním hodnot příslušných vlastností. Můžeme tak vytvořit animační „loutku“, jejíž výsledný tvar závisí na momentálním nastavení hodnot jejich jednotlivých komponent.

Pro snadnější komunikaci výtvarníka s programátorem, implementujícím vlastní simulátor, i pro komunikaci výtvarníka s autorem návrhu výukové aplikace, jsme vytvořili pomocný softwarový nástroj ***Animtester*** (Kofránek, Mateják, & Privitzer, 2009a), s jejichž využitím mohou designéři-grafici tyto animační „loutky“ vytvářet a ladit bez nutnosti dalšího programování (viz obr. 88-89). Animtester se vloží jako komponenta do vývojového nástroje Microsoft Expression Blend a umožní z vytvořených animací vygenerovat vlastnosti grafického prvku ovládané zvenčí pomocí ovládacích prvků (šoupátek a tlačítek). V následně vygenerované aplikaci si jak výtvarník, tak i autor návrhu výukové aplikace (pedagog) může ověřit, jak se animovaná komponenta bude chovat.

Výtvarník je tak odstíněn od programátorských detailů aplikace. Obdobně, tvůrce návrhu výukové aplikace se nemusí věnovat detailům implementace grafického návrhu a může v komunikaci s výtvarníkem snadněji dosáhnout realizaci svých představ.

Úlohou programátora, implementujícího vlastní simulátor, je propojit vygenerovaný grafický objekt se simulačním modelem na pozadí. Takto vytvořené animační „loutky“ je možné, přes hodnoty ovládající jejich tvar, ***přímo napojit na výstupy modelů*** a ***není potřeba*** zvlášť přidávat ***další programovou mezivrstvu pro propagaci dat***, jako tomu bylo při použití ***Flashových animací.***

Použití grafických možností platformy ***Silverlight*** tedy více než nahrazuje původní přístup s použitím animací založených na platformě Adobe Flash. Při tvorbě animací jako vizuálního rozhraní pro simulátory proto ***nepotřebujeme platformu Adobe Flash,*** kterou je možno ***plně nahradit novými nástroji pro tvorbu animací firmy Microsoft.***

4.4.5 Mozek výukové aplikace – simulační model

Implementace simulačních modelů do výukového programu není triviální problém.

Pro tvorbu simulačních modelů využíváme speciální vývojové nástroje, určené pro tvorbu, ladění a verifikaci simulačních modelů (***Matlab/Simulink*** či akauzální vývojové prostředí využívající jazyk ***Modelica***), o kterých jsme pojednávali v předchozích kapitolách.

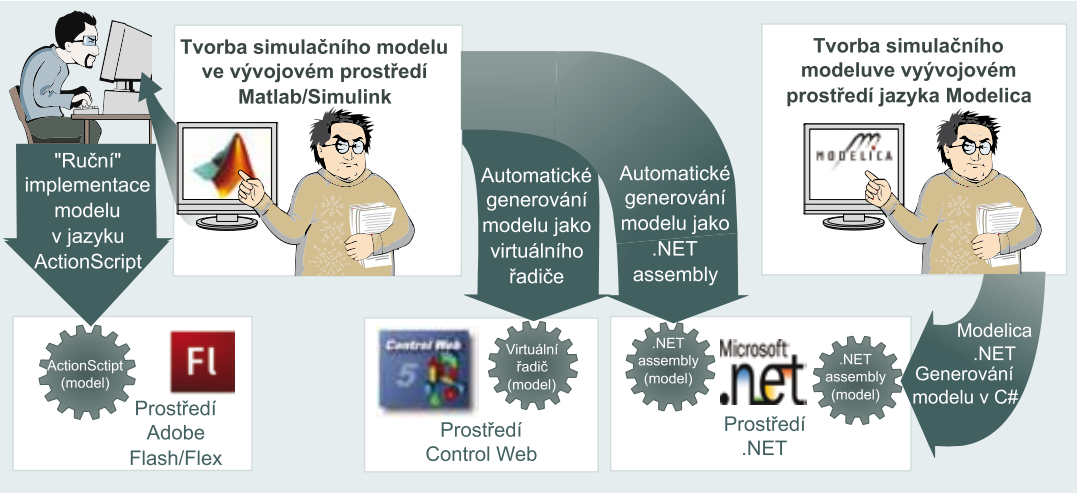
Odladěné a verifikované ***modely*** je ale potřeba ***převést*** z vývojového prostředí, v němž jsme vytvořili, odladili a verifikovali, ***do prostředí, kde je vytvářen vlastní výukový simulátor***.

U jednoduchých modelů se to dá udělat „ručně“ – tak to často děláme u čistě flashových výukových simulátorů, kde vývojovým prostředím pro tvorbu simulátorů je pouze Adobe Flash.

Pro složitější modely jsme si však raději vytvořili softwarové nástroje, které nám tuto práci zautomatizují (viz obr. 90).

V simulátoru Golem, implementovaném v prostředí ControlWeb byl model reprezentován jako ovladač virtuální měřící/řídící karty (viz obr. 46 na straně 44) Pro automatizaci převodu simulačního modelu z prostředí Matlab/Simulink jsme vytvořili generátor, který vytváří zdrojový text tohoto ovladače v jazyce C přímo ze Simulinkového modelu. (Kofránek, Kripner, Andrlík & Mašek, 2003).

Obdobně, abychom si ulehčili vytváření simulátorů, vytvářených ve Visual Studiu .NET (tj. aby nebylo nutné ve Visual Studiu .NET „ručně“ programovat již odladěný simulační model) vyvinuli jsme i zde



Obr. 90 – Pro vývoj simulačních modelů využíváme simulační prostředí Matlab/Simulink a v poslední době vývojové prostředí jazyka Modelica (např. Dymola nebo MathModelica). Identifikovaný simulační model můžeme ručně „přepsat" do prostředí, v němž bude vytvářen vlastní simulátor. To se vyplatí jen u jednoduchých simulátorů vytvářených v prostředí AdobeFlash (resp. Adobe Flex). U složitějších simulátorů vytvářených v prostředí .NET (nebo dříve i ControlWeb) jsme vytvořili speciální nástroje automaticky generující simulační jádro modelu ze simulinkového modelu. Pro vývojové prostředí jazyka Modelica vytváříme nástroj, generující model v jazyku C# což umožní tvorbu webových simulátorů přímo spustitelných v internetovém prohlížeči (pomocí platformy Silverlight).

speciální softwarový nástroj (Kofránek, Andrlík, Kripner & Stodulka, 2005; Stodulka, Privitzer, Kofránek & Vacek, 2007), který automaticky ze Simulinku vygeneruje simulační model ve formě komponenty pro prostředí .NET.

Výstupem programu v Modelice je vygenerovaný program simulátoru C++. Pokud bychom vystačili se simulátorem, který se bude následně na počítači klienta vždy instalovat, pak nám program modelu v C++ stačí. Pokud ale chceme využít nové možnosti prostředí .NET umožňující vytvářet pomocí platformy Silverlight webově spustitelné aplikace běžící v internetovém prohlížeči, pak musíme vytvořit nástroj, který bude z Modeliky generovat zdrojový text modelu v C#., což, jak bylo již zmíněno výše, je i náš současný úkol v rámci mezinárodního konsorcia Open Modelica Source Consortium – (viz http://www.ida.liu.se/labs/pelab/modelica/OpenSourceModelicaConsortium.html).

4.4.6 Tělo výukového simulátoru – instalovatelný program nebo webová aplikace

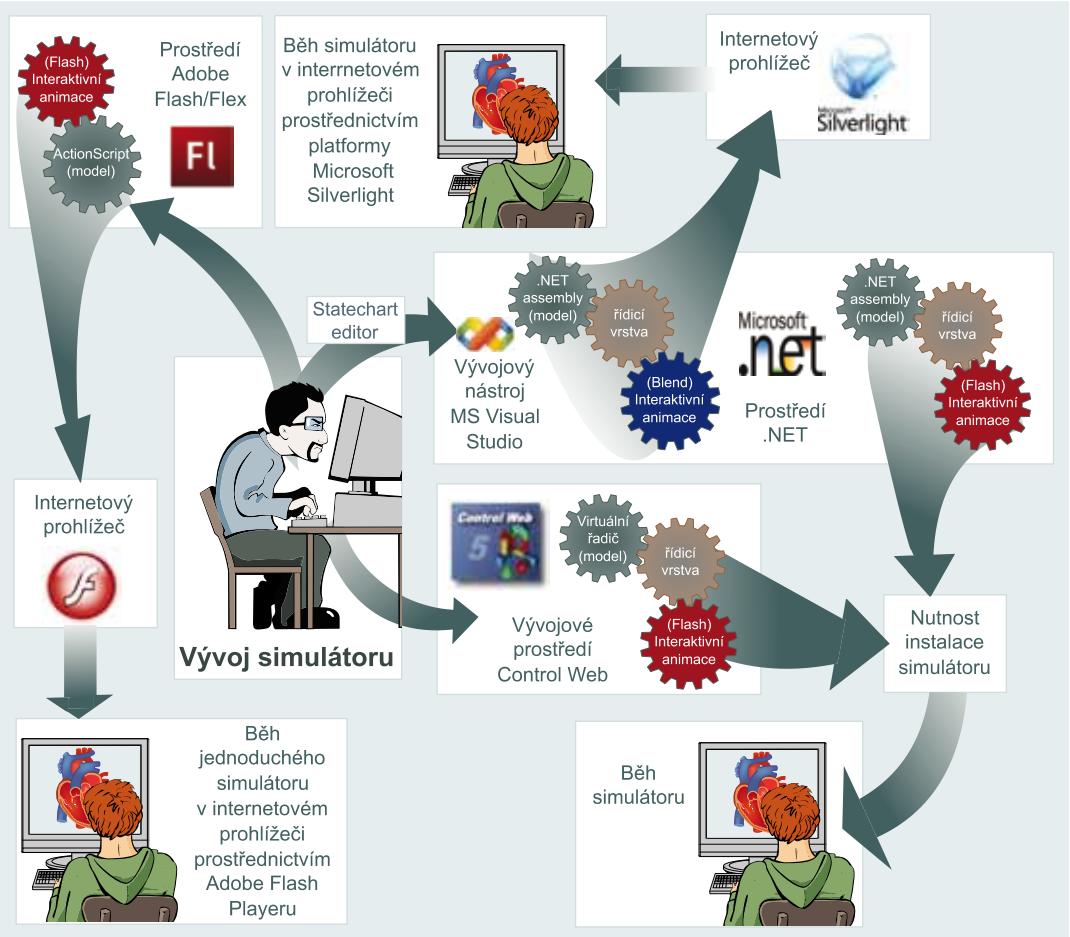
Tvorba výukové simulační aplikace je poměrně náročná programátorská práce, spočívající ve vytvoření simulačního jádra vyvíjené aplikace (pokud toto jádro již nebylo z příslušného simulačního vývojového nástroje automaticky vygenerováno) a jejího propojení s grafickými prvky vizuálního uživatelského rozhraní (viz obr. 91).

Pro výukové aplikace s jednoduchými modely vystačíme s prostředím Flashového přehrávače a jazykem Action Script, v němž naprogramujeme vlastní simulátor a celou aplikaci můžeme spouštět přímo v internetovém prohlížeči.

Pro složitější aplikace to ale nestačí.

Simulátory jsme v minulosti vytvářeli ve vývojovém prostředí ***Control Web*** české firmy Moravské přístroje (viz kapitola 3). Vytvořená aplikace vyžadovala instalaci do počítače uživatele, nebo (u webově šiřitelných aplikací) alespoň instalaci prostředí runtime Control Web.

Platformou pro vývoj simulátorů, kterou využíváme nyní, je platforma Microsoft .NET a programovací prostředí Microsoft Visual Studio .NET, které, poskytuje velké možnosti pro programátorskou práci. Můžeme využít grafické komponenty uživatelského rozhraní, vytvořené v Adobe Flash, které můžeme propojit (přes ActiveX) s jádrem simulátoru, kterým je simulační model ***a grafické komponenty se pak mohou chovat jako loutky řízené simulačním modelem***. Tímto způsobem byla například realizována kapitola našeho Atlasu fyziologie a patofyziologie, věnovaná základních dynamických vlastnostech fy-



Obr. 91 – Programátor je zodpovědný za vývoj vlastního výukového simulátoru. Na základě znalostí struktury identifikovaného simulačního modelu může v prostředí Adobe Flash (nebo pomocí nástroje Adobe Flex) naprogramovat simulační jádro a příslušné uživatelské rozhraní výukového simulátoru, využívajícího vytvořené Flashové animace. Celá výuková aplikace pak může být distribuována prostřednictvím internetu a pomocí Flash Playeru provozována v internetovém prohlížeči. Složitější simulátory jsou však vytvářeny v prostředí .NET (dříve jsme simulátory také vytvářeli ve vývojovém prostředí Control Web). Tvorba simulátorů předpokládá, že programátor propojí animace se simulační komponentou modelu (realizovanou jako .NET assembly). Pro usnadnění programování tohoto propojení (a vytvoření propojovací řídící vrstvy) jsme vyvinuly nástroj Statechart Editor, založený na využití hierarchických stavových automatů. Simulátory využívající flashové animace (propojené se svým okolím přes rozhraní Active X) vyžadují instalaci vytvořené aplikace v počítači. Vytvoříme-li animace v prostředí Microsoft Expression Blend pak můžeme vybudovat aplikaci spustitelnou v internetovém prohlížeči s nainstalovaným doplňkem Silverlight. Tímto způsobem můžeme vytvářet poměrně numericky náročné výukové simulátory distribuované jako webové aplikace spouštěné v internetovém prohlížeči klienta.

ziologických regulačních systémů (obr. 104 na str. 109) - http://physiome.cz/atlas/sim/RegulaceSys/ nebo výukový simulátor přenosu krevních plynů (obr. 106-112 na str. 113-116) - http://physiome.cz/atlas/sim/BloodyMary_cs/.

Nevýhodou tohoto přístupu je nutnost instalace programu (nabízeného přes internetové rozhraní) do počítače klienta. Vyžaduje to ale, aby klient měl příslušná instalační práva na počítači, na němž pracuje. To ale obvykle neplatí v počítačových učebnách, kde bývají počítače chráněny před instalací nevhodného softwaru, a uživatel musí o instalaci výukového programu nejprve požádat příslušného správce.

Proto je vhodné mít možnost ***spouštět a ovládat i složité modely přímo z webového prohlížeče***. Tato cesta je možná, pokud celý simulátor vytvoříme tak, aby byl spustitelný v prostředí – ***Silverlight***, tj. simulační jádro je vytvořeno formě řízeného kódu pro prostředí .NET (ve formě .NET assembly) a grafické komponenty jsou vytvořeny v prostředí Microsoft Expression Blend.

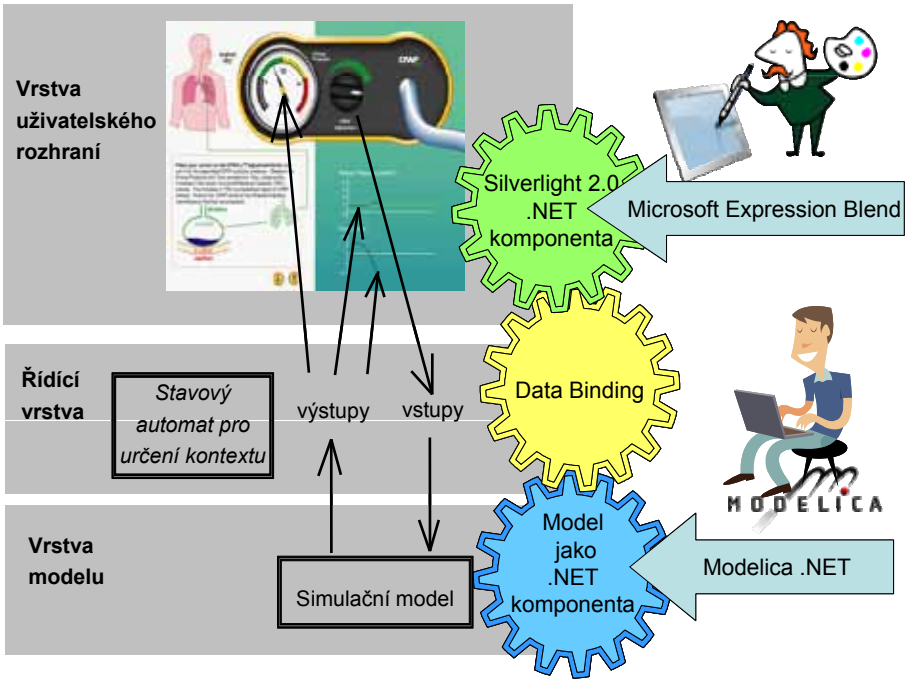
4.4.7 Struktura simulátoru – MVC architektura

V případě složitější architektury může být logika propojení vizuálního uživatelského rozhraní a simulačního modelu poměrně složitá, proto je vhodnější mezi vrstvu vizuálních elementů a vrstvu simulačního modelu vložit řídící vrstvu, která na jednom místě řeší veškerou logiku komunikace uživatelského rozhraní s modelem a kde je ukládán i příslušný kontext. V literatuře (Collins, 1995; Leff & Rayfield, 2007) se hovoří o tzv. MVC architektuře výstavby simulátorů (Model – View – Controller).

Toto uspořádání je nezbytné zejména při složitějších modelech a simulátorech, jejichž uživatelské zobrazení je reprezentováno mnoha virtuálními přístroji na více propojených obrazovkách. Výhody tohoto uspořádání zvláště vyniknou při modifikacích jak modelu, tak i uživatelského rozhraní (obr. 92).

Při návrhu řídící vrstvy, propojující vrstvu simulačního modelu s uživatelským rozhraním, se nám velmi osvědčilo využít propojené stavové automaty (jejichž pomocí je možno zapamatovat příslušný kontext modelu a kontext uživatelského rozhraní).

Vytvořili jsme proto speciální softwarový nástroj, ***Statechart Editor*** pomocí kterého můžeme propojené stavové automaty vizuálně navrhovat, interaktivně testovat jejich chování a automaticky generovat zdrojový kód programu pro prostředí Microsoft .NET (Kofránek, Mateják, & Privitzer, 2009b). Tento nástroj umožňuje zefektivnit programování propojek simulačního modelu s vizuálními objekty uživatelského rozhraní ve výukovém simulátoru.



Obr. 92 – MVC architektura vytvářeného simulátoru.

4.4.8 Propojení platforem pro tvorbu modelů, simulátorů i animací

Při tvorbě simulátorů jsme nuceni pracovat se třemi typy rozdílných softwarových nástrojů:

1. Softwarové nástroje pro tvorbu a odlaďování matematických modelů, které budou podkladem simulátoru – ***Matlab/Simulink*** a v poslední době s akauzálními nástroji využívajícími jazyk ***Modelica***. V tomto prostředí je výhodné a efektivní simulační modely vyvíjet, problematické je ale v tomto prostředí simulátory provozovat.

2. Softwarový ***nástroj pro vývoj vlastního simulátoru*** – zde především využíváme prostředí ***Microsoft Visual Studio .NET***. Dříve jsme také využívali vývojové prostředí ***Control Web***, české firmy Moravské přístroje, zejména proto, že má vynikající možnosti pro rychlé vytváření uživatelského rozhraní simulátoru – toto rozhraní má ale příliš „technicistní" charakter. Jednoduché modely implementujeme v jazyce ActionScript a vystačíme proto pouze s prostředím ***Adobe Flash***, doplněném případně o prostředí Adobe Flex.

3. ***Softwarové nástroje pro tvorbu interaktivní multimediální grafiky*** – uživatelského rozhraní pro simulátory. Zde jsme dlouhodobě využívali nástroj ***Adobe Flash*** (dříve Macromedia Flash). V tomto nástroji je možné vytvářet interaktivní animace, které ale lze zároveň programovat pomocí speciálního programového jazyka ActionScript. Důležité je, že animace mohou (díky výše zmíněné možnosti programování v jazyce ActionScript) softwarově komunikovat se svým okolím prostřednictvím komponenty Active X. Nyní při vývoji grafických aplikací dáváme před nástrojem firmy Adobe přednost vývojovému prostředí ***Microsoft Expression Blend*** jehož pomocí můžeme vytvářet grafické komponenty pro platformu ***Siverlight***.

Protože pro tvorbu simulačních modelů a pro vytváření vlastního simulátoru používáme odlišné vývojové nástroje, museli jsme zajistit dostatečně flexibilní přenos výsledků z jednoho vývojového prostředí do druhého – tj. např. zautomatizovat převod modelu z prostředí Matlab/Simulink do prostředí Visual studia Microsoft .NET (a v minulosti i do prostředí Control Web).

Tyto „propojovací" nástroje umožnily vyvíjet a průběžně aktualizovat matematický model v nejvhodnějším prostředí určeném pro vývoj matematických modelů a zároveň vyvíjet vlastní simulátor ve Visual Studiu .NET (případně ControlWeb), aniž bylo nutné matematický model „ručně" přeprogramovávat.

Umožnily snadnou multidisciplinární spolupráci členů řešitelského týmu – systémových analytiků, vytvářejících matematické modely a programátorů, implementujících simulátor (viz obr. 93).

Na druhé straně to ovšem znamenalo práci ve třech softwarových prostředích a při každé inovaci jednotlivého prostředí bylo nezřídka nutno inovovat příslušné propojovací nástroje.

Simulátory je z hlediska uživatele nejvýhodnější distribuovat přes webové internetové rozhraní, které může sloužit i pro příslušný interaktivní výklad. Webovou výkladovou aplikaci lze snadno propojit s jednoduchými modely implementovanými přímo v Action Scriptu na pozadí flashových animací (tímto způsobem jsme např. vytvořili výukovou aplikaci „mechanické vlastnosti kosterního svalu" - http://www.physiome.cz/atlas/sval/svalCZ/svalCZ.html).

Složitější modely, např. komplexní model přenosu krevních plynů (http://physiome.cz/atlas/sim/BloodyMary_cs/) již před vlastním spuštěním ale vyžadovaly instalaci modelu na počítači klienta (a také i přítomnost platformy .NET, která, pokud nebyla na počítači klienta nainstalována, byla automaticky stažena ze serveru Microsoftu).

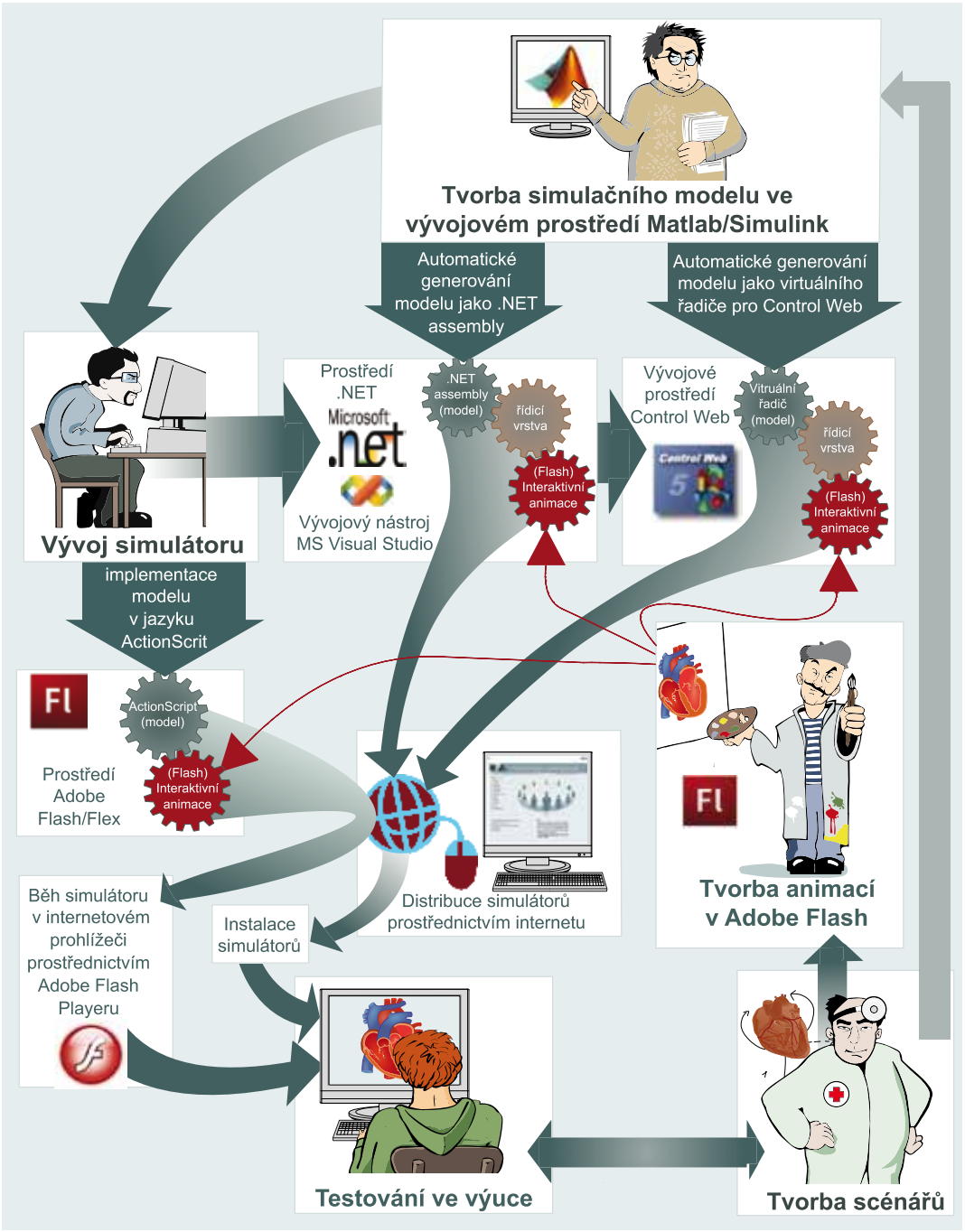
Instalace programů ale vyžaduje mít k počítači příslušná administrátorská práva. Kromě toho se model, který běží jako samostatná aplikace, jen nepřímo se propojuje s webovým rozhraním, kde je realizován interaktivní multimediální výklad.

Tento problém řeší naše nová technologie tvorby simulátorů, využívající platformu Silverlight (viz obr. 94). Grafické prvky jsou vytvářeny v prostředí Microsoft Expression Blend.

Simulační jádro ale vyžaduje být realizováno jako řízený kód v prostředí .NET – to by měla zajistit naše současně vyvíjená aplikace Modelica .NET generující kód modelu v C#.

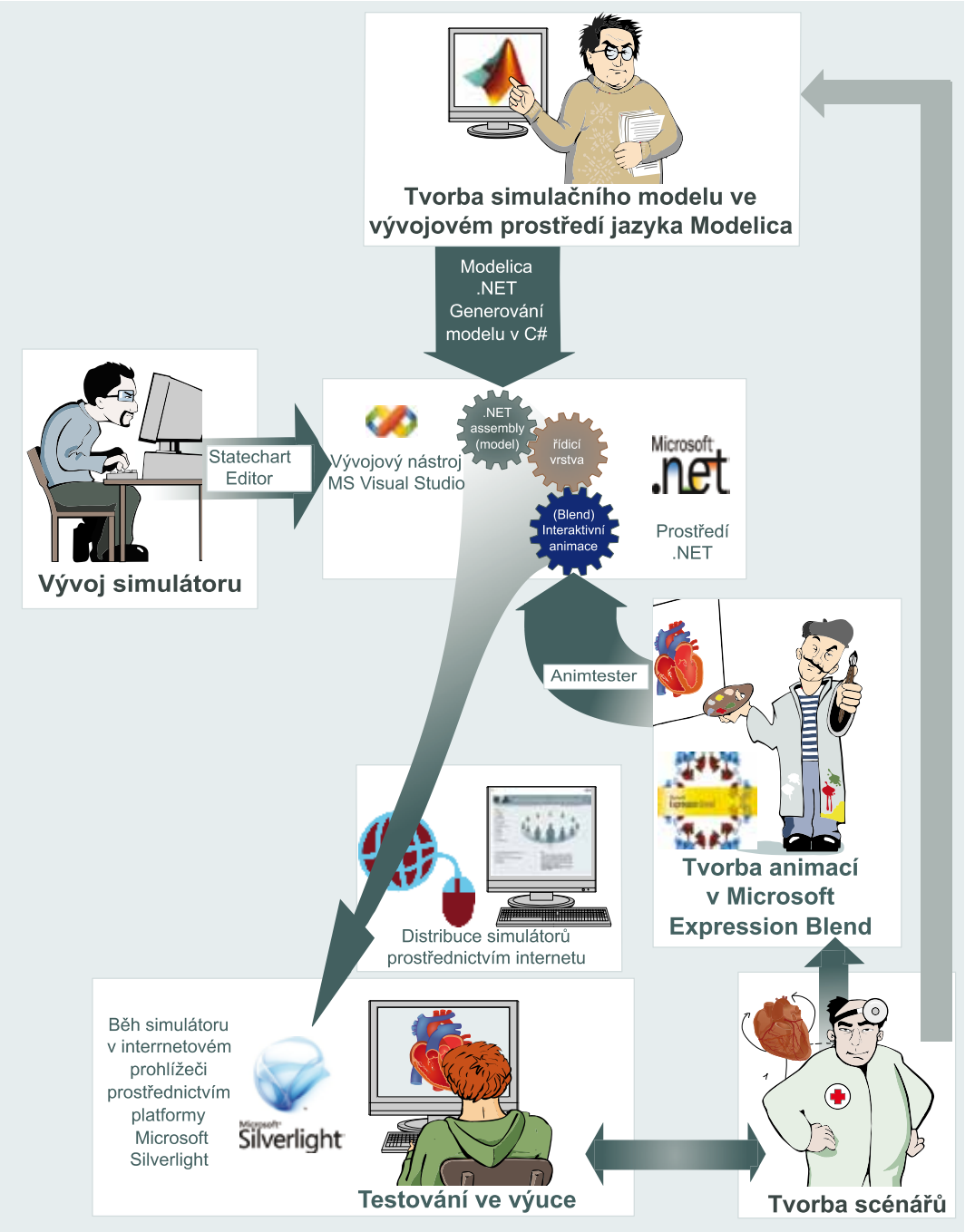
Pro návrh vnitřní logiky aplikace používáme hierarchické stavové automaty (jejichž pomocí je možno zapamatovat příslušný kontext modelu a kontext uživatelského rozhraní). Námi vyvinuté vizuální prostředí (Statecharts editor) umožňuje graficky automaty navrhnout, vygenerovat jejich kód a také je ladit.

Výhodou je že jak grafické interaktivní prvky tak i simulační jádro jsou vytvářeny na jedné platformě - odpadá tedy nutnost složitého přemosťování prostředí .NET a Adobe FLash přes ActiveX komponenty.



Obr. 93 – Původní řešení kreativního propojení nástrojů a aplikací pro tvorbu simulátorů a výukových programů využívajících simulační hry. Základem e-learningového programu je kvalitní scénář, vytvořený zkušeným pedagogem. Tvorba animovaných obrázků je odpovědnost výtvarníků, kteří vytvářejí interaktivní animace v prostředí Adobe Flash. Jádrem simulátorů je simulační model, vytvářený v prostředí speciálních vývojových nástrojů, určených pro tvorbu simulačních modelů. Dlouho jsme zde využívali prostředí Matlab/Simulink od firmy Matworks. Vývoj simulátoru je náročná programátorská práce, pro jejíž usnadnění jsme vyvinuli speciální programy, usnadňující automatický převod vytvořeného simulačního modelu z prostředí Matlab/Simulink do prostředí Control Web a do prostředí Microsoft .NET.

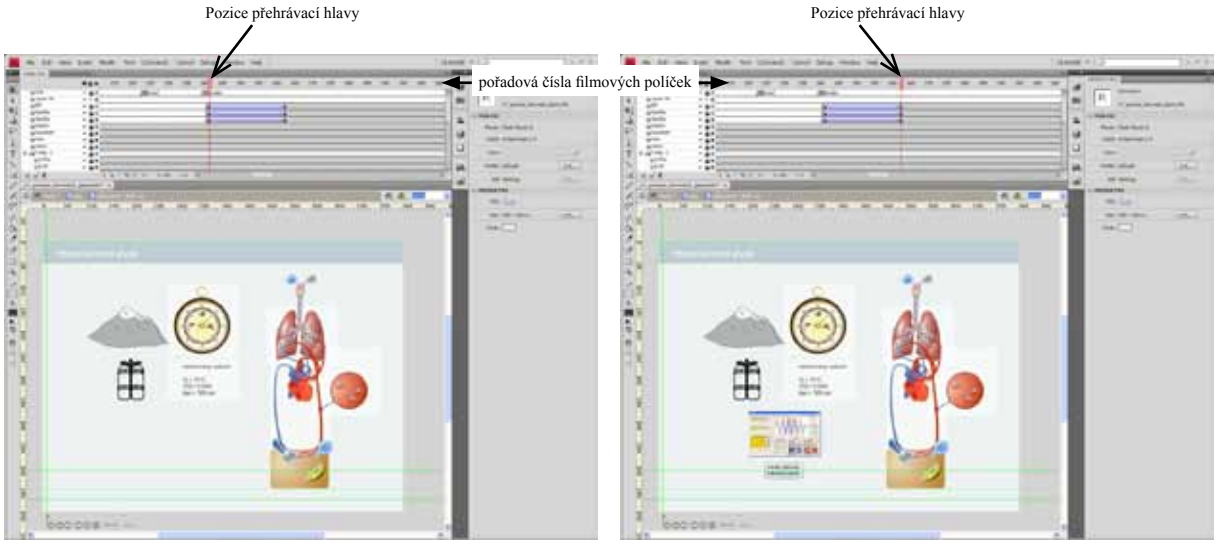
Simulátor být snadno kombinován s výkladovou kapitolou. Výsledná aplikace (jak simulátor, tak i výkladová kapitola) může být realizována jako webová aplikace spustitelná přímo v internetovém prohlížeči, bez nutnosti její instalace na počítači klineta. Může běžet v různých operačních systémech - jedinou podmínkou je instalovaný plugin Silverlight v internetovém prohlížeči.



Obr. 94 – Nové řešení kreativního propojení nástrojů a aplikací pro tvorbu simulátorů a výukových programů využívajících simulační hry. Základem e-learningového programu nadále zůstává kvalitní scénář, vytvořený zkušeným pedagogem. Tvorba animovaných obrázků je odpovědnost výtvarníků, kteří vytvářejí interaktivní animacev prostředí Expression Blend. Pro vytváření a testování animací, které budou posléze řízeny simulačním modelem výtvarník využívá námi vyvinutý softwarový nástroj Animtester. Jádrem simulátorů je simulační model, vytvářený ve vývojovém prostředí simulačního jazyka Modelica. V rámci Open Modelica Source Consortia vytváříme nástroj, který bude z Modeliky generovat zdrojový text modelu v C#. To umožní z modelu vygenerovat .NET komponentu pro výslednou aplikaci v platformě Silverlight, umožňující distribuovat simulátor jako webovou aplikaci běžící přímo internetovém prohlížeči (i na počítačích s různými operačními systémy).

4.4.9 Zabalení simulačních her do multimediálního výkladu

Simulátor bez výkladové části vyžaduje zkušeného pedagoga při jeho využívání. Simulátory je proto vhodné kombinovat s výkladovými lekcemi. Velký pedagogický efekt mají interaktivní výukové progra-



Obr. 95 – Animace, vytvářené v prostředí Adobe Flash se vytvářejí po jednotlivých políčkách jako v kresleném filmu. Při nastavené rychlosti přehrávání časová vizualizace jednotlivých animací závisí na tom, až se přehrávací hlava dostane k příslušnému klíčovému snímku.

my, dosažitelné po internetu, které kombinují výukový text, doprovázený kreslenými animovanými obrázky se simulační hrou, ozřejmující vykládanou látku pomocí modelu ve virtuální realitě.

V naší technologii jsou simulační hry součástí e-learningových multimediálních výukových lekcí, jejichž podkladem je scénář vytvořený zkušeným pedagogem. Pedagog navrhuje vysvětlující text a s textem propojené doprovodné obrázky a animace.

Animace jsou vytvářeny výtvarníkem v úzké spolupráci s pedagogem v prostředí Adobe Flash pro Flash Player v internetovém prohlížeči nebo (v novější technologii) v prostředí Microsoft Expression Blend pro platformu Siverlight.

Text je poté namluven a synchronizován se spouštěním jednotlivých animací a s odkazy na simulační hry. Jednotlivé komponenty jsou kompletovány do výukových lekcí.

Vytvořit ***synchronizaci animací se zvukem*** v prostředí ***Adobe Flash*** není ovšem zcela jednoduchý problém.

Protože se animace v Adobe Flash se vytvářejí jako v kresleném filmu – po jednotlivých políčkách (viz obr. 95), pak časová posloupnost vizualizace animací závisí na tom, kdy se „přehrávací hlava“ (při nastavené rychlosti přehrávání) dostane k příslušnému klíčovému snímku.

Do vrstvy filmových políček je možné uložit i zvukovou stopu. Pro synchronizaci přehrávání animací se zvukem je nutné zajistit (pomocí příkazu v jazyce ActionScript), aby se vložený filmový klip přehrávající danou animaci spustil ve správném v okamžiku, kdy se přehrávací hlava dostane do příslušné pozice.

Pro usnadnění této synchronizace jsme vytvořili speciální nástroj ***PlayDirector*** jako knihovní prvek vkládaný do vytvářeného flashového filmového klipu (obr. 96). Při jeho přehrávání ve FlashPlayeru se dají podle zvukové stopy interaktivně nastavit příslušné zarážky, podle kterých se pak vygenerují data pro vložený skript, který zajistí spouštění příslušných animací v požadovaném čase.

Pro kompletaci multimediálních výukových lekcí vytvořených v prostředí Adobe Flash jsme pak využívali vývojový nástroj Adobe Presenter, dodávaný jako softwarového prostředí serveru Adobe Connect (nyní možné Adobe Presenter zakoupit i separátně – viz http://www.adobe.com/products/presenter/).

Synchronizace animací se zvukem je ovšem v prostředí ***Silverlight*** mnohem jednodušší. Protože animační přístup v Silverlightu je založen na animaci pomocí změny některých vlastností grafických objektů v čase (oproti systému založeném na snímcích v Adobe Flash) je synchronizace animací se zvukovou stopou přímočará, pro tuto synchronizaci nám vývojové prostředí ***Microsoft Expression Blend*** zcela postačí a není zapotřebí vytvářet nový pomocný nástroj.

Obr. 96 – Nástroj PlayDirector slouží k přesnému nastavení synchronizace zvukové stopy s jednotlivými animacemi. Nástroj se ve vývojovém prostředí Adobe Flash přidá do vytvářené animace jako její komponenta. Při následném přehrání ve Flash Playeru nástroj umožní pomocí posuvu zarážek přesně nastavit čas přehrání jednotlivých animací.

4.5. Simulační hry na Webu

Jedním z konkrétních výsledků našeho úsilí je internetový ***Atlas fyziologie a patofyziologie,*** (Kofránek, a další, 2007-2009) koncipovaný jako multimediální výuková pomůcka, která názornou cestou prostřednictvím Internetu s využitím simulačních modelů by měla pomoci vysvětlit funkci jednotlivých fyziologických systémů, příčiny a projevy jejich poruch – http://physiome.cz/atlas. (Kofránek, a další, 2007; Kofránek, Mateják, Matoušek, Privitzer, Tribula, & Vacek, 2008; Kofránek, Privitzer, Matoušek, Vacek, & Tribula, 2009)

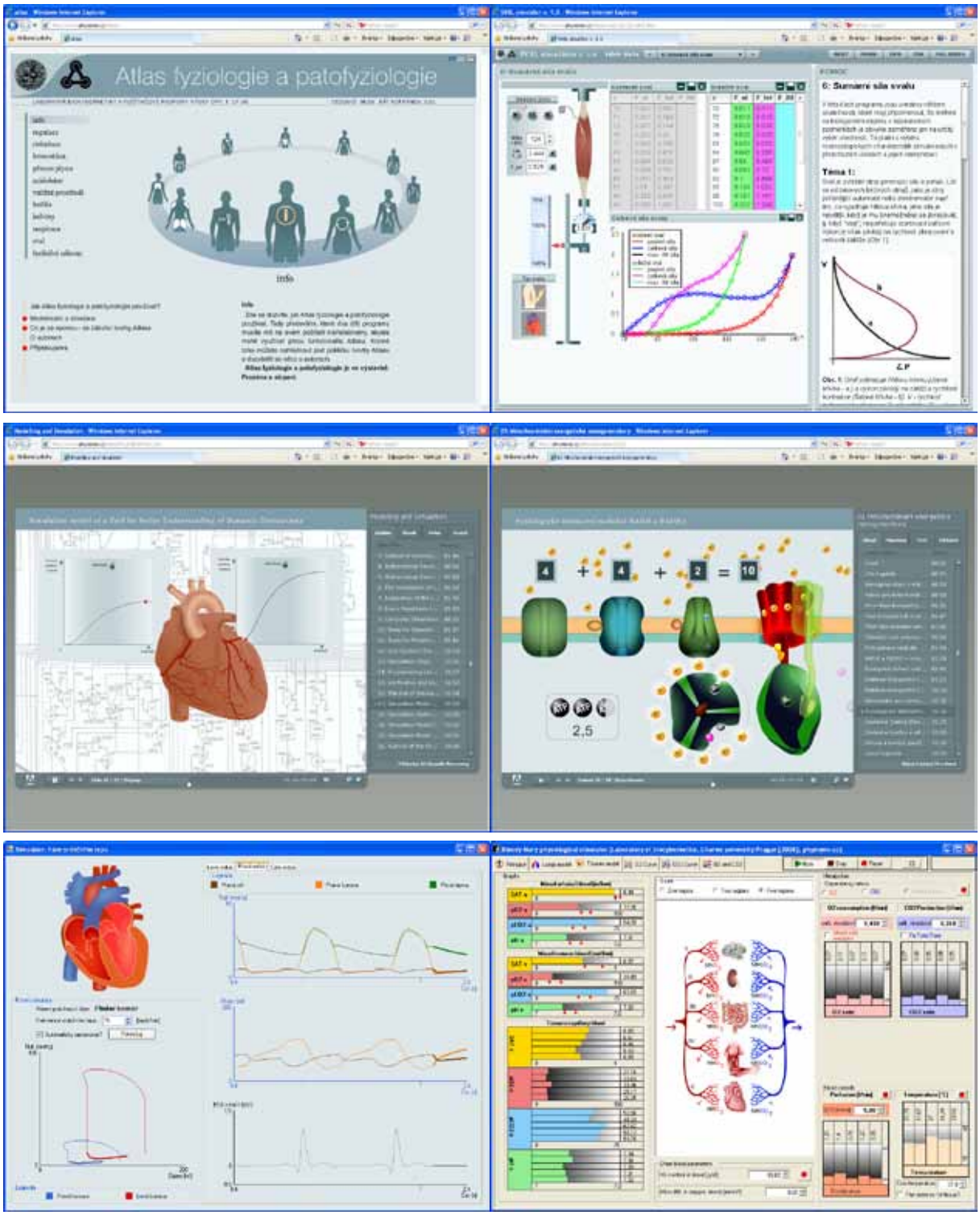
Atlas je součástí sítě MEFANET (MEdical FAculties NETwork), soustřeďující elektronické učební texty lékařských fakult ČR a SR (http://www.mefanet.cz/). Nyní se připravuje začlenění anglicky lokalizovaných součástí atlasu do „Teaching Resources“ Americké fyziologické společnosti http://www.apsarchive.org/)

Projekt atlasu je otevřený – jeho výsledky jsou volně přístupné na internetu. Je vytvářen v české a anglické verzi. Je součástí elektronických výukových pomůcek v rámci sítě Veškeré výukové texty, interaktivní animace a simulační modely včetně jejich zdrojových kódů jsou k dispozici všem zájemcům. Při jeho dalším vývoji uvítáme spolupráci se všemi, kdo se budou chtít podílet na jeho postupném budování.

4.5.1 Simulační modely jako „živé“ interaktivní ilustrace

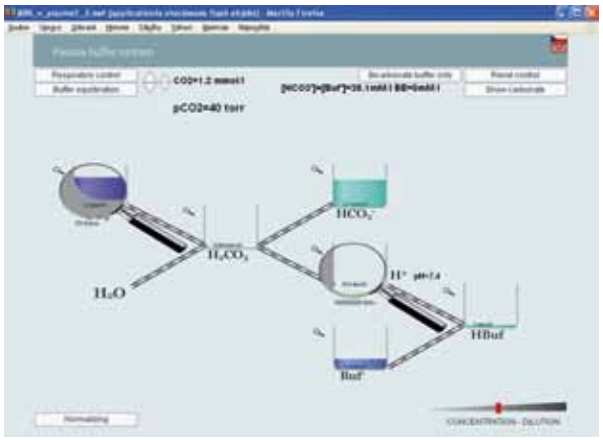
Atlas kombinuje interaktivní výkladové kapitoly a simulační hry s modely fyziologických systémů (viz obr. 97). Při tvorbě uživatelského rozhraní modelů, využívaných jako podklad pro simulační hry, připomíná spíše animované obrázky z tištěného Atlasu fyziologie (Silbernagl & Despopoulos, 2003, české vydání 2004) nebo Atlasu patofyziologie (Silbernagl & Lang, 1998, české vydání 2001), než abstraktní regulační schémata využívaná ve výuce bioinženýrů. Na rozdíl od tištěných ilustrací, jsou ale obrázky tvořící uživatelské rozhraní multimediálních simulátorů ***„živé“*** a ***interaktivní*** – změny ***proměnných simulačního modelu*** se projeví ***změnou obrázku***. Pomocí takto koncipovaných interaktivních ilustrací je možno realizovat simulační hry, které lépe než statický obrázek nebo i prostá animace pomohou vysvětlit dynamické souvislosti ve fyziologických systémech a napomoci především k pochopení příčinných souvislostí v rozvoji patogenez nejrůznějších chorob.

Jako příklad „obrázkového“ uživatelského rozhraní výukové simulační hry můžeme uvést ***model acidobazické rovnováhy plazmy***, kde jsou pufrační systémy v uživatelském rozhraní znázorněny jako propojené nádoby zobrazující kompartmenty jednotlivých látek. Model je součástí atlasu a je dostupný v české i anglické verzi na adrese http://www.physiome.cz/atlas/acidobaze/02/ABR_v_plazme1_2.swf.

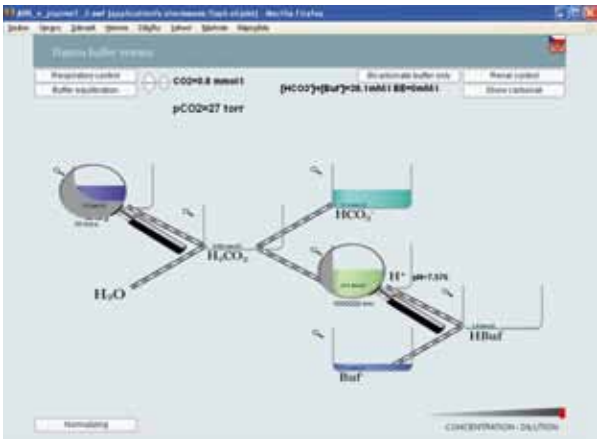


Obr. 97 – Atlas fyziologie a patofyziologie kombinuje interaktivní výklad se zvukovým a animačním doprovodem a simulační hry. Je vytvářen v české (a postupně i v anglické) verzi. Je volně dostupný na internetové adrese www.physiome.cz/atlas.

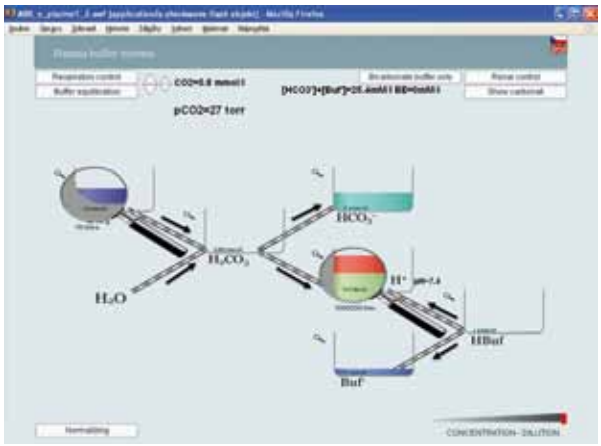
Výška „hladiny“ v těchto nádobách reprezentuje koncentrace. Chemické reakce jsou znázorněny jako „přelévání tekutiny“ mezi nádobami s jednotlivými složkami pufračních systémů. Do těchto nádob mohou „přitékat“ nebo „odtékat“ látky z/do metabolismu, respiračního systému nebo ledvin. Pomocí simulačních her s tímto modelem můžeme názorně vysvětlit vývoj různých poruch acidobazické rovnováhy. Na obr. 99-101 je uvedeno využití tohoto simulátoru v simulační hře vysvětlující ***patogenezi***



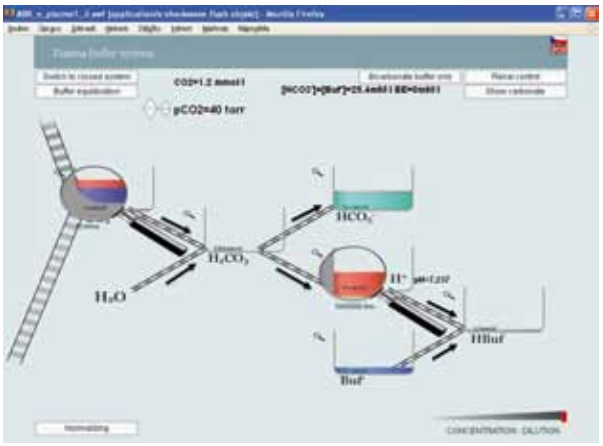
Obr. 98 – Anglická verze Interaktivního výukového modelu pufračního systému plazmy. Výšky hladin znázorňují hodnoty koncentrací. Počáteční stav.



Obr. 99 – Ovládacím šoupátkem vyvoláme v modelu diluci, hladiny všech látek, včetně koncentrace CO_2 a koncentrace vodíkových iontů, se sníží.



Obr. 100 – Stiskem tlačítka „Pufrační ekvilibrace“ zapojíme ustavení chemické rovnováhy v pufračním systému a pH plazmy se vrátí na hodnotu 7.4.



Obr. 102 – Respirace zvýší (původně po diluci sníženou) hodnotu koncentrace CO_2 na původní hladinu 1,2 mmol/l. Po ustavení nové chemické rovnováhy se koncentrace vodíkových iontů zvýší a pH plazmy se sníží.

diluční acidózy.

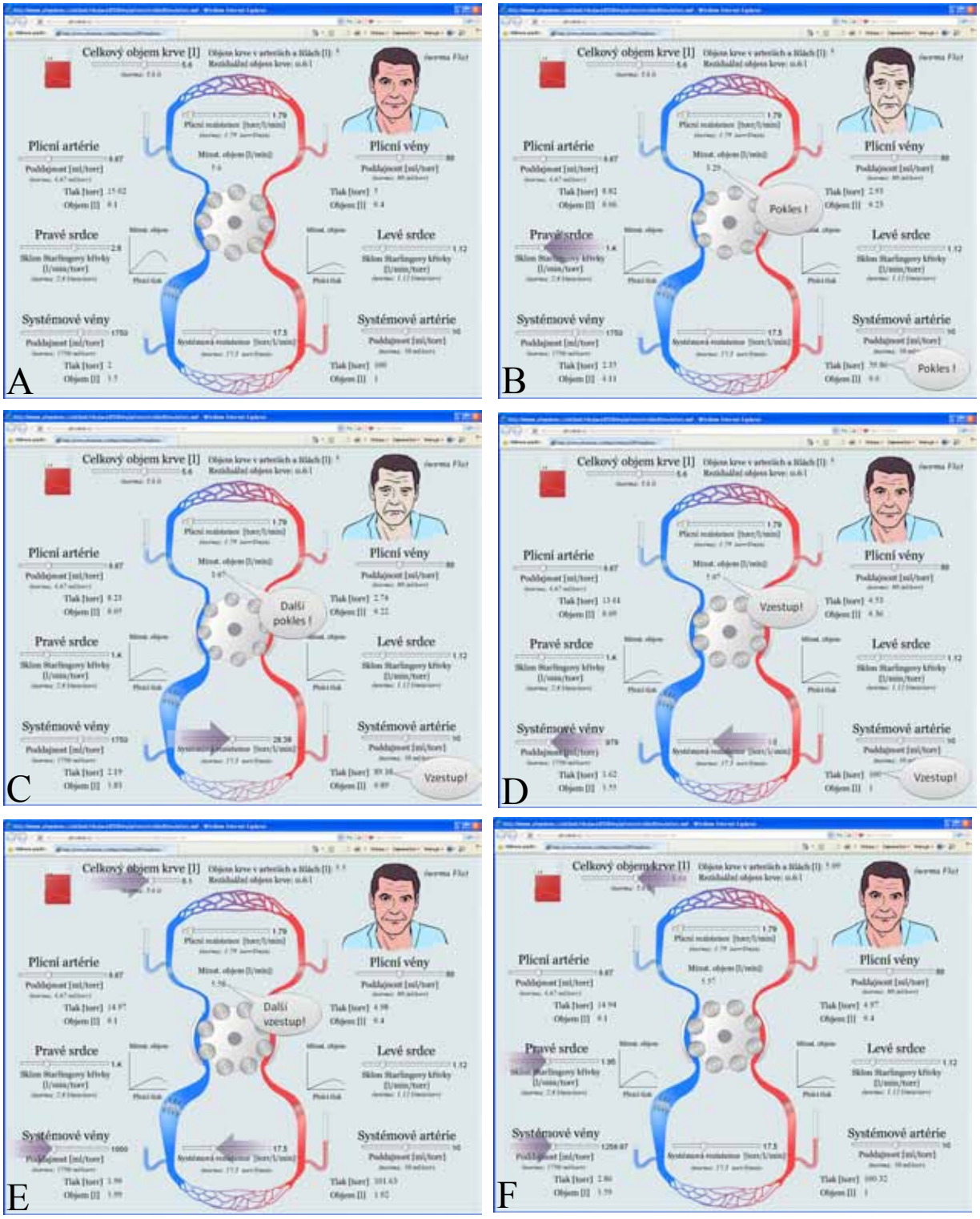
Stiskem tlačítka „Vše normalizuj“ dostaneme systém do výchozího normálního fyziologického stavu (obr. 98). Zředění jednotlivých komponent pufrů se znázorní jako rozšíření příslušných nádobek – protože množství komponent v nádobkách zůstává stejné, hladina (reprezentující koncentraci) se sníží. Sníží se i hladina vodíkových iontů (obr. 99). Stiskem tlačítka „Pufrační ekvilibrace“ spustíme chemické reakce v pufračních systémech vizualizované jako „přelévání“ jednotlivých komponent. Po proběhlé disociaci kyseliny uhličité a slabých pufračních kyselin (v modelu označených jako HBUF – v realitě reprezentovaných především albuminem a fosfáty), se hladina vodíkových iontů ustaví zpět na původní hodnotu (obr. 100). Nicméně hladina kyseliny uhličité, stejně jako hladina CO_2 zůstává díky diluci snížená. V organismu však respirace udržuje hladinu CO_2 v arteriální krvi na stálé úrovni (dané především hodnotou alveolární ventilace). Stiskem tlačítka „Respirační regulace“ (resp. „Respiratory control“ v anglické verzi) se hladina CO_2 zvýší na původní hodnotu před dilucí. Stiskem tlačítka „Pufrační ekvilibrace“ („Buffer equilibration“ v anglické verzi) proběhne chemická reakce která ustaví novou chemickou rovnováhu se zvýšenou koncentrací vodíkových iontů (obr. 101).

4.5.2 Princip „ceteris paribus“ ve výukových simulačních hrách

Z didaktického hlediska je nutné při výkladu vždy postupovat od jednoduchého ke složitějšímu. Podle tohoto principu je proto vhodné při výkladu využívat nejprve jednodušší agregované modely (s několi-

ka proměnnými), s jejich pomocí vysvětlit základní principy, a poté model (a popisovanou fyziologickou realitu) postupně zesložiťovat.

Výukové simulační hry, které jsou součástí atlasu, nemusí mít vždy podklad ve velmi složitém a výpočetně náročném modelu se stovkami proměnných – ***i jednoduchý interaktivní model může být dobrým pomocníkem pro vysvětlení patogenetických řetězců rozvoje nejrůznějších patologických stavů.***



Obr. 101 – Využití jednoduchého modelu (neřízeného) cirkulačního systému pro vysvětlení patogeneze pravostranného cirkulačního selhání. A - počáteční stav. B – snížení kontraktility pravého srdce. C – periferní vasokonstrikce. D – venokonstrikce velkých žil. E – retence objemu extracelulární tekutiny F – podání kardiotonik a diuiretik.

Jak jsme se již přesvědčili při využívání simulátoru Golem ve výuce, je didakticky velmi účinné v modelu nejprve ***rozpojit regulační smyčky*** a v simulační hře umožnit studentům ***studovat reakce zvoleného fyziologického subsystému na změny vstupních veličin*** (které jsou ovšem v reálném organismu samy regulovány). Nejprve sledujeme dynamiku chování při postupných změnách pouze jediného vstupu, zatímco jiné vstupy jsou nastaveny na zvolené konstantní hodnoty (tzv. ***princip „ceteris paribus“).***

To umožní studentům ***lépe pochopit význam jednotlivých regulačních okruhů*** a studovat vliv (rozpojených a zprvu ručně řízených) regulačních vazeb na chování organismu při nejrůznějších patologických poruchách a reakcích na příslušnou terapii. Podle našich zkušeností právě tento přístup vede k lepšímu pochopení významu jednotlivých regulačních smyček a porozumění jejich úlohy v patogeneze nejrůznějších onemocnění a chápání patofyziologických principů příslušných léčebných zásahů.

Tak například při výkladu fyziologie a patofyziologie oběhu není vhodné začínat simulační hrou s modelem, jehož složitost je zhruba na stejné nebo vyšší úrovni jako v úvodu zmíněný Guytonův model cirkulace (obr. 1 na straně 3). Je vhodnější zpočátku zvolit jednoduchý agregovaný model, na němž je možné demonstrovat základní principy struktury a chování krevního oběhu a možnosti regulačního ovlivnění. Nejjednodušší ***model cirkulačního systému s rozpojenými regulačními vazbami***, jako součást našeho atlasu, je přístupný na adrese: http://www.physiome.cz/atlas/cirkulace/05/SimpleUncontrolledSimulation.swf.

Jeho ovládání (obr. 102) je velmi jednoduché a slouží především k ujasnění základních vztahů mezi jednotlivými proměnnými oběhového systému (tj. tlaky a průtoky v malém a velkém oběhu) a základními veličinami, které tlaky a průtoky ovlivňují (a samy jsou ale neurohumorálně řízené).

Jsou to:

- periferní odpory (systémový a plicní),
- čerpací funkce pravé a levé komory – v modelu realizovaná tím nejjednodušším způsobem jako sklon Starlingovy křivky (vyjadřující závislost minutového objemu srdečního na plnících tlacích v pravé a levé síni),
- poddajnosti artérií a žil (vyjadřující závislost tlaku na náplni cév),
- celkový objem cirkulující krve.

Organismus tyto veličiny reguluje (rezistence je řízena nervovou a humorální regulací, změna frekvence a inotropie myokardu mění tvar Starlingovy křivky, venózní tonus velkých žil mění jejich poddajnost a objem cirkulující krve je ovlivňován především činností ledvin, renin-angiotenzinovou regulací aj.). V agregovaném modelu jsou však tyto veličiny vstupními (tj. neregulovanými) veličinami – cílem simulační hry s modelem je ozřejmit si význam těchto veličin pro řízení tlaků, průtoků a distribuci objemu krve mezi jednotlivými částmi krevního řečiště.

Simulační hrou s tímto modelem je možné studentům názorně vysvětlit jakým způsobem se uplatňuje regulace základních veličin oběhového systému v patogeneze různých poruch oběhového systému.

Jako příklad uveďme ***simulační hru rozvoje pravostranného cirkulačního selhání***.

Stiskem tlačítka „norma Vše¨ uvedeme model do výchozího normálního fyziologického stavu (obr. 102 A). Pak šoupátkem snížíme sklon Starlingovy křivky v pravém srdci – modelujeme tím snížení stažlivosti pravého srdce při akutním pravostranném oběhovém selhání (obr. 102 B). Minutový objem srdeční poklesne na hodnotu 3,29 l/min, střední systémový arteriální tlak klesne na 59,86 torr.

Sympaticus reaguje na pokles krevního tlaku výraznou vasokonstrikcí především ve splanchnické oblasti, s účelem zachovat perfúzi koronárních cév. Posunem šoupátka doprava proto zvýšíme periferní systémovou rezistenci na 28.36 torr/l/min (obr. 102 C) – střední arteriální tlak stoupne na hodnotu 89, 18 torr, minutový objem srdeční však dále poklesne z 3,29 l/min na 3,07 l/min!

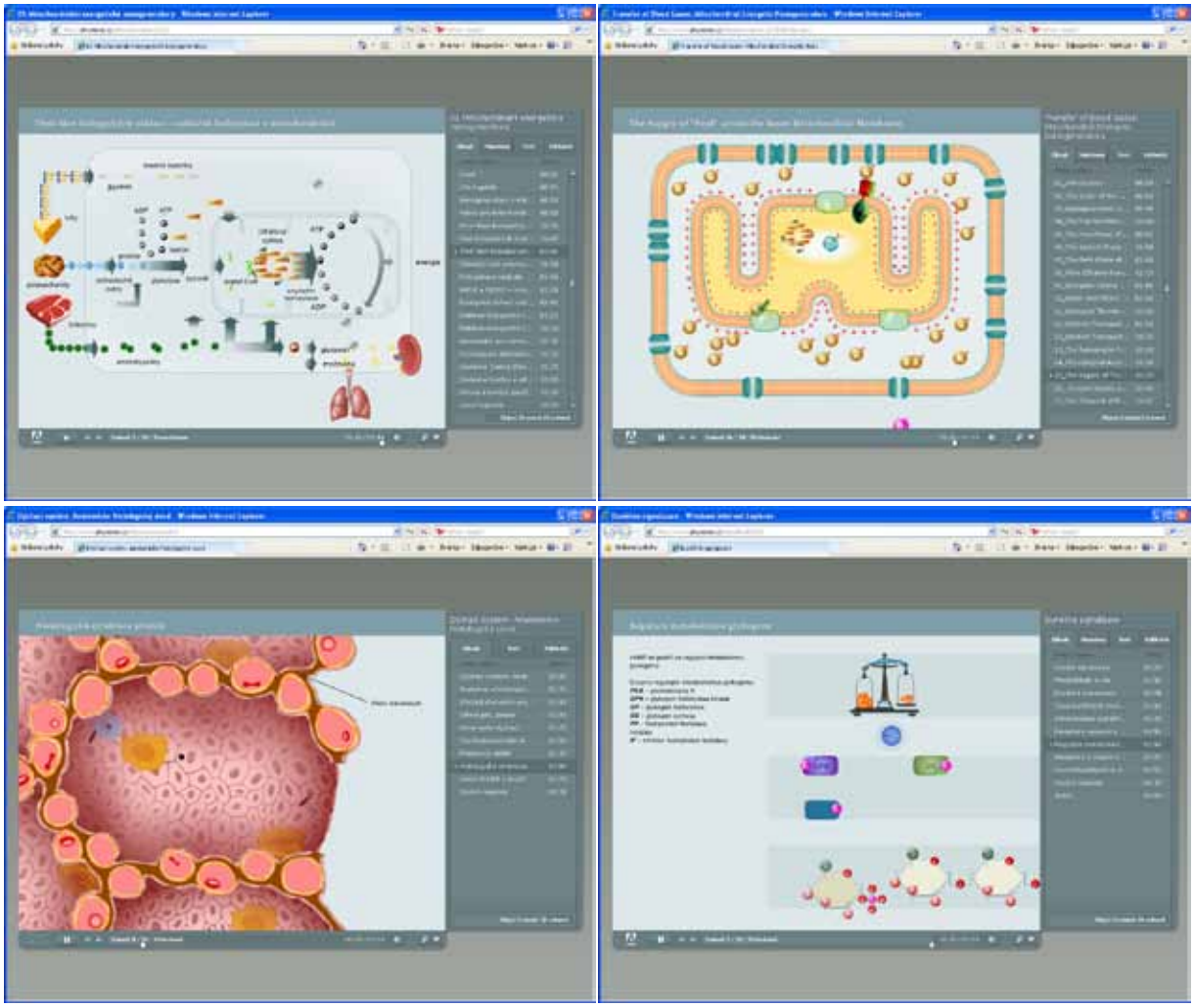
Sympaticus nemá ovšem vliv jen na vasokonstrikci arteriol a následné zvýšení periferního odporu. Zvyšuje též tonus velkých žil, což se projeví tím, že při stejné náplni krve v nich stoupne tlak – zvýšení venózního tonu se dá modelovat snížením poddajnosti systémových vén (obr. 102 D). Snížení poddajnosti z 1750 ml/torr na hodnotu 979 ml/torr zvedne tlak ve velkých systémových žilách, a tím i plnící tlak v pravé síni, což vede ke zvýšení minutového objemu srdečního (zároveň ale zvýšení venózního

tlaku vede k vyšší filtraci v kapilárách a ke vzniku edémů). Střední arteriální tlak stoupne a k jejímu udržení na normální hodnotě 100 torr není nutno udržovat rezistenci v systémovém řečišti na příliš vysoké hodnotě. Šoupátkem ji proto snížíme z hodnoty 28,36 torr/l/min na hodnotu 19 torr/l/min. Minutový objem srdeční také stoupl z 3,29 l/min k hodnotě 5,07 l/min.

V simulační hře lze pokračovat dále tím, že ukážeme význam zvýšení celkového objemu krve, který nastane v důsledku aktivace renin-angiotenzin-aldosteronové smyčky (obr. 102 E). Pokud šoupátkem zvýšíme objem cirkulující krve o 500 ml tj. z 5,6 l na 6,1, je možné na modelu ukázat, že k udržení normálního minutového objemu srdečního a normálního středního arteriálního tlaku je možno dále snížit periferní rezistenci až na normu a zároveň snížit venokonstrikci (tj. zvýšit poddajnost velkých žil – postačí ji udržovat na 100 ml/torr).

V simulačním experimentu lze dále demonstrovat i vliv terapie: léčení kardiotoniky zobrazíme zvýšením sklonu Starlingovy křivky a podání diuretik simulujeme snížením zvýšeného objemu cirkulující krve – důsledkem je snížení tlaku ve velkých žilách s následným snížením otoků (obr. 102 F).

Zmíněný příklad je ukázkou toho, jak ***simulační hry*** i s jednoduchým modelem ***přispějí k lepšímu pochopení významu uplatnění jednotlivých regulačních okruhů*** v patogeneze nejrůznějších patologických stavů a v následných terapeutických zásazích.



Obr. 103 – Ozvučené interaktivní přednášky ve výkladové části Atlasu fyziologie a patofyziologie. Každý výklad je provázen animovanými obrázky synchronizovanými s výkladem. Výklad lze v libovolném okamžiku přerušit a podrobně si prohlížet doprovodnou animaci. Pomocí jezdce v dolní části přehrávače, lze také výklad včetně synchronizovaných animací, vracet zpět.

4.5.3 Atlas jako webová interaktivní učebnice

Atlas fyziologie a patofyziologie je v současné době koncipovaný jako webová aplikace spustitelná v internetovém prohlížeči (předpokladem je v prohlížeči nainstalovaný přehrávač flashových animací). Některé simulační modely vyžadují mít v počítači nainstalovaný Microsoft .NET framework (pokud tato součást není nainstalovaná, nabízí se její instalace před instalováním prvního simulátoru, který .NET vyžaduje).

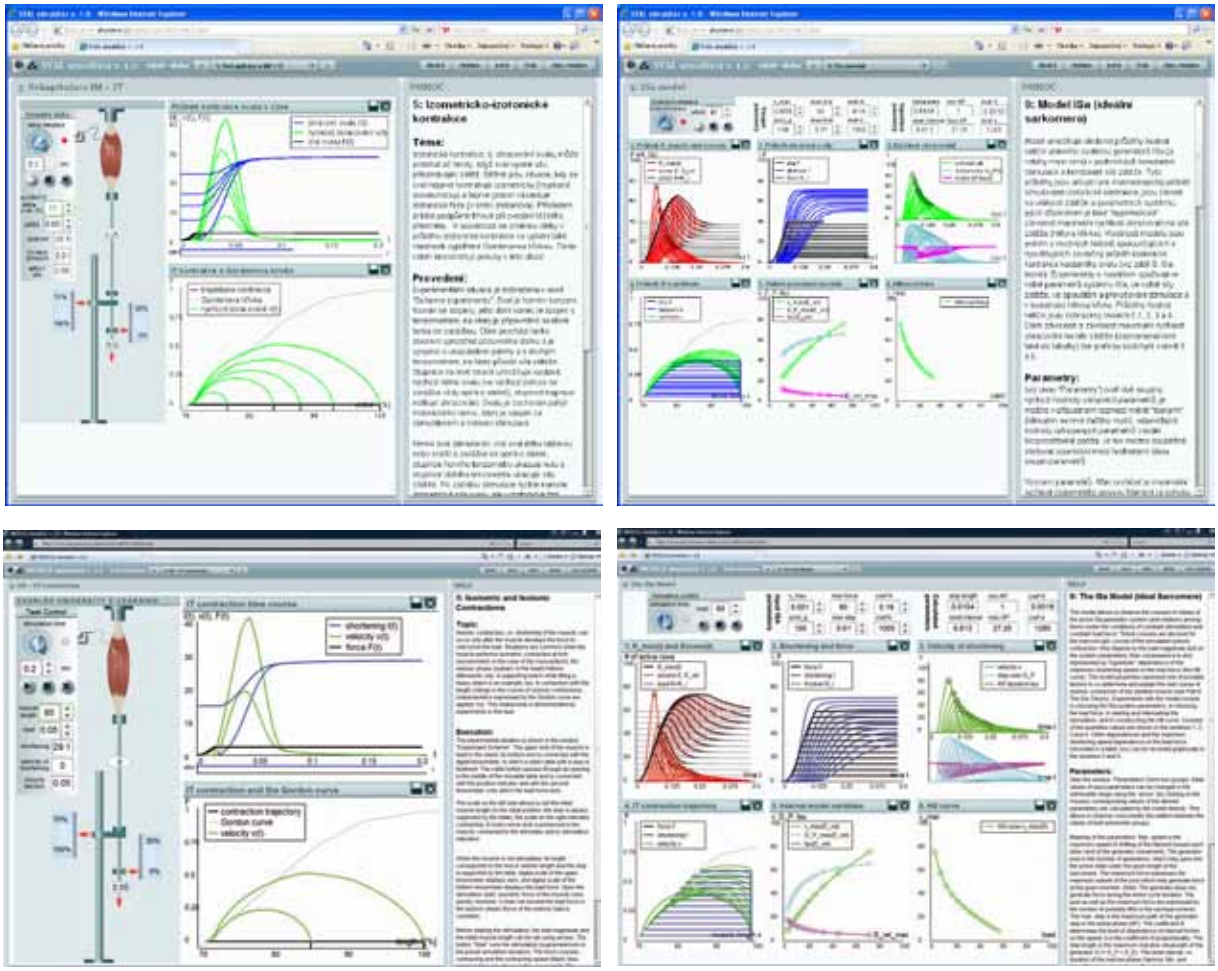
Výkladové kapitoly Atlasu jsou koncipované jako ozvučené přednášky provázené interaktivními multimediálními obrázky (obr. 103). Každá animace je přesně synchronizovaná s výkladovým textem.

Internetový Atlas fyziologie a patofyziologie je ale mnohem více než pouhý ozvučený animovaný výklad.

Základem didaktické efektivity je ***výklad provázený simulační hrou***. Simulační modely, které jsou součástí atlasu, jsou realizovány jako flashové aplikace a nemusí se zvlášť instalovat (jako např. simulátory na obr. 98-101), nebo (u složitějších modelů) se vyžaduje jejich samostatná instalace přímo z internetového prohlížeče.

Složitější modely vyžadují poněkud komplikovanější ovládání – důležitý je proto vhodný scénář, podle kterého se model dá v simulační hře využít jako výuková pomůcka pro vysvětlení komplikovanějších fyziologických vztahů.

Některé simulátory mají v sobě kombinovaný model i výkladovou část – příkladem je ***simulátor mechanických vlastností svalu*** (Wünsch, Kripner, Kofránek & Andrlík, 2004) vytvořený původně v prostředí Control Web a později celý přeprogramován jako čistě flashová aplikace v anglické i české verzi (obr. 104).

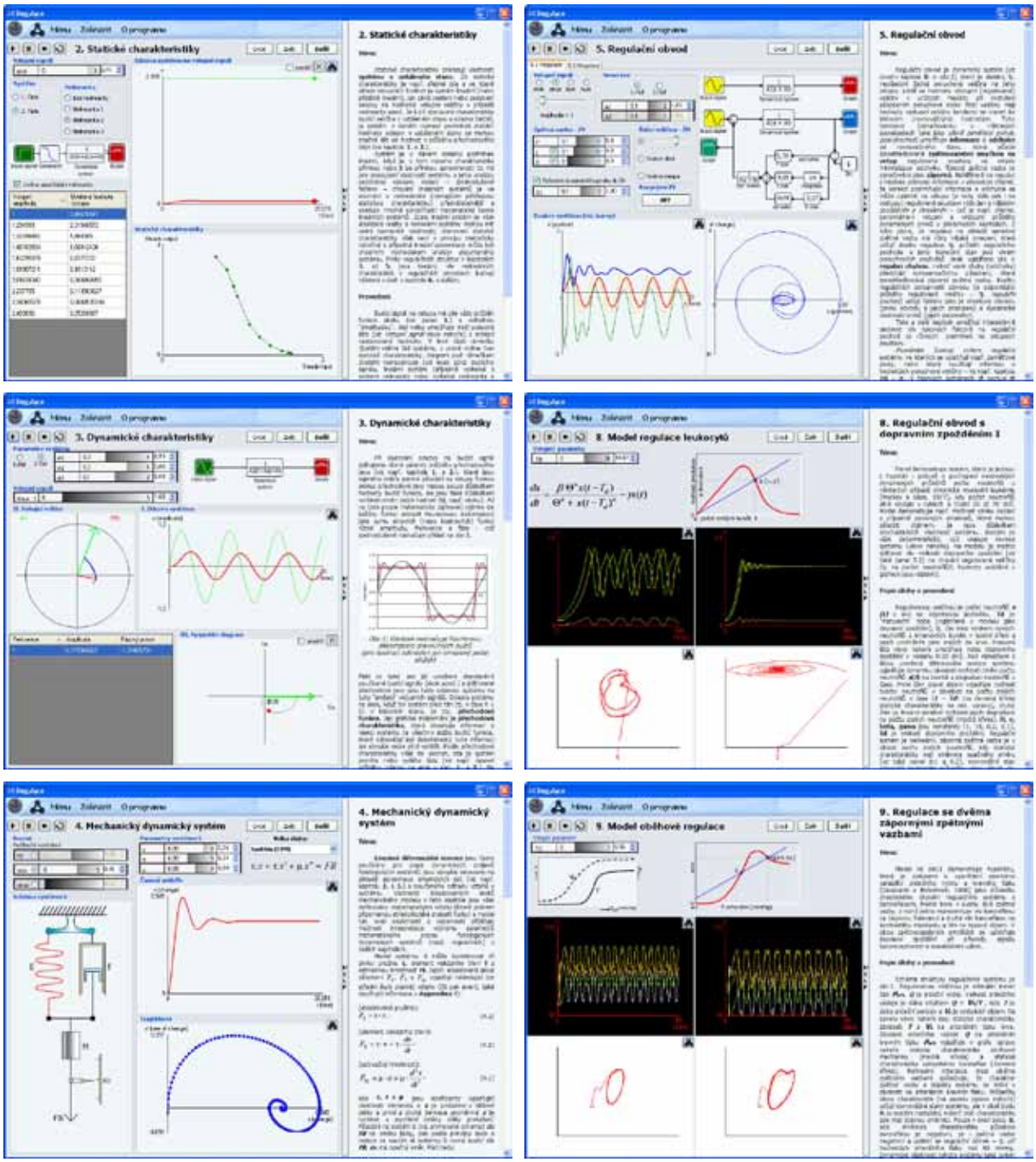


Obr. 104 – Simulátor mechanických vlastností kosterního svalu je flashová aplikace koncipovaná jako výkladová kapitola zahrnující praktická cvičení s modelem v české i anglické verzi.

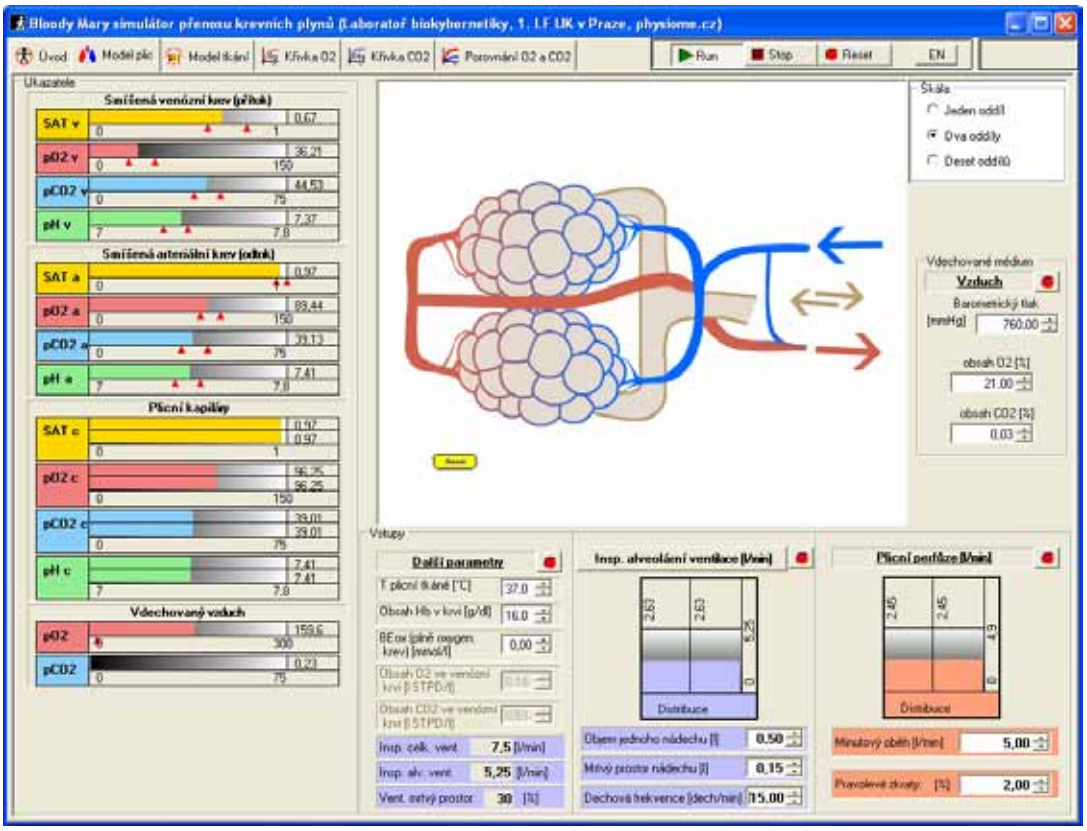
Je přístupný na adrese http://www.physiome.cz/atlas/sval/svalCZ/svalCZ.html.

Dalším příkladem internetem dostupného výukového textu, kde jednotlivé interaktivní modely jsou nedílnou součástí vykládané látky, je kapitola Atlasu fyziologie a patofyziologie pojednávajících o základních dynamických vlastnostech fyziologických regulačních systémů. Je dostupná na adrese http://physiome.cz/atlas/sim/RegulaceSys/.

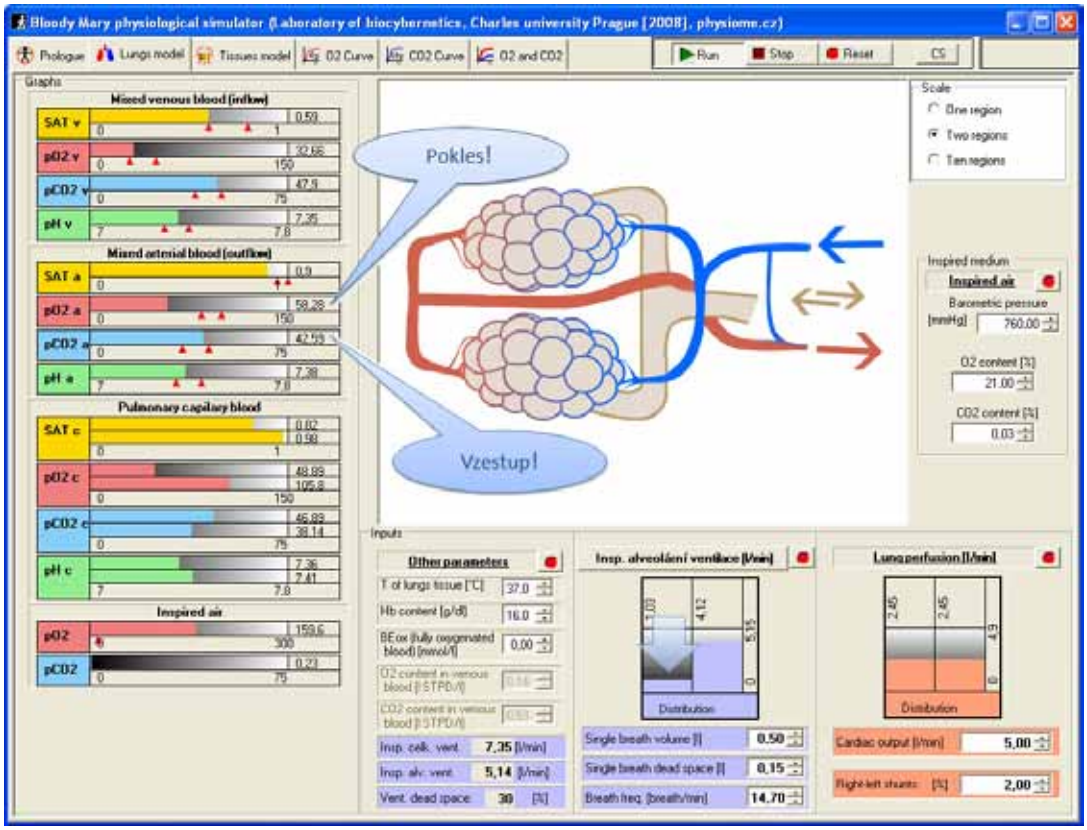
Regulace jsou všudypřítomnou složkou organizace životních procesů, jejich dynamické vlastnosti a chování jsou rozmanité a nezřídka se mění v závislosti na fyziologických nebo patologických faktorech. Učebnice a monografie např. (Wünsch, Dostál & Veselý, 1977) prezentují statická schémata struktur těchto systémů a verbální popis některých jejich funkčních vlastností, ani praktická cvičení neposkytují



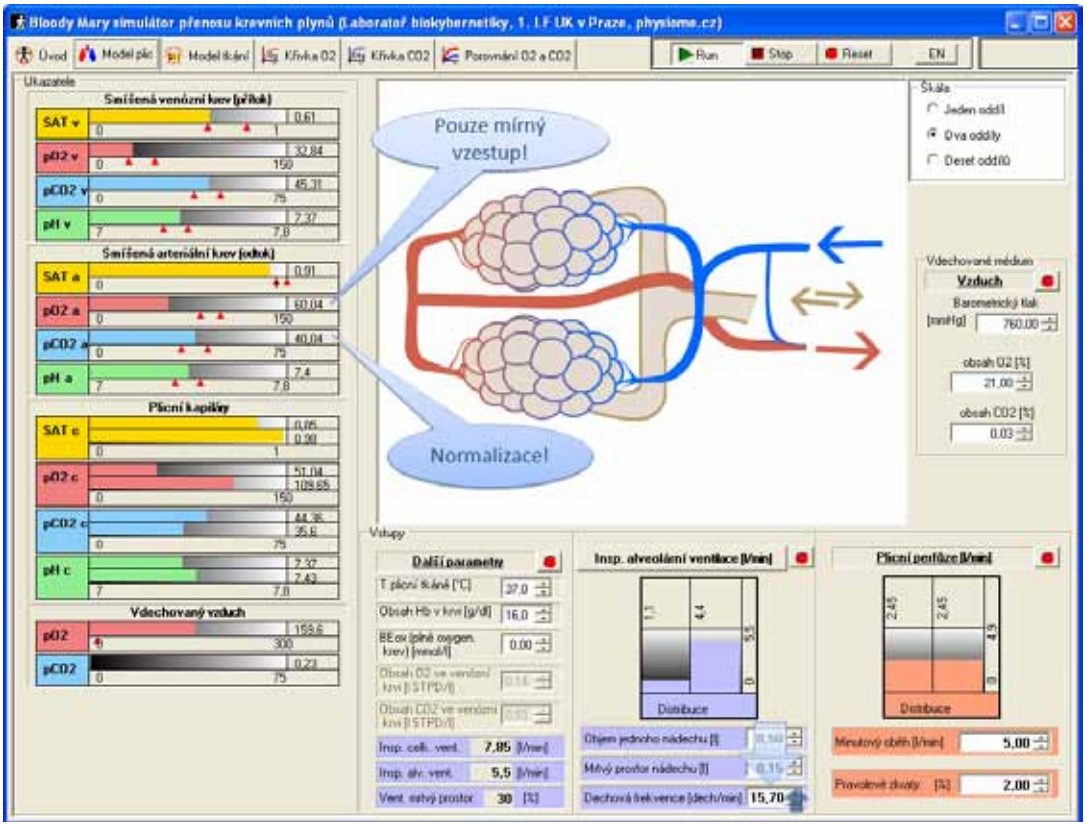
Obr. 105 – Seznamovat posluchače medicíny se základními dynamickými vlastnostmi fyziologických regulačních systémů pouze pomocí tištěného textu, obsahujícího statická schémata struktur těchto systémů a popis některých jejich funkčních vlastností, je obtížné. Tuto problematiku se snaží přiblížit internetem volně dostupný výukový program, kde vykládaná látka je provázána se sadou simulačních experimentů.



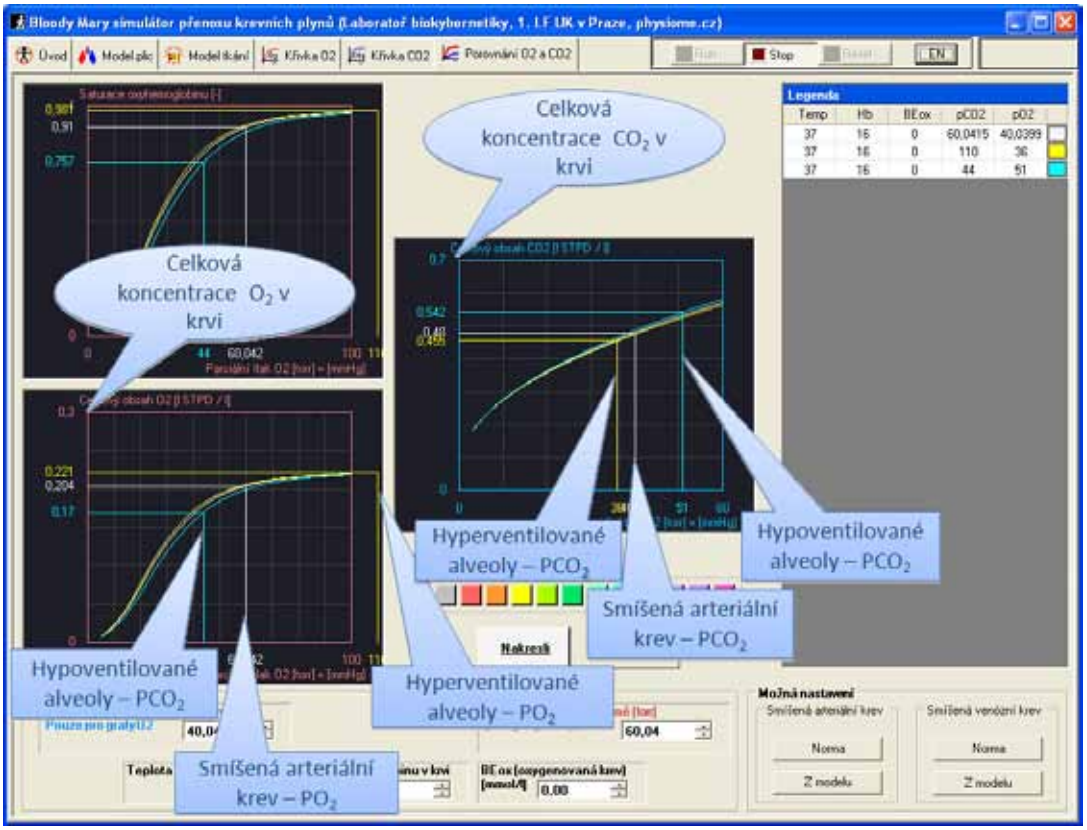
Obr. 106 – Simulační hra s modelem přenosu krevních plynů k vysvětlení následků poruch nerovnoměrnosti ventilace-perfúze. Počáteční stav.



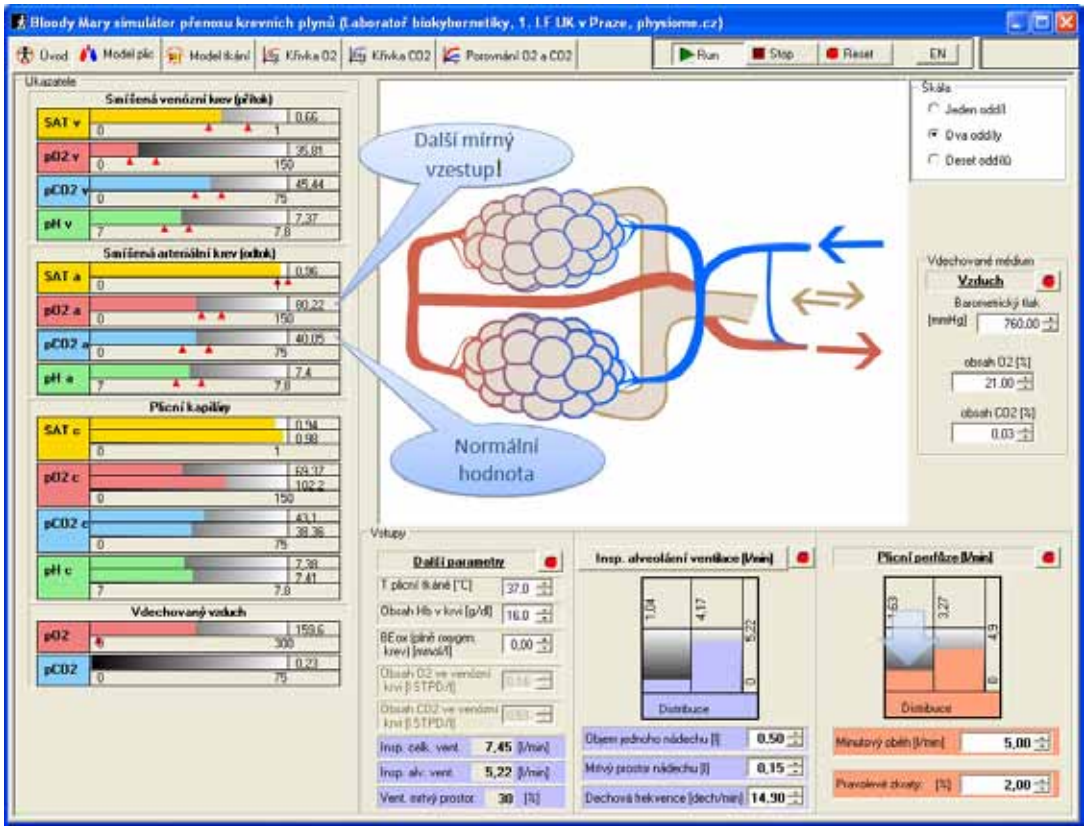
Obr. 107 – Nastavením rozdílné distribuce ventilace se ve smíšené arteriální krvi pO_2 sníží a pCO_2 zvýší.



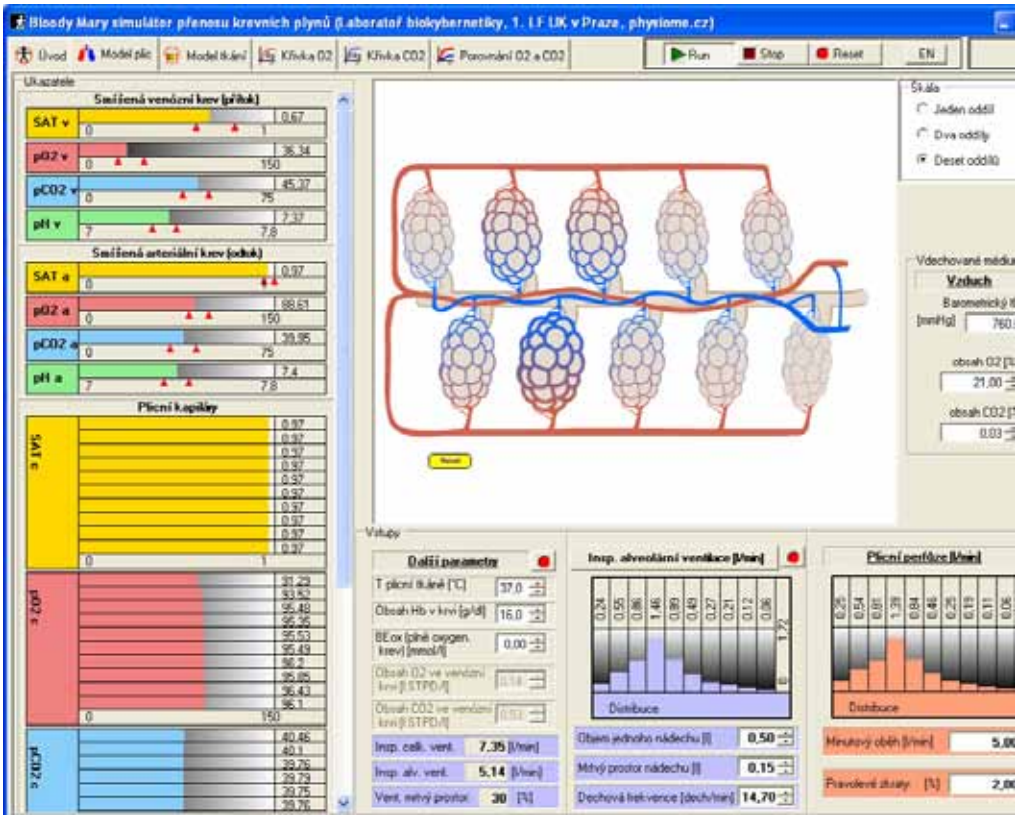
Obr. 108 – Nepatrným zvýšením dechové frekvence dosáhneme normalizace PCO_2 ve smíšené arteriální krvi, PO_2 však zůstává stále nízké. Příčina je v rozdílném tvaru disociačních křivek O_2 a CO_2 - následující obrázek.



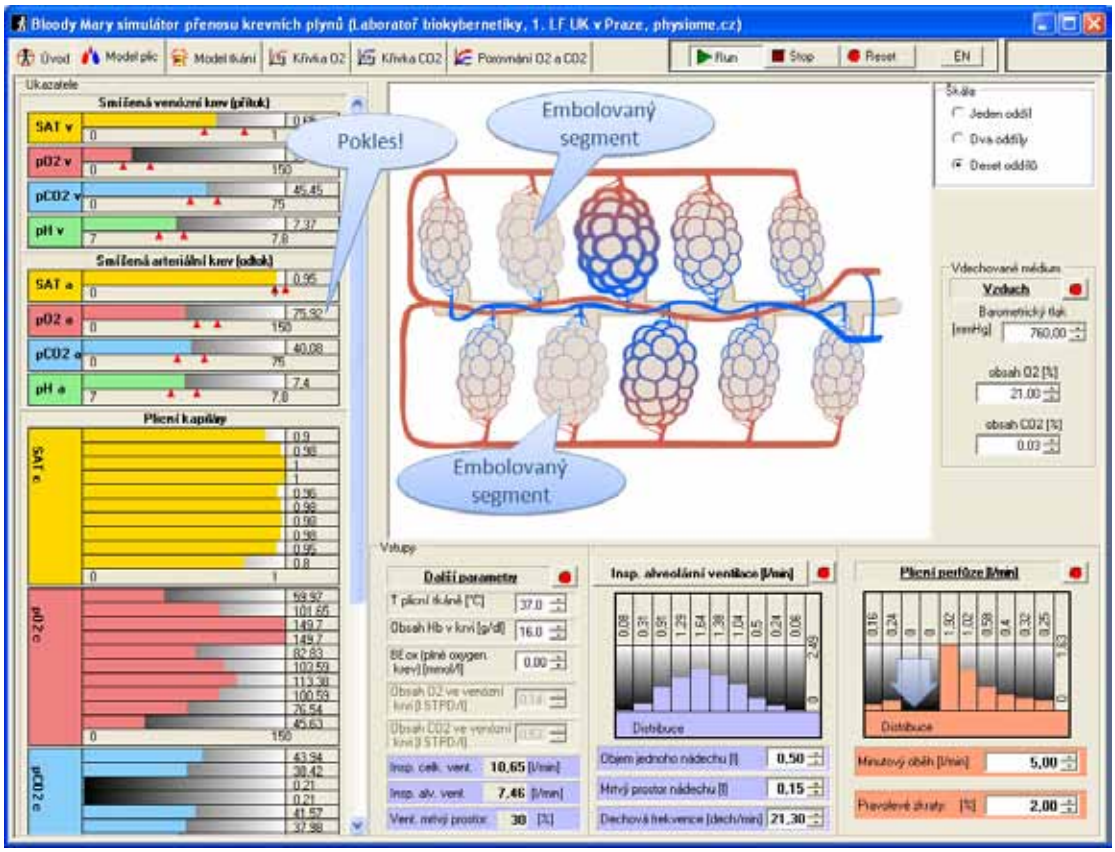
Obr. 109 – Porovnání celkových koncentrací a parciálních tlaků O_2 a CO_2 v hyperventilovaných hypoventilovaných alveolech a ve smíšené arteriální krvi.



Obr. 110 – Omezení perfúze špatně ventilovanými alveoly omezí příměs hypooxygenované krve z hypoventilovaných alveolů, ve smíšené arteriální krvi se proto parciální tlak kyslíku zvýší. Důsledkem ale je také zvýšení odporu plicního krevního řečiště a rozvoj prekapilární plicní hypertenze.



Obr. 111 – Simulace plicní embolie. Využíváme toho, že simulátor umožňuje provádět simulační experimenty i s jemnějším rozložením distribuce ventilace-perfúze do deseti ventilačních a perfúzních segmentů. Výchozí stav.



Obr. 112 – Plicní embolie narušila normální distribuci ventilace-perfúze a způsobila výpadek plicní perfúze v třetím a čtvrtém segmentu (simulujeme tím plicní embolii). Pacient je dušný, respirace kompenzuje pH a parciální tlak oxidu ohličitého na normálních hodnotách, nicméně porušení normální distribuce ventilace-perfúze vede k poklesu arteriálního pO2. Jak můžeme vidět na simulátoru, krev, která nemůže proudit embolovanými plicními segmenty, proudí přes okolní plicní segmenty, kde se poměr ventilace-perfúze snižuje (a alveoly v těchto segmentech jsou vzhledem ke zvýšené perfúzi hypoventilované a hodnoty PO2 a saturace hemoglobinu kyslíkem v kapilární krvi, oddtékající z těchto segmentů jsou nízké).

téměř žádnou možnost, aby se posluchači mohli cestou experimentu s biologickým originálem podrobněji seznámit s jejich dynamickými vlastnostmi a se souvislostmi dynamických vlastností regulačních systémů s parametry těchto systémů.

Tuto mezeru jsme se pokusili částečně vyrovnat pomocí interaktivního výukového textu provázeného sadou simulačních experimentů, které umožňují formou simulačních her seznámení se základními projevy a způsoby stanovení charakteristik prvků regulačních systémů, s vlivem různých parametrů obvodu na průběh regulačního pochodu, stabilitu systému apod. (obr. 105). Součástí výukového textu jsou též příklady několika modelů fyziologických systémů (Wünsch, Matúš & Kofránek, 2007).

To co kdysi Zdeněk Wünsch popisoval staticky ve skriptech a své monografii, nyní mohl realizovat jako simulační hru.

Některé simulátory je možno spouštět samostatně a scénáře k jejich ovládání jsou koncipovány jako součást příslušných výkladových kapitol.

Příkladem je ***komplexní model přenosu krevních plynů,*** který je využíván jako výuková pomůcka při výkladu fyziologie a patofyziologie přenosu kyslíku a oxidu uhličitého. Je volně stažitelný z našeho Atlasu. Příklad využití tohoto simulátoru při výkladu následků poruch ventilačně-perfúzních vztahů zobrazují obrázky 106-110.

Využití simulátoru při výkladu následků plicní embolie na přenos krevních plynů ilustrují obr. 111-112.

4.5.4 Od entuziazmu k technologii a multidisciplinární spolupráci

Již dávno pryč je doba etuziuastů, kteří na přelomu osmdesátých let v nadšení nad novými možnostmi osobních počítačů vytvářeli první výukové programy. Počítače jsou dnes mnohem výkonnější, numerické a grafické možnosti jsou dnes, oproti sklonku osmdesátých let, enormní, pavučina vysokorychlostního internetu obepíná prakticky celý svět a pro uplatnění počítačů ve výuce přináší velký potenciál možného využití.

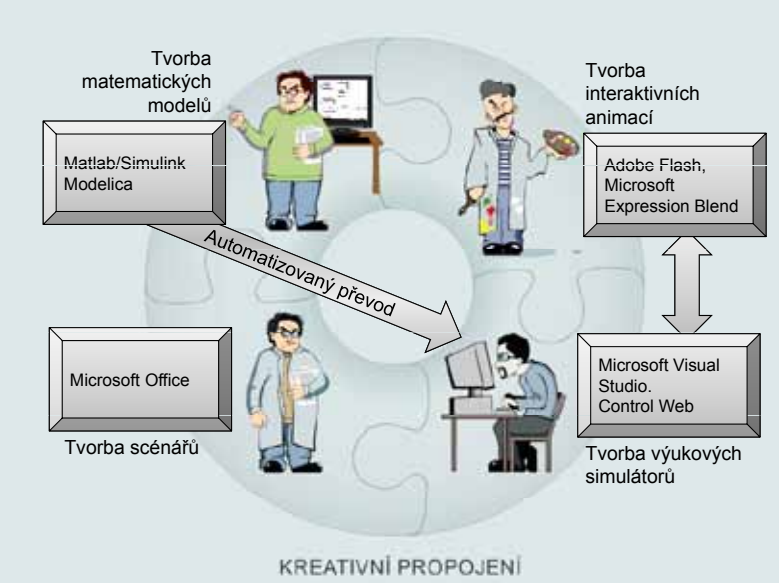
Značně pokročily vývojové nástroje i metodiky softwarové tvorby. Zároveň se ale zvýšily nároky a očekávání uživatelů softwarových aplikací.

Tvorba kvalitního výukového softwaru, který by dokázal využít potenciál, který rozvoj informačních a komunikačních technologií přinesl, dnes nestojí na píli a nadšení jednotlivců. Je náročným a komplikovaným procesem, tvůrčího týmu odborníků různých profesí (obr. 113):

- zkušených učitelů, jejichž scénář je základem kvalitní výukové aplikace
- systémových analytiků, kteří ve spolupráci s profesionály daného oboru jsou odpovědni za vytvoření simulačních modelů pro výukové simulační hry
- výtvarníků, kteří vytvářejí vnější vizuální podobu
- informatiků (programátorů), kteří celou aplikaci „sešijí“ do výsledné podoby

Aby mezioborová spolupráce byla účinná, je zapotřebí pro každou etapu vývoje mít k dispozici řadu vývojových nástrojů a metodologii, které práci jednotlivých členů týmu usnadní a pomohou jim překonat mezioborové bariéry. K ovládnutí těchto nástrojů je zapotřebí věnovat značné úsilí, které se ale nakonec vyplatí.

Domnívám se proto, že nejdůležitějším praktickým výsledkem, kterého se nám v našem Oddělení biokybernetiky a počítačové podpory výuky zatím podařilo dosáhnout, není ani webově dostupný ***Atlas fyziologie a patofyziologie***, nejsou to ani příslušné ***softwarové nástroje, které usnadňují propojení jednotllivých etap vývoje výukového softwaru***, ale ***vybudování mezioborového týmu*** řady profesí – lékařů, matematiků, informatiků, programátorů i výtvarníků, který je ***schopen překonávat mezioborové bariéry*** a je příslibem vývoje dalších prakticky využitelných výukových aplikací.



Obr. 113 – Výukové simulátory jsou vytvářeny multidisciplinárním týmem pedagogů, systémových analytiků, výtvarníků a informatiků. Při jeho tvorbě je nezbytné propojovat jak odborníky různých profesí, tak i vývojové nástroje.

5. Trendy a směry dalšího vývoje

Pokrok v informačních technologiích je velmi rychlý. Objevují se nové technologie, které posouvají možnosti jak v nástrojích pro modelování rozsáhlých systémů, tak i prostředků pro implementaci výukových simulátorů.

Měnit zavedené technologie ve vývojovém týmu je však zapotřebí obezřetně. Jednak ne každá nová technologie časem skutečně naplní výhledy, které se zpočátku jevily jako slibné a perspektivní. Ale především proto, že každá nová technologie znamená určitý čas než je ve vývojovém týmu „absorbována“, než se s ní všichni pracovníci natolik podrobně seznámí, že ji jsou schopni efektivně využívat.

My jsme ve volbě vývojových nástrojů, které využíváme pro tvorbu lékařských simulátorů, v nedávné době učinili dvě zásadní rozhodnutí.

Místo prostředí Matlab/Simulink od firmy Mathworks jsme přešli na novou platformu postavenou na jazyce Modelica. Dlouho jsme s tímto rozhodnutím váhali. Ne proto, že bychom si neuvědomovali důležitost a nové možnosti akauzálního přístupu k modelování, ale proto, že v prostředí firmy Mathworks, které se stalo de fakto průmyslovým stadardem, jsme léta vyvíjeli modely fyziologických systémů a získali při práci s tímto prostředím značné zkušenosti. Udržovali jsme simulinkovou knihovnu Physiolibrary, kterou jsme zpřístupnili odborné veřejnosti na internetu jako otevřený zdroj (www.physiome.cz/simchips) – např. některé bloky z knihovny Physiolibrary, týkající se přenosu krevních plynů a acidobazické rovnováhy, nedávno použil H. Thamin ve své disertační práci na University of Glasgow (Thamin, 2008).

Jak vyplývá z obsáhlého rozboru vlastností akauzálního přístupu k modelování v kapitole 4.2, věnované technologii tvorby výukových simulátorů, má tento přístup zásadní výhody, zejména při tvorbě rozsáhlých hierarchicky členěných modelů. Firma Mathworks v roce 2007 také přišla s akauzálními simulinkovými knihovnami Simscape, v současné době distribuované již ve verzi 3.2 (Mathwoorks, 2009) a na nich navazujícími akauzálními aplikačními knihovnami SimHydraulics, SimMechanics, SimElectornics, SimDriveline a SimPowerSystems) nicméně v porovnání s akauzální koncepcí jazyka Modelica, se nové akauzální simulinkové knihovny jeví, obrazně řečeno, jako nové rouby na starém stromě.

K nejpádnějším důvodům (podrobněji diskutovaném v kapitole 4.3), které nakonec vedly k našemu zásadnímu rozhodnutí o změně platformy pro vývoj simulačních modelů nakonec, patřilo i to, že Modelica není jen firemní standard (jako je třeba Simulink), ale programovací jazyk pro modelování.

Znamená to, že vývojové nástroje může vytvářet více navzájem si konkurujících komerčních i nekomerčních výrobců, což vytváří neustálý tlak na zlepšení vývojových prostředí.

Jak již bylo uvedeno v kapitole 4.3, do jednoho z nekomerčních sdružení, tvořených 12 firmami a 9 univerzitami jsme se zapojili i my. Jedná se o ***Open Source Modelica Consortium*** – http://www.ida.liu.se/labs/pelab/modelica/OpenSourceModelicaConsortium.html které společným úsilím vytváří vývojové prostředí ***Open Modelica*** šiřitelné jako open source. Náš vývojový tým (v rámci firmy Creative Connections s. r. o., která je členem výše zmíněného konsorcia, http://www.creativeconnections.cz/) vytváří v rámci překladače Open Modelica ***generátor kódu v C#***. Model vyvinutý a odladěný v Modelice pak bude možné spouštět v prostředí Silverlight, což nám umožní vytvářet výukové simulátory spustitelné přímo v internetovém prohlížeči.

S tím souvisí i naše druhé rozhodnutí, týkající se ***změny vývojového prostředí pro tvorbu grafického rozhraní výukových simulátorů***. Jednoduché, numericky nenáročné simulátory spustitelné v internetovém prohlížeči bylo možné vytvářet přímo v prostředí Adobe Flash (a zdrojový kód simulačního modelu psát v jazyce ActionScript). Složitější simulátory jsme dosud vyvíjeli v prostředí Microsoft .NET a jejich grafické rozhraní v prostředí Adobe Flash. Nyní je vytváříme v prostředí Microsoft Expression Blend (především z díky možnosti sjednotit vývojové nástroje pro vývoj simulátorů a zjednodušení propojování vytvářených animací se simulačním modelem). Znamenalo to sice věnovat určité úsilí na přeškolení s námi spolupracujících výtvarníků, a vytvoření i některých specializovaných softwarových nástrojů pro ulehčení jejich práce (např. nástroj Animtester – obr. 87-89), ale otevřelo nám to ***možnost provozovat plně funkční simulátory v internetovém prohlížeči*** obsahujícím plugin Siverlight.

Tuto možnost využijeme především pro provozování složitých simulátorů, zejména v ***simulátoru***

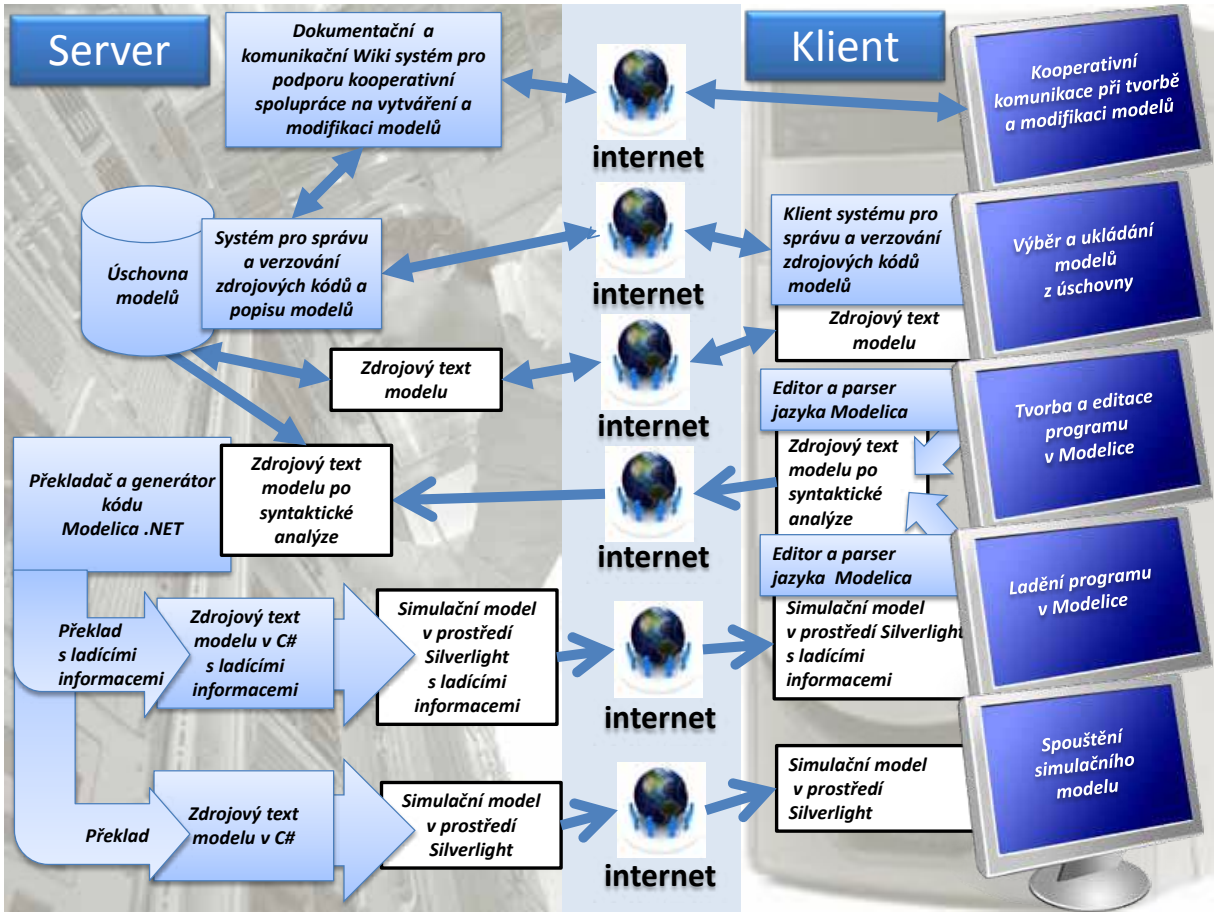
eGolem (vyvíjeném v rámci projektu MŠMT č. 2C067031, veškerá průběžná dokumentace o řešení tohoto projektu je dostupná na adrese http://patf-biokyb.lf1.cuni.cz/wiki/projekty/e-golem).

Simulátory propojené s výkladovými kapitolami vytvořenými již v nové technologii využívající platformu Silverlight využijeme i při dalším rozšiřování a rozvíjení našeho ***výukového Atlasu fyziologie a patofyziologie*** (Kofránek, a další, 2007-2009).

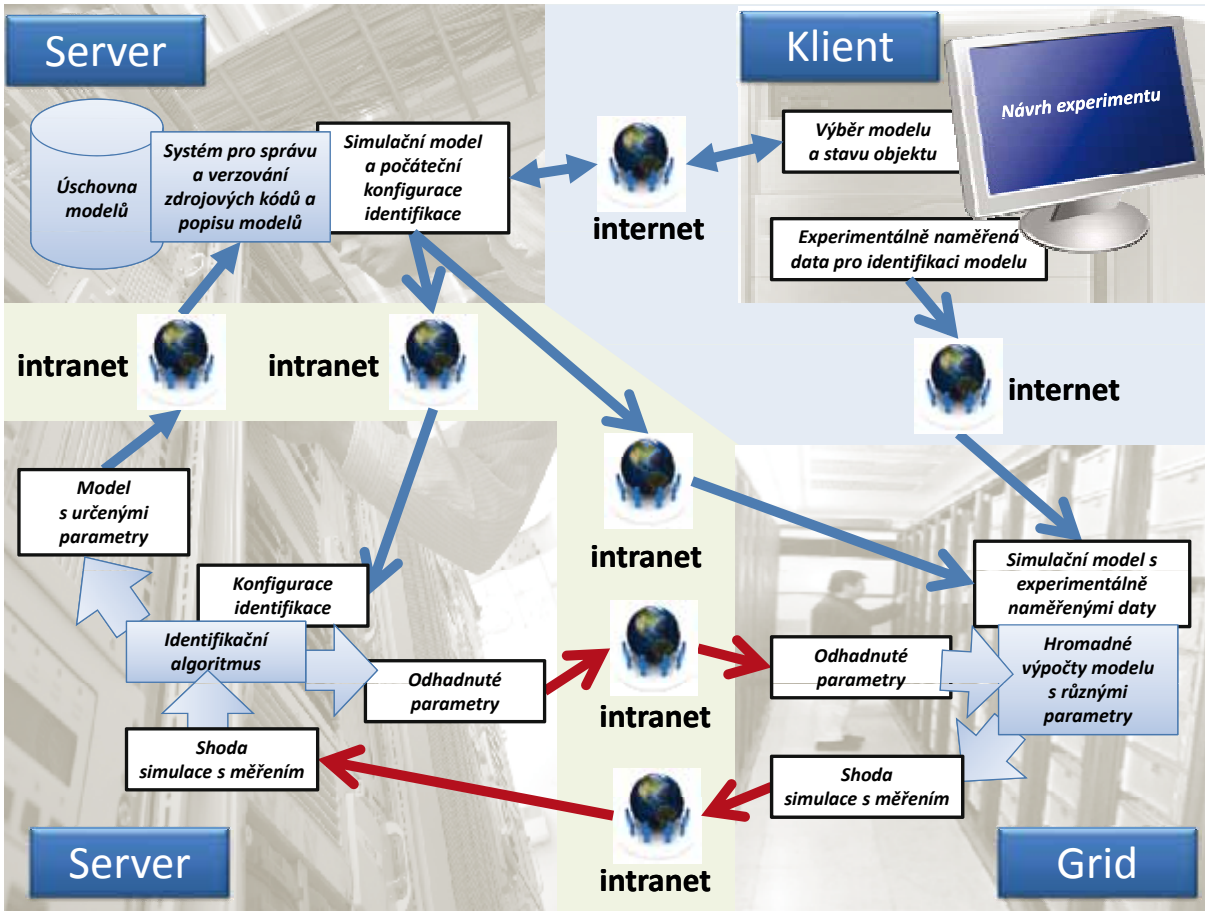
Naším budoucím cílem v rámci projektu Open Modelica je také vytvoření prostředí pro kooperativní tvorbu modelů v jazyce Modelica prostřednictvím internetu, který jsme pracovně nazvali ***Modelica Web Workbench*** (viz obr 114). Zdrojové kódy modelů v jazyce Modelica, včetně jejich podrobného popisu budou ukládány v úložišti zdrojových textů modelů spravovaném příslušným systémem pro správu a verzování zdrojových kódů. Aby bylo možné modely vytvářet, editovat a modifikovat v internetovém prohlížeči, vytváříme editor jazyka Modelica pro prostředí Siverlight (včetně syntaktického analyzátoru). Součástí prostředí bude kompilátor jazyka Modelica (do prostředí .NET) na serveru, generujícího zdrojový kód simulačního modelu pro prostředí Silverlight v jazyce C#.

To umožní vytvářet, editovat a ladit modely v akauzáním prostředí Modeliky přímo v internetovém prohlížeči, překládat jej na serveru a spouštět výsledný model opět v prohlížeči (v prostředí Silverlight).

Systém bude rovněž obsahovat dokumentační a komunikační Wiki systém (Grace, 2009) pro podporu vzdálené komunikace uživatelů při vytváření a editaci modelů. Umožníme tím vzdálenou spolupráci výzkumných týmů na vytváření a modifikaci složitých modelů v jazyce Modelica prostřednictvím internetu. Vzhledem k našemu zaměření je naším cílem podpořit tuto spolupráci zejména v biomedi-



Obr. 114 – Zjednodušená struktura plánovaného vývojového prostředí Modelica Web Workbench. Tvorba, editace modelu v Modelice bude prováděna v internetovém prohlížeči (s využitím platformy Silverlight), překlad bude realizován na serveru a v internetovém prohlížeči bude (opět s využitím platformy Silvelight) spouštěn a testován simulační model. Vývojové prostředí bude umožňovat správu a verzování zdrojových textů modelů na serveru a bude podporovat kooperativní práci několika uživatelů. na tvorbě modelu.



Obr. 115 – Modelica Web Benchmark bude nabízet i možnost identifikace parametrů modelu s využitím možností hromadných výpočtů s využitím výpočetního gridu.

cínské oblasti.

V rámci projektu Physiome zatím existují úložiště modelů založených na kauzálním, blokově orientovaném popisu – např. úložiště fyziologických modelů vytvořených v otevřeném prostředí jazyka JSIM (Raymond, Butterworth & Bassingthwaighte, 2003) je dostupné na adrese http://www.physiome.org/model/doku.php, úložiště modelů popsaných jazykem jazyka CellML (Lloyd, Halstead & Nielsen, 2004) včetně blokově orientovaného překladače těchto modelů je na adrese http://models.cellml.org/.

My bychom chtěli do mezinárodního projektu Physiome přispět tím, že umožníme vytvoření internetem dostupného úložiště biomedicínských modelů ***implementovaných v akauzálním jazyce Modelica***, k němuž nabídneme možnost uložené modely spouštět, editovat a modifikovat prostřednictvím internetového prohlížeče.

Náš vyvíjený systém Modelica Web Workbench bude uživatelům nabízet i možnosti identifikace parametrů modelu. Protože identifikace vyžaduje provádět mnohonásobné výpočty modelu s různými parametry, hodláme pro tuto službu využívat výpočetní grid (viz obr. 115).

Velmi přitažlivým prostředím pro uplatnění výukových simulátorů je prostředí Second Life http://secondlife.com/, jehož první verze byla vytvořena v roce 1999 v Linden Lab (http://lindenlab.com/). Pro programování prostředí (včetně vytváření agentů) se využívá speciální jazyk Linden Scripting Language (LSL) (Heaton, 2007). V srpnu 2008 oznámila Linden Lab možnost využívat pro překlad jejich skriptovacího jazyka LSL platformu Mono (což je „open source“ platforma vývojového prostředí .NET (http://www.mono-project.com/Main_Page) - viz http://wiki.secondlife.com/wiki/Mono. To otevírá možnost využívat pro vytváření agentů v prostředí Second Life kromě LSL, i jazyky založené na prostředí .NET, např. C# (Crooks, Husdon-Smith, & Dearden, 2008). To ale pro nás vytváří lákavou možnost využít vygenerovaný kód C# po překladu modelu z Modelicy jako podklad pro ovládání ***virtuálního pacienta v 3D***

prostředí Second Life.

Prostředím pro provozování lékařských smulátorů ale nemusí být jen obrazovka počítače. Stále větší pedagogický význam budou mít i ***simulátory, které využívají i robotizovanou figurínu pacienta propojenou se simulačním modelem*** (Clay, Que, Petrusa, Sebastian & Govert, 2007; Wayne, Didwania, Feniglass, Fudala, Barsuk & McGaghie, 2008; Rosen, 2008; Kobayashi, Patterson, Overly, Shapiro, Williams & Jay, 2008; Jones & Lorraine, 2008; McGaghie, Siddall, Mazmanian & Myers, 2009).

To je i strategický směr našeho dalšího výzkumu (ve spolupráci dvou vývojových firem, Creative Connections s.r.o. a Inomech s.r.o.), kde v budoucnu chceme uplatnit rozsáhlý simulační model, vytvořený pro simulátor „eGolem“.

6. Závěr

Současná epocha je charakterizována zásadními změnami v technologiích, které ve svém důsledku mění ekonomiku, společnost i způsob života. Pokrok v technologiích vytváří tlak na flexibilitu pracovní síly a zvyšuje požadavek na průběžné rekvalifikace.

Celoživotní vzdělávání se stává nutností ve stále větším počtu oborů. V medicíně je ovšem nutností již dávno. Simulátory a výukové programy využívající simulační hry mohou být v této výuce velmi užitečným prostředkem.

Dlouhá léta jsem se zabýval problematikou formalizace popisu fyziologických regulací prostřednictvím simulačních modelů. Aby výsledky této teoretické práce nebyly jen obsahem vědeckých publikací, ale dostaly se i tam, kde mohou být velmi užitečné, tj. ke studentům medicíny a klinickým lékařům, věnoval jsem se vytváření výukových pomůcek, které by pomocí simulačních her usnadnily pochopení dynamických souvislostí ve zdravém a nemocném organismu.

Založil jsem Oddělení biokybernetiky a počítačové podpory výuky na Ústavu patologické fyziologie 1. LF UK (http://physiome.cz/wiki/) a spolupracující vývojovou firmu Creative Connections s.r.o. (http://www.creativeconnections.cz/) Vytvořil jsem multidiscilinární tým zaměřený na teoretický výzkum fyziologických systémů pomocí matematických modelů, využití modelování pro interpretaci výsledků experimentálního výzkumu, vyhodnocování klinickofyziologických dat a na praktické uplatnění modelů v lékařských simulátorech.

Ve výuce medicíny se využívání simulátorů stále více prosazuje a této oblasti praktického uplatnění simulačních modelů je věnována tato práce. Práce popisuje technologii tvorby výukových simulátorů a výsledky její aplikace.

Pokrok v informačních technologiích je velmi rychlý a použitelné technologie se nám doslova měnily pod rukama. V průběhu posledních patnácti let jsme naši technologii tvorby simulátorů třikrát zásadně změnili. Tyto změny musely být dostatečně dobře promyšlené, protože každá změna technologie znamená zpočátku zdržení, způsobené tím, že členové vývojového týmu musí nejprve tuto technologii „vstřebat“, což určitou dobu trvá.

1. V první verzi technologie v polovině devadesátých let jsme simulátory vytvářeli ve vývojovém prostředí Control Web, původně určeném pro řídící a měřící aplikace v průmyslu. Pomocí rafinovaného triku (kdy jsme do softwarového ovladače měřicí/řídící karty vložili simulační model a signály, místo do připojených zařízení, pak směrovaly jako vstupy do modelu, zatímco místo čtení hodnot z periferií se četly výstupní hodnoty modelu) jsme získali možnost využívat Control Web pro rychlé sestavování simulátorů s bohatým uživatelským rozhraním. Modely jsme navrhovali, odlaďovali a identifikovali v tehdy relativně novém prostředí Matlab/Simulink. Když jsme za v simulinkových modelech propojovali jednotlivé pomocí počítačové myši bloky, tvorbu modelů to velmi zefektivnilo. Jako pamětník jsem při tom vzpomínal na dobu, kdy propojení mezi integrátory a dalšími prvky se na analogové části hybridního počítači EAI 690 v Oborové hybridní laboratoři realizovalo drátovými propojkami. A když jsme nakonec vytvořili softwarový nástroj, který automaticky generoval zdrojový kód ovladače virtuální měřicí/řídící karty pro Control Web, měli jsme možnost snadné a pohodlné aktualizace simulačního jádra vytvářených výukových simulátorů. Touto technologií jsme mimo jiné vytvořili simulátor fyziologických funkcí Golem.
2. Časem se ale ukázalo, že naše vývojové simulátory připomínaly spíše velín automatizované průmyslové linky, než elektronický nástroj pro lékařskou výuku. Zároveň se objevila možnost pomocí programu Macromedia Flash vytvářet ovladatelné animované obrázky, propojitelné pomocí technologie AxtiveX se svým okolím. Navázali jsme proto úzkou spolupráci se Střední uměleckou školou Václava Hollara a věnovali velké úsilí naučit pracovat s tímto nástrojem profesionální výtvarníky. Iniciovali jsme založení Vyšší odborné školy se zaměřením na obor interaktivní grafika, kde nyní také učíme (http://www.hollarka.cz). To nám otevřelo možnosti vkládat do uživatelského rozhraní výukových simulátorů graficky atraktivní obrázky, které jako loutky na nitích ovládal simulační model na pozadí. Naše simulátory pak obsahovaly interaktivní obrázky jak z lékařské učebnice. Místo vývojového

nástroje Control Web jsme začali používat vývojové prostředí Microsoft .Net nebo, pro jednodušší simulátory, vývojové prostředí jazyka ActionScript a FlashPlayer v internetovém prohlížeči. Znamenalo to ale také vytvořit nové softwarové nástroje pro automatizaci převodu modelu z vývojového prostředí Matlab/Simulink do prostředí .NET. Touto technologií jsme začali vytvářet náš internetový Atlas fyziologie a patofyziologie (http://www.physiome.cz/atlas). Simulátory realizované v prostředí Adobe Flash bylo možné spouštět přímo v okně internetového prohlížeče. Složitější simulátory ale vyžadovaly instalaci na počítači klienta, který musel mít příslušná přístupová práva.

3. Nedávno se objevila nová technologie Siverlight, kterou Microsoft reagoval na dnes velmi rozšířený Adobe Flash. Nová technologie od Microsoftu svými možnostmi Flash v mnohém překonává. V Silverlightu je nyní možné vytvářet numericky náročné simulátory s přitažlivým grafickým rozhraním spustitelné přímo v internetovém prohlížeči. Abychom tuto technologii mohli využít, museli jsme naučit naše spolupracující výtvarníky pracovat ve vývojovém prostředí Microsoft Expression Blend (které je pro ně náročnější než graficky více intuitivní prostředí Adobe Flash). Zároveň bylo nutno vytvořit softwarové nástroje, které umožňují lépe oddělit vývojové prostředí určené pro výtvarníka od prostředí pro programátora. Tyto nástroje výtvarníkům usnadňují tvorbu animací snadno propojitelných se simulačním modelem na pozadí. Na trhu se objevila i nová (tzv. akauzální) simulační prostředí, která umožňují jednotlivé části modelu popisovat přímo jako soustavu rovnic a nikoli jako algoritmus řešení těchto rovnic. Ukázalo se, že je daleko efektivnější začít naše modely vytvářet v akauzálním prostředí využívajícím simulační jazyk Modelica, než se spoléhat jen na nové akauzální knihovny v prostředí Simulink. Při vývoji simulačních modelů jsme se proto přeorientovali z vývojového prostředí Matlab/Simulink na vývojové prostředí pro jazyk Modelica. Nyní (v rámci mezinárodního sdružení Open Source Modelica Consortium). vyvíjíme pro překladač Modeliky generátor kódu v C#. To nám umožní spouštět v prostředí Silverlight model vyvinutý a odladěný v Modelice. V nové technologii tvorby výukových simulátorů vytváříme nový internetový výukový simulátor „eGolem“.

Domnívám se, že technologie tvorby výukových simulátorů, popisovaná v této práci, není ohraničena pouze oblastí medicíny a může mít i uplatnění v jiných oblastech.

Konkrétními přínosy této práce je:

- odstranění chyb a následná implementace klasického Guytonova schématu z roku 1972 v prostředí Simulink;
- vytvoření nástroje QHPView pro vizualizaci matematických vztahů v modelu Quantitative Human Physiology
- vytvoření technologie tvorby výukových simulátorů podporujících mezioborovou spolupráci včetně vytvoření příslušných softwarových propojovacích nástrojů;
- vznik multioborového týmu pedagogů, tvůrců simulačních modelů, výtvarníků a programátorů;
- vytvoření simulátoru Golem, využívaného ve výuce na lékařských fakultách u nás i v zahraničí;
- vytvoření internetového Atlasu fyziologie a patofyziologie, využívaného na lékařských fakultách v ČR a SR, nyní se připravuje začlenění anglicky lokalizovaných součástí atlasu do „teaching resources“ Americké fyziologické společnosti;
- vytvoření knihovny fyziologických modelů Physiolib pro prostředí Simulink;
- vytvoření komplexního modelu fyziologických regulací v akauzálním prostředí jazyka Modelica.

Za nejdůležitější výsledek ovšem považuji tým mladých následovníků, který se při řešení problematiky popisované v této práci vytvořil.

7 Literatura

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

[2.] Amosov, N. M., Palec, B. L., Agapov, G. T., Ermakova, I. I., Ljabach, E. G., Packina, S. A., a další. (1977). *Těoretičeskoe issledovanie fyziologičeskich sistěm.* Kiev: Naukova Dumka.

[3.] Anderson, J., Goplen, C., Murray, L., seashore, K., Soundarrajan, M., Lokuta, A., a další. (2009). Human respiratory mechanics demonstration model. *Advan. Physiol. Edu. , 33*, stránky 53-59.

[4.] Bassingthwaighte, J. B. (2000). Strategies for the Physiome Project. *Annals of Biomedical Engeneering , 28*, stránky 1043-1058.

[5.] Binstadt, E. S., Walls, R., White, B. A., Nadel, E. S., Takavesu, J. K., & Barker, T. D. (2006). A Comprehensive Medical Simulation Education Curriculum for Emergency Medicine Residents. *Annals of Emergency Medicine , 49*, stránky 495-504.

[6.] Boulos, M. N., Hetherington, L., & Wheeler, S. (2007). Second Life: an overview of the potential of 3-D virtual worlds in medical and health education. *Health Information & Libraries Journal* (24), stránky 233-245.

[7.] Branstrom, M. J., Haynes, L. W., Hoehn, K., LePage, P., Marieb, E. N., Mitchell, S. J., a další. (2008). *Interactive Physiology, 10 System Suite*. (S. Beauparlant, Editor, & Benjamin Cummings, San Francisco, CA) Načteno z http://www.interactivephysiology.com.

[8.] Brudgárd, J., Hedberg, D., Cascante, M., Gedersund, G., Gómez-Garrido, A., Maier, D., a další. (2009). Creating a Bridge between Modelica and the Systems Biology Community. *Proceedings 7th Modelica Conference, Como, Italy, Sep. 20-22, 2009.* Como: The Modelica Association.

[9.] Burkhoff, D., & Dickstein, M. L. (2003). *The heart simulator*. Načteno z http://www.columbia.edu/itc/hs/medical/heartsim.

[10.] Carnevale, N. T., & Hines, M. L. (2006). *The Neuron Book.* Cambridge: Cambridge University Press.

[11.] Cason, C. L., Kardong-Edgren, S., Cazzell, M., Behan, D., & Mancini, M. E. (2009). Innovations in Basic Life Support Education for Healthcare Providers: Improving Competence in Cardiopulmonary Resuscitation Through Self-Directed Learning. *Journal for Nurses in Staff Development , 25*, stránky E1-E13.

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

[13.] Clark, R. C., & Mayer, R. E. (2008). *E-learning and the science of instruction: proven guidelines for consumers.* San Francisco: Pfeiffer.

[14.] Clay, A. S., Que, L., Petrusa, E. R., Sebastian, M., & Govert, J. (2007). Debriefing in the intensive care unit: A feedback tool to facilitate bedside teaching. *Critical Care Medicine: , 35*, stránky 738-754.

[15.] Coleman, T. G., & Randall, J. E. (1983). HUMAN. A comprehensive physiological model. *The Physiologist, , 26*, stránky 15-21.

[16.] Coleman, T. G., & Summers, R. L. (1997). Using mathematical models to better understand integrative physiology. *Journal of Physiology and Biochemistry , 53*, stránky 45–46.

[17.] Coleman, T. G., Hester, R. L., & Summers, R. L. (2009). *Quantitative Human Physiology*. Načteno z http://physiology.umc.edu/themodelingworkshop.

[18.] Collins, D. (1995). *Designing object-oriented user interfaces.* Redwood City, CA: Benjamin Cummings (ISBN: 0-8053-5350-X).

[19.] Comenius, J. A. (1656). *Schola Ludus, seu Encyclopaedea Viva.* Sarospatak.

[20.] Crooks, A., Husdon-Smith, A., & Dearden, J. (2008). Agent street: an environment for exploring agent-based models in Second Life. *Journal of Artificial Societies and Social Simulation , 12*, stránky 1-10 (http://jasss.soc.surrey.ac.uk/12/4/10.html).

[21.] Dabney, J. B., & Harman, T. L. (2004). *Mastering Simulink.* Houston: Prentice Hall.

[22.] Danforth, D., Procter, M., Heller, R., Chen, R., & J. M. (2009). Development of Virtual Patient Simulations for Medical Education. *Journal of Virtual World Research , 2*, stránky https://journals.tdl.org/jvwr/article/view/707/503, 1-11.

[23.] Davis, M. J. (2001). Basic principles of synaptic physiology illustrated by computer model. *Advan. Physiol. Edu., 25*, stránky 1-12.

[24.] Davis, M. J., & Gore, R. W. (2001). Determinants of cardiac function: simulation of a dymamic cardiac pump for physiology instruction. *25*, stránky 13-35.

[25.] Day, R. S. (2006). Challenges of biological realism and validation in simulation-based medical education. *Artificial Inteligence in Medicine , 38*, stránky 47-66.

[26.] de Freitas, S. I. (2006). Using games and simulations for supporting learning. *Learning, Media and Technology , 31*, stránky 343 – 358.

[27.] Diener, S., Windsor, J., & Bodily, D. (2009). Design and Development of Medical Simulations in Second Life and OpenSim. *EDUCAUSE Australasia 2009, Perth Western Australia. 3-6 May 2009* (stránky http://hdl.handle.net/2292/4305, 1-13). Perth: Educause Australasia.

[28.] Dunkin, B., Adrales, G. L., Apelgren, K., & Mellinger, J. D. (2007). Surgical simulation : a current review. *Surgical endoscopy , 21*, stránky 357-366.

[29.] Ellaway, R. H., Kneebone, R., Lachapelle, K., & Topps, D. (2009). Practica continua: Connecting and combining simulation modalities for integrated teaching, learning and assessment. *Medical Teacher , 31*, stránky 725-731.

[30.] Felipini, C. L., de Adrande, J. P., Lucchi, J. C., da Fonseca, J. W., & Nicolosi, D. (2008). An Electro-Fluid-Dynamic Simulator for the Cardiovascular System. *Artificial Organs , 32*, stránky 348-354.

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

[32.] Fritzon, P. (2003). *Principles of object-oriented modeling and simulation with Modelica 2.1.* Wiley-IEE Press.

[33.] Fukui, Y., & Smith, N. T. (1981a). Interaction among Ventilation, the Circulation, and the Uptake and Distribution of Halothane. Use of a Hybrid Computer Model I. The Basic Model. *Anesthesiology , 54*, stránky 107-118.

[34.] Fukui, Y., & Smith, N. T. (1981b). Interaction Among Ventilation, the Circulation, and the Uptake and Distribution of Halothane. Use of a Hybrid Computer Model II. Spontaneous vs. Controlled Ventilation, and the Effects of CO2. *Anesthesiology , 54*, stránky 119-124.

[35.] Gondžilašvili, J., Kofránek, J., Pokorný, Z., & Brelidze, Z. (1987). Matěmatičeskoje moděliro-vanie intěnsivnosti obmennych processov pri različnych uslovijach vněšněj sredy. *Věstnik Akadě-mii Medicinskich Nauk SSSR* , stránky 81-87.

[36.] Grace, T. P. (2009). Wikis as a knowledge management tool. *Journal of knowladge management , 13*, stránky 64-74.

[37.] Grodins, F. S., Buell, J., & Bart, A. J. (1967). Mathematical analysis and digital simulation of the respiratory control system. *J.Appl. Physiol. , 22*, stránky 260-276.

[38.] Guyton, A. C., Coleman, T. G., & Grander, H. J. (1972). Circulation: Overall Regulation. *Ann. Rev. Physiol. , 41*, stránky 13-41.

[39.] Guyton, A. C., Jones, C. E., & Coleman, T. G. (1973). *Circulatory Physiology: Cardiac Output and Its Regulation.* Philadelphia, London, Toronto: WB Saunders Company.

[40.] Guyton, A. C., Taylor, A. E., & Grander, H. J. (1975). *Circulatory physiology II. Dynamics and control of the body fluids.* Phildelphia, London, Toronto: W. B. Saunders.

[41.] Haas, O. C., & Burnham, K. J. (2008). Systems Modeling and Control Applied to Medicine. V O. C. Haas, & K. J. Burnham, *Intelligent and Adaptive Systems in Medicine* (stránky 17-52). Boca Raton Fl, USA: CRC Press.

[42.] Hall, J. E. (2004). The pioneering use of system analysis to study cardiac output regulation. *Am.J.Physiol.Regul.Integr.Comp.Physiol. , 287*, stránky R1009-R1011.

[43.] Hammond, J., Berman, M., Chen, B., & Kushins, L. (2002). Incorporation of a Computerized Human Patient Simulator in Critical Care Training: A Preliminary Report. *The Journal of Trauma, Injury, Infection, and Critical Care , 53*, stránky 1064-1067.

[44.] Harel, D. (1987). Statecharts: a visual formalism for complex systems. *Science of computer Programming , 8*, stránky 231-274.

[45.] Harel, D., Kugler, H., & Pnueli, A. (2005). Synthesis revisited: Generating statechart models from scenario-based requirements. In *Formal Methods in Software and Systems Modeling* (Vol. 3393, pp. 309-324). Berlin / Heidelberg: Springer.

[46.] Heaton, J. (2007). *Introduction to Linden Scripting Language for Second Life.* Heaton Research, Inc.(ISBN: 9781604390056).

[47.] Hines, M. L., & Carnevale, N. T. (2001). NEURON: a tool for neuroscientists. *The Neuroscientist , 7*, stránky 123-135.

[48.] Hodgkin, A., & Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology , 117*, stránky 500-544.

[49.] Hunter, P. J., Robins, P., & Noble, D. (2002). The IUPS Physiome Project. *Pflugers Archive-European Journal of Physiology* (445), stránky 1-9.

[50.] Ikeda, N., Marumo, F., & Shirsataka, M. (1979). A Model of Overall Regulation of Body Fluids. *Ann. Biomed. Eng. , 7*, stránky 135-166.

[51.] Jackson, M. E., & Gnadt, J. W. (1999). Numerical simulation of nonlinear feedback model of saccade generation circuit implemented in the LabView graphical programming language. *Journal of Neuroscience Methods , 82*, stránky 137-145.

[52.] Jones, A., & Lorraine, S. (2008). Can human patient simulator be used in physiotherapy education? *The Internet Journal of Allied Health Sciences and Practice , 5*, stránky 1-5.

[53.] Kho, M. C. (2000). *Physiological control systems.* New York: IEE Press.

[54.] Kobayashi, L. K., Patterson, M. D., Overly, F. L., Shapiro, M. J., Williams, K. A., & Jay, G. D. (2008). Educational and research implications of portable human patient simulation in acute care medicine. *Academic Emergency Medicine , 15*, stránky 1166-1174.

[55.] Kofránek, J. (Producent), Klučina, P. (Autor), Kofránek, J. (Režisér), Rejl, V. (Výtvarník), & Obdržálek, J. (Hudba). (2006). *Möten i historien, Historica encounter, Dějinné setkání* [Animovaný film]. Bajt servis s.r.o. ve spolupráci s Univerzitou Karlovou pro 1. Sovineckou a.s.

[56.] Kofránek, J. (2009). Komplexní model acidobazické rovnováhy. (Anglická verze: Complex model of acid-base balance je dostupná na adrese http://www.physiome.cz/references/medsoft2009acidbase.pdf, model je na adrese http://www.physiome.cz/acidbase). V M. Zeithamlová (Editor), *MEDSOFT 2009* (stránky 23-60). Praha: Agentura Action M.

[57.] Kofránek, J. (1980). *Modelování acidobazické rovnováhy krve. Disertační práce.* Praha: Univerzita Karlova v Praze, Fakulta všeobecného lékařství.

[58.] Kofránek, J. (2009). What is behind the curtain of a multimedia aducational games? *EATIS 09 Contribution Proceedings. Euro American Conference on Telematics & Information Systems, 2009.* (stránky 229-236). Praha: Wirelesscom sro., ISBN 978-80-87205-07-5. Práce je dostupná na adrese http://www.physiome.cz/references/eatis2009.pdf.

[59.] Kofránek, J., & Rusz, J. (2007). Od obrázkových schémat k modelům pro výuku. *Československá fyziologie, 56*, stránky 69-78. Práce je dostupná na adrese http://www.physiome.cz/references/CSFyziol2007.pdf.

[60.] Kofránek, J., & Svačina, Š. (2001). Multimedia simulators of glycoregulatory mechanisms as an interactive teaching tool. V J. G. Anderson, & M. Kapzer (Editor), *Simulation in the Health and Medical Sciences 2001* (stránky 165-170). San Diego: Society for Computer Simulation International, Simulation Councils.

[61.] Kofránek, J., Andrlík, M., & Kripner, T. (2005). Biomedical Educations with Golem. V V. Mařík, P. Jacovkis, O. Štěpánková, & J. Kléma (Editor), *Interdisciplinary Aspects of Human-Machine Co-existence and Co-operation.* (stránky 142-151). Praha: Czech Technical University in Prague. Práce je dostupná na adrese http://www.physiome.cz/references/eGolem2005.pdf.

[62.] Kofránek, J., Andrlík, M., & Kripner, T. (2003). Virtual patient behind the screen using computer simulator GOLEM. V D. D. Feng, & E. R. Carson (Editor), *Proceedings volume from the 5th IFAC Symposium, Melbourne 2003.*, (stránky 479-485). Práce je dostupná na adrese http://www.physiome.cz/references/IFAC2003.pdf.

[63.] Kofránek, J., Andrlík, M., Hlaváček, J., Mateják, P., Matoušek, S., Obdržálek, J., a další. (2007-2009). (J. Kofránek, Editor) Získáno 2009, z Atlas fyziologie a patofyziologie (ISSN 1803-6619): http://www.physiome.cz/atlas.

[64.] Kofránek, J., Andrlík, M., Kripner, T., & Mašek, J. (2002a). From art to industry in design of biomedical simulators. Experience of the Golem simulator project. V M. Callaos, G. Whymark, & W. Lesso (Editor), *The 6th World Multiconference on Systemics, Cybernetics and Informatics. Orlando, Florida, USA. Proceedings. Volume XIII. Concepts and Applications of Systemics, Cybernetics and Informatics III.*, (stránky 249-256).

[65.] Kofránek, J., Andrlík, M., Kripner, T., & Mašek, J. (2002d). From art to industry: experiences with the design and development of biomedical simulators. V A. Boukerche, M. Sechi Moretti, & B. G. Riso (Editor), *Proceedings I2TS'2002 International Information Technology Symposium, October 1-5, 2002, Florianopolis, SC, Brazil.* (stránky Technical Sections 3: 1-8). Fundaco Barddal de Educatio e Cultura.

[66.] Kofránek, J., Andrlík, M., Kripner, T., & Mašek, J. (2002c). From simulation chips to biomedical simulator. V K. Amborski, & H. Meuth (Editor), *Modelling and Simulation 2002. Proceedings of !6th European Simulation Multiconference* (stránky 431-436). Darmstadt: SCS Publishing House. Práce je dostupná na adrese http://www.physiome.cz/references/EUROSIM2002.pdf.

[67.] Kofránek, J., Andrlík, M., Kripner, T., & Mašek, J. (2002b). Simulation chips for GOLEM – multimedia simulator of physiological functions. V J. G. Anderson, & M. Kapzer (Editor), *Simulation in the Health and Medical Sciences 2002.* (stránky 159-163). San Diego: Society for Computer Simulation International, Simulation Councils.

[68.] Kofránek, J., Andrlík, M., Kripner, T., & Stodulka, P. (2005). From Art to Industry: Development of Biomedical Simulators. *The IPSI BgD Transactions on Advanced Research , 1 #2(Special Issue on the Research with Elements of Multidisciplinary, Interdisciplinary, and Transdisciplinary: The Best Paper Selection for 2005)*, stránky 62-67. Práce je dostupná na adrese http://www.physiome.cz/references/IPSI2005.pdf.

[69.] Kofránek, J., Anh Vu, L. D., Snášelová, H., Kerekeš, R., & Velan, T. (2001). GOLEM – Multimedia

simulator for medical education. V L. Patel, R. Rogers, & R. Haux (Editor), *MEDINFO 2001, Proceedings of the 10th World Congress on Medical Informatics. 1042-1046.* London: IOS Press. Práce je dostupná na adrese http://www.physiome.cz/references/MEDINFO2001.pdf.

[70.] Kofránek, J., Brelidze, Z., & Gondžilašvili, J. (1984). Modělirovanie dinamiki gazoobmena i jego analiz posredstvom imitacionnych eksperimentov na EVM. V *Voprosy biologičeskoj i medicinskoj techniki.* (stránky 90-107). Tbilisi: Mecniereba.

[71.] Kofránek, J., Kripner, T., Andrlík, M., & Mašek, J. (2003). Creative connection between multimedia, simulation and software development tools in the design and development of biomedical educational simulators. *Simulation Interoperability Workshop, Position papers, Volume II, paper 03F-SIW-102.*, (stránky 677-687).

[72.] Kofránek, J., Maruna, P., Andrlík, M., Stodulka, P., Kripner, T., Wünsch, Z., a další. (2004). The design and development of interactive multimedia in educational software with simulation games. *Proceedings of the Seventh IASTED International Conference on Computer Graphics And Imaging.* (stránky 164-170). Anaheim, Calgary, Zurich: The International Assotiation of Science and Technology for Development. Práce je dostupná na adrese http://www.physiome.cz/references/IASTED2004.pdf.

[73.] Kofránek, J., Mateják, M. Privitzer, P., Tibula, M. & (2008). Causal or acausal modeling: labour for humans or labour for machines. V C. Moler, A. Procházka, R. Bartko, M. Folin, J. Houška, & P. Byron (Editor), *Technical Computing Prague 2008, 16th Annual Conference Proceedings, CD ROM* (stránky 058_kofranek.pdf: 1-16). Praha: Humusoft s.r.o. Práce je dostupná na adrese http://www.physiome.cz/references/TCP2008.pdf.

[74.] Kofránek, J., Mateják, M., & Privitzer, P. (2009a). School as a (multimedia simulation) game. The use of object tools for designing multimedia applications for biomedical teaching. V V. C. Moler, A. Procházka, R. Bartko, M. Folin, J. Houška, & P. Byron (Editor), *Technical Compuning Prague 2009, 17th Annual Conference Proceedings. CD ROM, ISBN 978-80-7080-733-0, internetový sborník: http://dsp.vscht.cz/konference_matlab/MATLAB09/* (stránky 55: 1-27). Praha: Humusoft s.r.o. Práce je dostupná na adrese http://www.physiome.cz/references/TCP09.pdf.

[75.] Kofránek, J., Mateják, M., & Privitzer, P. (2009b). Leaving toil to machines - building simulation kernel of educational software in modern software environments. CD ROM. V L. Dušek, D. Schwarz, & S. Štípek (Editor), *Mefanet 2009, Conference Proceedings* (stránky kofranek.pdf: 1-39). Brno: Masarykova Univerzita. Práce je dostupná na adrese http://www.physiome.cz/references/MEFANET2009.pdf.

[76.] Kofránek, J., Mateják, M., Matoušek, S., Privitzer, P., Tribula, M., & Vacek, O. (2008). 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* (stránky kofranek.pdf: 1-26). Brno: Masarykova Univerzita, Brno. Práce je dostupná na adrese http://www.physiome.cz/references/MEFANET2008.pdf.

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

[78.] Kofránek, J., Matoušek, S., Andrlík, M., Stodulka, P., Wünsch, Z., Privitzer, P., a další. (2007). Atlas of physiology - internet simulation playground. V B. Zupanic, R. Karba, & s. Blažič (Editor), *Proceedings of the 6th EUROSIM Congress on Modeling and Simulation, Vol. 2. Full Papers (CD ROM)* (stránky MO-2-P7-5: 1-9). Ljubljana: University of Ljubljana. Práce je dostupná na adrese http://ww.physiome.cz/references/EUROSIM2007.pdf.

[79.] Kofránek, J., Munclinger, M., Fusek, M., Gondžilašvili, J., Brelidze, Z., Šerf, B., a další. (1988). Evaluation of cardiorespiratory functions during heart catheterisation throught simulation

model evaluation. V E. R. Carson, P. Kneppo, & P. Krekule, *Advances in biomedical measurement* (stránky 311-320). New York: Plenum Press.

[80.] Kofránek, J., Pokorný, Z., Wünsch, Z., Brelidze, Z., Gondžilašvili, J., & Verigo, V. (1982a). Kislotno-ščoločnaja reguljacija vnutrennej sredy organizma. V M. Kotva (Editor), *Proceedings of the symposium Simulation of Systems in Biology and Medicine, Prague 1982.* (stránky B13-C4). Praha: Dům Techniky, ČSVTS.

[81.] Kofránek, J., Pokorný, Z., Wünsch, Z., Brelidze, Z., Gondžilašvili, J., & Verigo, V. (1982b). Matěmatičeskaja moděl gazoobměna v alveoljach i tkaňach kak časť obščej moděli vnutrenněj sredy organizma. V M. Kotva (Editor), *Proceedings of the symposium Simulation of Systems in Biology and Medicine, Prague 1982.* (stránky C11-D2). Praha: Dům Techniky, ČSVTS.

[82.] Kofránek, J., Pokorný, Z., Wünsch, Z., Brelidze, Z., Gondžilašvili, J., & Verigo, V. (1982c). Moděl vodnosolevogo osmotičeskogo i kislotnoščoločnogo gomeostazisa vnutrenněj sredy. V M. Kotva (Editor), *Proceedings of the symposium Simulation of Systems in Biology and Medicine, Prague 1982.* (stránky C11-D2). Praha: Dům techniky, ČSVTS.

[83.] Kofránek, J., Privitzer, P., Matoušek, S., Vacek, O., & Tribula, M. (2009). Schola Ludus in modern garment: use of web multimedia simulation in biomedical teaching. *Proceedings of the 7th IFAC Symposium on Modelling and Control in Biomedical Systems, Aalborg, Denmark, August 12-14, 2009*, (stránky 425-430). Práce je dostupná na adrese http://www.physiome.cz/references/IFAC2009.pdf.

[84.] Kofránek, J., Rusz, J., & Matoušek, S. (2007). Guytons Diagram Brought to Life - from Graphic Chart to Simulation Model for Teaching Physiology. V P. Byron (Editor), *Technical Computing Prague 2007. Full paper CD-ROM proceedings.* (stránky 1-13). Praha: Humusoft s.r.o. & Institute of Chemical Technology. Práce je dostupná na adrese http://www.physiome.cz/references/TCP07.pdf.

[85.] Kofránek, J., Snášelová, H., Anh Vu, L. D., & Svačina, Š. (2001). Multimedia simulation games in medical education. V E. J. Kerckhoffs, & M. Šnorek (Editor), *Proceedings of European Simulation Multiconference, Prague 2001.* (stránky 995-999). Erlagen: SCS Publishing House.

[86.] Kofránek, J., Snášelová, H., Anh Vu, L. D., Janicadis, P., & Velan, T. (2001). Virtual patients behind the screen. V E. J. Kerckhoffs, & M. Šnorek (Editor), *Proceedings of European Simulation Multiconference, Prague 2001.* (stránky 1000-1008). Erlagen: SCS Publishing House.

[87.] Kofránek, J., Velan, T., & Janicadis, P. (2000). Golem – Computer simulator of body fluids and acid-base disorders as an effitient teleeducation tool. *Proceedings of IFAC Symposium, Karlsburg, Greifswald, 30.3.- 1.4.2000.* (stránky 233-242). Oxford: Pergamon, Elsevier Science.

[88.] Kofránek, J., Velan, T., & Kerekeš, R. (1997). Golem: a Computer Simulator of Physiological Functions as an Efficient Teaching Tool. V Y. M. Theo, W. C. Wong, & T. J. Okeu (Editor), *Legacy for 21 Century. Proceedings of the World Congress on System Simulation.* (stránky 407-411). Singapore: IEE Singapore Section.

[89.] Kofránek, J., Velan, T., Janicadis, P., & Kerekeš, R. (2001). Diagnostic and treatment of virtual patients with Golem – multimedia simulator of physiological functions. V J. G. Anderson, & M. Kaptzer (Editor), *Simulation in the Health and Medical Sciences, 2001.* (stránky 157-164). San Diego: Society for Computer Simulation International, Simulation Councils.

[90.] Kofránek, J., Vrána, C., Velan, T., & Janicadis, P. (2001). Virtual Patients on the net – Multimedia simulation Golem. V L. Badernstern, & E. Ossinnilsson (Editor), *Proceedings of the European Conference on e-Learning in a Lifelong Learning Perspective.* (stránky 228-238). Lund: Lund University.

[91.] Kössi, J., & Luostarinen, M. (2009). Virtual reality laparoscopic simulator as an aid in surgical resident education two years experience. *28*, stránky 48-54.

[92.] Lammers, R. L. (2006). Simulation: The New Teaching Tool. *Annals of Emergency Medicine ,*

49, stránky 505-507.

[93.] Lane, J. L. (2001). Simulation in medical education: a review. *Simulation&Gaming , 32*, stránky 297-314.

[94.] Leff, A., & Rayfield, J. T. (2007). Web-application development using the Model/View/Controller design patterns. *Fifth IEEE International Enterprise Distributed Object Computing Conference, ISBN: 0-7695-1345-X* (str. 118). Seattle: IEE International.

[95.] Lehmann, E. D., Tarin, C., Bondia, J., Teufel, E., & Deutsch, T. (2007). Incorporating a Generic Model of Subcutaneous Insulin Absorption into the AIDA v4 Diabetes Simulator. *Journal of Diabetes Science and Technology , 1*, stránky 780-793.

[96.] Lighthall, G. K. (2007). The Use of Clinical Simulation Systems to Train Critical Care Physicians. *Journal of Intensive Care Medicine , 22*, stránky 257-269.

[97.] Lipovszki, G., & Aradi, P. (2006). Simulating complex system and processes in LabView. *Journal of mathematical Sciences , 132*, stránky 629-636.

[98.] Little, J. A., Beres, J., Hinkson, G., Rader, D., & Croney, J. (2009). *Silverlight 3 programer's reference (Wronx programer to programer).* Indianopolis: Wronx Wiley.

[99.] Liu, A., Tendick, F., Cleary, K., & Kaufmann, C. (2003). A Survey of surgical simulation: applications technology, and education. *Presence , 12*, stránky 559-613.

[100.] Lloyd, C. M., Halstead, M. D., & Nielsen, P. F. (2004). CellML: its future, present and past. *Progress in Biophysics and Molecular Biology , 85*, stránky 433-450.

[101.] Loew, L. M., & Schaff, J. (2001). The Virtual Cell: a software environment for computational cell biology. *Trends in biotechnology* , stránky 401-406.

[102.] Logan, J. D., & Wolesensky, J. D. (2009). Mathematical methods in biology. Hoboken, NJ: John Wiley & Sons, Inc.

[103.] Mandel, J. E., Martin, J. F., Schneider, A. M., & Smith, M. T. (1985). Toward realism in modeling the clinical administration of a cardiovascular drug. *Anesthesiology , 63*, str. A504.

[104.] Maršálek, P., & Kofránek, J. (2005). Spike encoding mechanisms in the sound localization pathway. *Biosystems , 79*, stránky 191-198.

[105.] Martin, J. F., Schneider, A. M., Mandel, J. E., Prutow, R. J., & Smith, N. T. (1986). A new cardiovascular model for real-time applications. *Transactions of the Society for Computer Simulation , 3*, stránky 31-66.

[106.] Masuzawa, T., Fukui, Y., & Smith, N. T. (1991). Cardiovascular simulation using a multiple modeling method on a digital computer—Simulation of interaction between the cardiovascular system and angiotensin II. *Journal of Clinical Monitoring and Computing , 8*, stránky 50-58.

[107.] Masuzawa, T., Fukui, Y., & Smith, N. T. (1988). Simulation Model of Cardiovascular Response for Drug Administration. *Journal of Clinical Monitoring , 4*, str. 63.

[108.] Mateják, M., Privitzer, P., & Kofránek, J. (2008). Modelica vs. blokovo-orientované jazyky matematického modelovania. V J. Janech (Editor), *Objekty' 2008* (stránky 79-94). Žilina: Edis.

[109.] Mathwoorks. (2009). *Simscape 3.2 Model and simulate multidomain physical systems.* Načteno z http://www.mathworks.com/products/simscape/.

[110.] Matoušek, S., Kofránek, J., & Rees, E. S. (1989). Independence of Variables in Stewart's model of the acid-base chemistry of the blood plasma . *Proceedings of the IFAC Symposium on Modeling and Control in Biomedical Systems, Aalborg, Denmark August 12-14, 2009*, (pp. 246-250). Aalborg. Práce je dostupná na adrese http://www.physiome.cz/references/AcidBaseIFAC09.pdf.

[111.] McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous acti-

vity. *Bulletin of Mathematical Biology , 5*, stránky 115-133.

[112.] McGaghie, W. C., Siddall, V. J., Mazmanian, P. E., & Myers, J. (2009). Lessons for continuing medical education from simulation research in undergraduate and graduate medical education. *Chest , 165*, stránky 625-685.

[113.] McLeod, J. (1967). PHYSBE...a year later. *Simulation , 10*, stránky 37-45.

[114.] McLeod, J. (1966). PHYSBE: A ophysiological simulation benchmark experiment. *Simulation , 15*, stránky 324-329.

[115.] McLeod, J. (1970). Toward uniform documentation-PHYSBE and CSMP. *Simulation , 14*, stránky 215-220.

[116.] Meyers, R. D., & Doherty, C. L. (2008). *Web-Human physiology teaching simulation (Physiology in health, disease and during therapy)*. Načteno z http://placid.skidmore.edu/human/index.php.

[117.] Milhorn, H. T. (1966). The application of control theory to physiological systems. Philadelphia, London, Toronto: W.B. Saunders.

[118.] Montani, J. P., Adair, T. H., Summers, R. L., Coleman, T. G., & Guyton, A. C. (1989). A simulation support system for solving large physiological models on microcomputers. *Int. J. Blomed. Comput. , 24*, stránky 41-54.

[119.] Obdržálek, J., & Kofránek, J. (2004). *Teorie ladění.* (P. 1. LF UK, Producent) Načteno z http://patf-biokyb.lf1.cuni.cz/~obdrzalek/ladeni.htm.

[120.] Oomnes, C., Breklemans, M., & Baaijens, F. (2009). Biomechanics: concepts and computation. Cambridge: Cambridge University Press.

[121.] Ottesen, M. S., Olufsen, M. J., & Larsen, J. K. (2004). *Applied Mathematical Models in Human Physiology.* Philadelphia: Society for Indistrial amd Applied Mathematics.

[122.] Pitts, W. S., & McCulloch, W. (1947). How we know universals the perception of auditory and visula forms. *Bulletin of Mathematical Biophysics* , stránky 127-147.

[123.] Raymond, G. M., Butterworth, E., & Bassingthwaighte, J. (2003). JSIM: Free Software Package for Teaching Physiological Modeling and Research. *Experimental Biology , 280*, stránky 102-107.

[124.] Reed, K., & Lehmann, E. D. (2005). Diabetes website review: www. 2aida. org. *Diabetes Technology & Therapeutics* , stránky 741-754.

[125.] Roccetti, M. (2001). A Design for a Simulation-based Multimedia Learning Environment. *Simulation , 76*, stránky 214-221.

[126.] Rodriguez-Barbero, A., & Lopez-Novoa, J. M. (2009). Teaching integrative physiology using the quantitative circulatory physiology model and case discussion method: evaluation of the learning experience. *Advances in Physiology Education , 32*, stránky 304-311.

[127.] Rosen, K. R. (2008). The history of medical simulation. *Journal of Critical Care , 23*, stránky 157-166.

[128.] Sadot, A., Fischer, J., Admanit, I., Stern, M. J., Hubbard, E. J., & Harel, D. (2008). Toward Verified Biological Models. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on Comutational biologz and informatics* (5), stránky 223-234.

[129.] Satava, R. M. (2008). Historical review of surgical simulation— a personal perspective. *World Journal of Surgery , 32*, stránky 141-148.

[130.] Sawyer, T., Hara, K., Thompson, M. W., Chan, D. S., & Berg, B. (2009). Modification of the Laerdal SimBaby to Include an Integrated Umbilical Cannulation Task Trainer. *Simulation in Healthcare , 4*, stránky 174-178.

[131.] Sells, C., & Griffins, I. (2007). *Programming WPF.* Beijing, Cabridge,Farhan, Köln, Sebastopol, Taipe, Tokyo: O'Reilly.

[132.] Seropian, M. A., Brown, K., Gavilanes, J. S., & Deriggers, B. (2004). Simulation: not just a manikin. *J Nurs Educ. , 43*, stránky 164-9.

[133.] Sethi, A. S., Peine, W. J., Mohammadi, Y., & Sundaram, C. P. (2009). Validation of a Novel Virtual Reality Robotic Simulator. *Journal of Endourology , 23*, stránky 503-508.

[134.] Sheppard, C. W. (1948). The theory of the study of transfers within a multi-compartment system using isotopic tracers. *Journal of Applied Physics , 19*, str. 70.

[135.] Siggaard-Andersen, M., & Siggaard-Andersen, O. (1995). Oxygen status algorithm, version 3, with some applications. *Acta Anaesth Scand , 39, Suppl 107*, stránky 13-20.

[136.] Silbernagl, S., & Despopoulos, A. (2003, české vydání 2004). *Tashenatlas der Physiologie; české vydání: Atlas fyziologie člověka* (6. vyd.). Stuttgart: Georg Thieme Verlag, české vydání Praha: Grada.

[137.] Silbernagl, S., & Lang, F. (1998, české vydání 2001). *Taschenatlas der Pathophysiologie.* Stuttgart: Georg Thieme Verlag, české vydání Praha: Grada.

[138.] Sirker, A. A., Rhodes, A., & Grounds, R. M. (2001). Acid-base physiology: the ‚traditional' and ‚modern' approaches. *Anesthesia , 57*, stránky 348-356.

[139.] Smith, N. T. (1987). Mathematical Model of Uptake and Distribution of Inhalation Anaesthetic Agents. V *Anesthesia par Inhalation* (stránky 87-118). Paris: Arnette Publishers.

[140.] Smith, N. T., & Starko, K. (1995). PC-based anesthesia simulators. *Journal of Anesthesia (Japanese) , 9*, stránky 1-14.

[141.] Smith, N. T., Starko, K., & Davidson, T. (1996). PC-based anesthesia simulators. V K. Ikeda (Editor), *Anesthesia: Implications for the Coming Century* (stránky 1-14 (plus CD ROM)). Tokyo: Churchill Livingstone.

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

[143.] Stodulka, P., Privitzer, P., Kofránek, J., & Vacek, O. (2007). Development of WEB accessible medical educational simulators. V B. Zupanic, R. Karba, & S. Blažič (Editor), *Proceedings of the 6th EUROSIM Congress on Modeling and Simulation, Vol. 2. Full Papers (CD).* (stránky MO-3-P4-2, 1-6). Ljubljana: University of Ljubljana. Práce je dostupná na adrese http://www.physiome.cz/references/EUROSIM2007Stod.pdf.

[144.] Thamin, H. (2008). *Modelling the respiratory control system in human subjects for excercise conditions. Ph.D. Thesis.* Glasgow: University of Glasgow (dostupno z http://theses.gla.ac.uk/421/01/2008thamrinphd.pdf).

[145.] Thomas, R. S., Baconnier, P., Fontecave, J., Francoise, J., Guillaud, F., Hannaert, P., a další. (2008). SAPHIR: a physiome core model of body fluid homeostasis and blood pressure regulation. *Philosophical Transactions of the Royal Society , 366*, stránky 3175-3197.

[146.] Toro-Troconis, M., & Boulos, M. N. (2009). Musings in the state of ‚3D virtual worlds for health and healthcare in August 2009. *Journal of Virtual Worlds Research , 2*, stránky https://journals.tdl.org/jvwr/article/view/629/496, 1-15.

[147.] Toro-Troconis, M., Partridge, M., & Barret, M. (2008). Game-based learning for the delivery of virtual patients in Second Life. *Issues and News on Learning and Teaching in Medicine, Dentistry and Veterinary Medicine , 1*, stránky 3-5.

[148.] van Oosterom, A., & Oostendorp, T. F. (2004). ECGSIM: an interactive tool for studying the genesis of QRST waveforms. *Heart , 9*, stránky 165-168.

[149.] Van Vliet, B. N., & Montani, J. P. (2005). Circulation and fluid volume control. V *Integrative Physiology in the Proteomica and Post Genomics Age.* (stránky 43-66). Humana Press.

[150.] Verigo, V. V. (1987). Systěmnye metody kometody v kosmičeskoj medicine i biologii. (NASA technical translation NASA TT-20291: Systems methods in space biology and medicine). *Problemy kosmičeskoj biologii , 55*, stránky 1-216.

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

[152.] Wayne, D., Didwania, A., Feniglass, J., Fudala, M. J., Barsuk, J. H., & McGaghie, W. C. (2008). Simulation-based education improves quality of care during cardiac arrest team responses at an academic teaching hospital. *Chest , 133*, stránky 56-61.

[153.] White, R. J., & Phee, J. C. (2006). The Digital Astronauts: an integrated modeling and database system for space biomedical research and operations. *Acta Astronautica , 60*, stránky 273-280.

[154.] Williams, B. (2008). *Microsoft Expression Blend unleashed, First edition.* Indeanopolis: Sams.

[155.] Wünsch, Z. (1969). Minimum lékařské kybernetiky (Základní koncepce, metodiky a způsoby aplikací). Praha: Učební texty vysokých škol, UK, Fakulta všeobecného lékařství, SPN.

[156.] Wünsch, Z. (1974). Biokybernetika ve výuce fyziologie. *Československá fyziologie* , stránky 569-573.

[157.] Wünsch, Z., & Trojan, S. (1986). Koncepce výuky biokybernetiky v kurzu fyziologie. *Československá fyziologie , 35*, stránky 385-386.

[158.] Wünsch, Z., Dostál, C., & Veselý, A. (1977). *Základy lékařské kybernetiky.* Praha: Avicenum.

[159.] Wünsch, Z., Kripner, T., Kofránek, J., & Andrlík, M. (2004). The mechanical properties of skeletal muscle - Multimedia simulation educational software. V G. Attiya, & Y. Hamam (Editor), *Proceedings of the 5th EUROSIM Congress on Modeling and Simulation.Full Papers CD Volume.*, (stránky 28-32). Marne La Vallée. Práce je dostupná na adrese http://www.physiome.cz/references/EUROSIMWunsch2004.pdf.

[160.] Wünsch, Z., Matúš, M., & Kofránek, J. (2007). Physiological feedback modelling in medical education. V B. Zupanic, R. Karba, & S. Blažič (Editor), *Proceedings of the 6th EUROSIM Congress on Modeling and Simulation, Vol. 2. Full Papers (CD).* (stránky TU-1-P7-5: 1-7.). Ljubljana: University of Ljubljana. Práce je dostupná na adrese http://www.physiome.cz/references/EUROSIMWunsch2007.pdf.